Centre Énergie, Matériaux, et Télécommunications

DFROSTNET : A DEEP NEURAL NETWORK FOR REAL-TIME PULSE CHARACTERIZATION FROM FREQUENCY-RESOLVED OPTICAL SWITCHING (FROST) TIME-FREQUENCY REPRESENTATIONS

Par

Sydney Lepard

Mémoire présenté pour l'obtention du grade de
Maître es Sciences, M.Sc.
en science de l'énergie et des matériaux

Jury d'évaluation

Examinateur externe	Prof Denis Seletskiy Department of Engineering Physics Polytechnique Montréal
Examinateur interne	Prof François Vidal Institut national de la recherche scientifique Énergie, Matériaux, et Télécommunications
Directeur de recherche	Prof François Légaré Institut national de la recherche scientifique Énergie, Matériaux, et Télécommunications

ACKNOWLEDGMENTS

First and foremost, I'd like to thank from the bottom of my heart my research supervisor, Prof François Légaré, for the invaluable support he has offered me throughout this degree. The impact that he has had, both on my time at INRS and on my career, cannot be overstated. It is directly thanks to his kindness and integrity that I've been able to complete this degree. I'd also like to thank my post-docs, Drs Gaëtan Jargot and Marie Ouille, and ALLS director, Dr Heide Ibrahim, who reviewed scholarship applications, helped to brainstorm solutions, and supported me throughout this project. Thank you as well to my fellow ALLS students who contributed to this work : Drs Mayank Kumar and Adrien Longa for their help setting up PtyChoPy and providing experimental data, as well as summer intern Enzo Sebiane who was a very enthusiastic rubber duck.

While my work in Paris did not contribute directly to this thesis, I would still like to thank those who made it such a wonderful experience : Drs Boris Vodungbo, Emmanuelle Jal, and Marcel Hannes, for their supervision, guidance, and support. As well, thank you to the students that welcomed me with open arms : Moundji, Angèl, Octave, Robert, Amina, and especially, Houda, for showing me the best time in Europe.

Additionally, I'd like to thank the people I meet while at INRS. Phoebe, for being the best intern ever. Rob, for fixing all my problems. Celine, for a friendship that made it out of the lab. Giacomo, for teaching me how to balance photo-diodes.

Thank you to miscellaneous friends and family. Aunt Bernice and Uncle Gerry, for tea dates, career advice, and BBQs. Sophia, my queen, the coolest girl I know. Bibi, for surviving the trenches of UoG with me, as well as Anita and Guinness Benito (aka Printed Circuit Board) for opening their home to me. My best friend, Mason, for being the kindest, steadiest, and truest person I have the honor of knowing. My cat, Caramelle, who did everything in her power to distract me from this thesis. Lastly, thank you to my parents and family, who contributed significantly to the person I am today.

Finally, as I reflect on the decisions that brought me to this point, I'd like to thank two people without whom I would not have proceeded on this path. First was my 12th grade physics teacher at the Ontario Science Center, Felipe Almeida. His enthusiasm for Physics Friday demos was infectious and, for the first time, I really enjoyed physics. Second was my undergraduate electrical circuits professor, Prof Christopher Collier, who offered me my first research position in optics. We happened to strike up a conversation at the Tims' on campus and I was sold the second I heard "laser". I continued to work with Dr Collier's lab throughout my bachelor's degree and this significantly contributed to my decision to start this master's degree.

Lastly, I'd like to acknowledge the huge amount of work and commitment that I've put in to get to this point. Almost a decade of higher education has culminated in this thesis. So, to whomever is reading this, thanks for taking the time and I hope you enjoy !

ABSTRACT

This thesis investigates the use of artificial intelligence for the characterization of ultrafast laser pulses, focusing on Frequency-Resolved Optical Switching (FROSt). Ultrafast lasers, with pulse durations on the order of femtoseconds, are essential for applications in telecommunications, medicine, and material science. However, the precise characterization of their amplitude and phase remains challenging due to the phase retrieval problem, which arises from the inability to directly measure the complex electric field. Traditional recovery methods, such as ptychography, are computationally intensive and susceptible to experimental noise. This work introduces an artificial intelligence-based approach to address these limitations. The study aims to replicate the numerical recovery process used on FROSt traces with comparable accuracy and significantly reduced computation time. The research encompasses the generation of simulated training datasets, the design of a multi-resolution architecture, and the development of training protocols employing both supervised and unsupervised learning methods. Experimental results confirm the feasibility of the proposed model, achieving reliable reconstruction of pulse profiles ($< 5\%$ error) while reducing reliance on pre-processing steps. This work represents a step toward real-time, robust pulse characterization and sets the stage for future improvements in ultrafast pulse characterization and material sciences.

Keywords Ultrafast Laser Characterization ; Phase Retrieval ; Time-Frequency Representation ; Frequency-Resolved Optical Switching (FROSt) ; Ptychography ; Deep Neural Networks ; Artificial Intelligence.

RÉSUMÉ

Cette thèse étudie les méthodologies avancées pour la caractérisation des impulsions laser ultrarapides, en se concentrant sur la commutation optique résolue en fréquence (frequency-resolved optical switching, FROSt). Les lasers ultrarapides, avec des durées d'impulsion de l'ordre de la femtoseconde, sont essentiels pour des applications diverses. Cependant, la caractérisation précise de leurs composantes d'amplitude et de phase reste difficile en raison du problème de récupération de la phase. Les méthodes de récupération traditionnelles, telles que la ptychographie, sont très gourmandes en ressources informatiques et sensibles au bruit expérimental. Ce travail introduit une approche basée sur l'intelligence artificielle pour répondre à ces limitations. L'étude vise à reproduire le processus ptychographique avec une précision comparable et un temps de calcul considérablement réduit. La recherche englobe la génération d'ensembles de données simulées, la conception d'une architecture et le développement de protocoles d'entraînement utilisant à la fois l'apprentissage supervisé et non supervisé. Les résultats expérimentaux confirment la faisabilité du modèle proposé, qui permet une reconstruction fiable des profils d'impulsion (< 5 % erreur) tout en réduisant la dépendance à l'égard des étapes de prétraitement. Ce travail représente une étape vers une caractérisation robuste des impulsions en temps réel et ouvre la voie à de futures améliorations dans la caractérisation des impulsions ultrarapides et les sciences des matériaux.

Mots-clés Caractérisation ultrarapide ; Récupération de phase ; Représentation temps-fréquence ; Commutation optique résolue en fréquence (FROSt) ; Ptychographie ; Réseaux neuronaux profonds ; Intelligence artificielle.

SYNOPSIS

DFROStNET : Un réseau neuronal profond pour la characterization des impulsions optiques en temps réel à partir de représentations temps-fréquence de la commutation optique résolue en fréquence

0.1 Introduction

Le développement des lasers a eu un impact transformateur et considérable sur la technologie au cours des 60 dernières années, en particulier dans des domaines tels que les télécommunications, la médecine et la science des matériaux (Strickland et al., 1985; Knox, 2000; Sugioka et al., 2014). Dans le domaine des télécommunications, la fibre optique a révolutionné la communication mondiale en permettant la transmission de données à grande vitesse (Pogna et al., 2021; Foster et al., 2008; Fermann et al., 1997; Martínez et al., 2010). De même, en médecine, les lasers sont devenus des outils essentiels pour les techniques de diagnostic non invasives et les chirurgies de précision (Errico et al., 2015; Hebden et al., 1995; Yun et al., 2017; Hoy et al., 2013). En science des matériaux, les lasers ont fait progresser des méthodes de fabrications et fourni des outils analytiques précis pour l'étude des propriétés des matériaux (Rosker et al., 1987; Cireasa et al., 2015). Les lasers ultrarapides, avec des durées d'impulsion de l'ordre de la femtoseconde (1×10^{-15} s), sont indispensables dans ces domaines.

La caractérisation des lasers ultrarapides est essentielle pour leur utilisation efficace. Des impulsions mal caractérisées peuvent entraîner des problèmes tels que la distorsion du signal dans les télécommunications (Knox, 2000; Wada, 2004) et des artefacts d'imagerie en médecine (Hoy et al., 2013; Hebden et al., 1995). Le défi de la caractérisation des impulsions ultrarapides réside dans leurs échelles de temps extrêmement brèves, qui sont intrinsèquement plus rapides que ce que les outils de mesure conventionnels peuvent enregistrer (Diels et al., 2006; Born et al., 2019). Les lasers ultrarapides doivent être décrits par leurs champs électriques complexes, qui nécessitent à la fois l'amplitude et la phase, en temps, t, ou en fréquence angulaire, ω. Le champ électrique, $E(t)$, et son spectre, $\tilde{E}(\omega - \omega_0)$, sont liés par la transformée de Fourier (TF), où ω_0 est la fréquence angulaire centrale. Le champ électrique est décrit par son enveloppe temporelle, $E_0(t)$, et sa phase, $\phi(t)$. De même, le spectre est décrit par l'amplitude, $\tilde{E}_0(\omega - \omega_0)$, et la phase spectrale, $\varphi(\omega - \omega_0)$.

$$\begin{aligned} E(t) &= E_0(t)\, e^{j(-\omega_0 t + \phi(t))} \\ \tilde{E}(\omega) &= \tilde{E}_0(\omega - \omega_0)\, e^{-j\varphi(\omega - \omega_0)} \end{aligned} \tag{1}$$

Pour caractériser les lasers ultrarapides, une méthode d'échantillonnage à grande vitesse est nécessaire. Selon le théorème de Nyquist, le taux d'échantillonnage doit être au moins deux fois

supérieur à la fréquence de l'événement le plus rapide enregistré (Oppenheim et al., 2007). Toutefois, cela est rarement possible pour les lasers ultrarapides, car ils sont intrinsèquement l'événement le plus rapide disponible. Par conséquent, les méthodes expérimentales sont limitées à la récupération des intensités du champ électrique en temps, $|E(t)|^2$, ou en fréquence, $|\tilde{E}(\omega)|^2$ (Diels et al., 2006; Trebino, 2002). À cause de ceci, il y a une perte inhérente d'informations puisque la récupération du champ électrique complexe à partir de l'intensité donne lieu à une infinité de phases possibles. Il s'agit d'un problème fondamental dans la caractérisation ultrarapide, connu sous le nom du problème de la récupération de phase (PRP). Une méthode courante pour réduire ce problème consiste à utiliser l'auto-échantillonnage, où l'impulsion est utilisée pour se caractériser elle-même. Pour le faire, on utilise des expériences pompe-sonde, dans lesquelles une impulsion est divisée en faisceaux pompe et sonde (l'un étant la copie retardée de l'autre) et dont l'intensité est mesurée au fil du temps. Un certain nombre de techniques expérimentales ont été élaborées à partir de cette idéologie, qui sera brièvement décrit.

0.1.1 Techniques de characterization

Dans le domaine temporel, les photodiodes sont souvant utilisées pour mesurer l'intensité d'un champ électrique dans le temps. Cependant, comme la dynamique d'une photodiode se produit sur une échelle de temps beaucoup plus lente que celle de l'impulsion ultrarapide, la mesure correspond à une intégration temporelle de l'intensité optique, et non du champ électrique lui-même. Il s'agit d'une limitation fondamentale de nombreuses mesures dans le domaine temporel, car la nature des impulsions ultrarapides est si brève dans le temps. Toutefois, des techniques de caractérisation ont été mises au point pour générer et mesurer des effets résolubles dans le temps qui sont proportionnels au champ électrique que l'on souhaite connaître (comme l'interférence entre pompe-sonde ou une modulation de polarization) (van der Valk et al., 2004; Carnio et al., 2024; Born et al., 2019; Hecht, 2017). Bien que les techniques temporelles fournissent certaines informations sur les impulsions ultrarapides, elles ne parviennent pas à décrire à la fois l'amplitude et la phase temporelle de l'impulsion. Cela est dû aux limites inhérentes à la mesure d'un événement ultrarapide. Toutefois, on peut généralement comprendre que les domaines du temps et de la fréquence sont inversement liés, de sorte qu'un événement temporel étroit aura un large spectre. Ainsi, une impulsion brève dans le temps sera généralement associée à un spectre plus large. Cette propriété peut être exploitée en appliquant les techniques de pompe-sondes au domaine spectral.

Dans le domaine spectral, une technique révolutionnaire est le "Frequency-Resolved Optical Gating" (FROG), qui utilise la caractérisation simultanée en temps et en fréquence pour générer une représentation temps-fréquence (RTF) 2D des deux variables (Trebino, 2002). L'impulsion en question est divisée et un effet secondaire est généré, en utilisant des effets non-linéaires tels que la génération de la seconde harmonique (GSH). La seconde harmonique est résolue spectralement avec un spectromètre à de nombreux décalages pompe-sonde. Cette technique permet une récupération complète du champ électrique (amplitude et phase) par le biais d'un post-traitement numérique, ce qui nous permet de surmonter les limites de la caractérisation temporelle (DeLong et al., 1994). Un certain nombre de techniques sont nées de la FROG, notamment la GSH FROG, dont nous avons parlé plus haut. Ces méthodes offrent des informations précieuses sur la dynamique des impulsions ultrarapides, permettant aux chercheurs de mieux comprendre et manipuler la lumière ultrarapide. Cependant, la FROG présente des limites expérimentales, principalement en ce qui concerne les conditions nécessaires pour générer la seconde harmonique. Par conséquent, la FROG s'applique

principalement aux faisceaux de haute intensité dont les exigences en matière de correspondance de phase sont faciles à satisfaire.

0.1.2 Frequency-resolved optical switching

Compte tenu de ces limites expérimentales, les techniques FROG traditionnelles ont stimulé le développement de variantes telles que la "Frequency-Resolved Optical Switching" (FROSt), qui offre une plus grande flexibilité dans les conditions expérimentales (c'est-à-dire pour les impulsions de faible intensité qui ne peuvent pas générer des effets secondaires tel qu'une seconde harmonique) (Leblanc et al., 2019, 2021; Brizard et al., 2020).

La différence entre FROSt et autres techniques réside dans le fait que le FROSt repose sur un matériau appelé commutateur (ou "switch"), qui est essentiel pour générer les interférences nécessaires à la caractérisation des impulsions. Le switch est un semi-conducteur qui devient optiquement opaque lorsqu'il est photo-excité par un faisceau de haute intensité, ce qui lui permet d'ouvrir la porte de l'impulsion d'une manière similaire à FROG. Ce processus est illustré par la Figure 1, où l'on voit une impulsion interagir avec le switch à deux délais temporels distincts. À $\tau = \tau_1$, la pompe à haute intensité, $E_{HI}(t-\tau)$, interagit avec le switch bien avant la sonde, $E_P(t)$. Dans ce cas, le switch est désactivé (transmission de près de 0 % sur l'ensemble du spectre) et seule une très petite partie de la sonde est propagée. À $\tau = \tau_2$, la pompe interrompt le switch après la sonde, laissant le switch allumé (une transmission de 100 %) et propageant la totalité de la sonde. En conséquence, le spectre de sortie est la convolution entre le switch, $S(t-\tau)$, et la sonde, $E_P(t)$, et dépend du délai temporel, τ. L'intensité peut être décrit de manière équivalente dans l'Équation 2.

$$I(\omega,\tau) = \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt \right|^2 \tag{2}$$

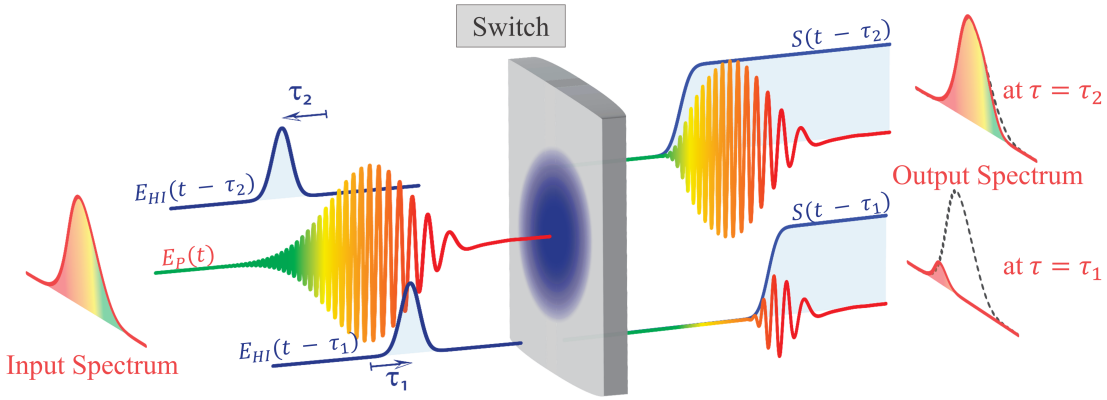

Figure 1 : Théorie Principal de FROSt.

Un matériau de commutation solide (Switch) est photo-excité par l'impulsion de haute intensité, $E_{HI}(t-\tau)$, à plusieurs retards temporels, τ. L'inversion de population qui en résulte dans le switch génère la fenêtre coulissante, $S(t-\tau)$, qui est convoluée avec l'impulsion de la sonde, $E_P(t)$. L'impulsion convolu est mesurée à la sortie par un spectromètre pour de nombreuses valeurs du délai temporel, τ. À $\tau = \tau_1$, le matériau est presque entièrement opaque et seule une petite partie du spectre de la sonde est enregistrée. À $\tau = \tau_2$, le matériau est transparent, ce qui permet d'enregistrer la totalité du spectre. Reproduite de Leblanc et al. (2019).

Il est important de noter que les modèles d'interférence résultants, collectés sous la forme d'une trace 2D, $I(\omega, \tau)$, encodent l'évolution du spectre de la sonde en fonction du délai temporel. Cette trace fournit les données nécessaires pour retrouver les profils de la sonde et du switch à l'aide de méthodes de récupération numérique appropriées, telles que la ptychographie. La récupération ptychographique est utilisée pour extraire le champ électrique de l'impulsion et du switch à partir des traces FROSt. Cette technique exploite les redondances des données acquises pour reconstruire les variables du système. Les défis de la récupération ptychographique comprennent le potentiel d'erreur dans la solution reconstruite et la difficulté de garantir la solution exacte en raison de la nature itérative de la méthode. Les contraintes et les étapes de prétraitement peuvent contribuer à atténuer ces difficultés, mais elles introduisent de la complexité et des biais potentiels. Même si la dépendance à l'égard de la récupération ptychographique est une limitation, FROSt ne repose pas sur les effets non linéaires des cristaux, ce qui le rend adapté aux lasers de faible intensité et le libère des limitations liées à la correspondance de phase. Par conséquent, FROSt est généralement une technique attrayante pour la caractérisation précise des impulsions dans des scénarios où d'autres méthodes sont limitées.

0.1.3 Intelligence artificielle

Les progrès récents de l'intelligence artificielle (IA), en particulier sous la forme de réseaux de neurones (RN), offrent des solutions prometteuses aux processus numériques difficiles, tels que la récupération ptychographique (Alzubaidi et al., 2021; Brunton et al., 2017; Goodfellow et al., 2016; Bishop, 2006). Ces modèles d'IA, lorsqu'ils sont correctement entraînés pour une tâche spécifique, peuvent approximer les solutions traditionnellement obtenues par des méthodes numériques, en atteignant une grande précision en temps réel. Essentiellement, les RNs peuvent reproduire le processus de résolution mathématique des méthodes conventionnelles sans effectuer les calculs directs (Brunton et al., 2017; Goodfellow et al., 2016). Ce concept s'aligne sur les analyses statistiques, telles que la régression non-linéaire, où le modèle établit des relations entre les paramètres et les réponses du système (Kingma et al., 2015; Li et al., 2025). Lorsqu'elle est bien entraînée, les RNs peuvent produire des solutions très précises, offrant ainsi une alternative puissante aux méthodes numériques traditionnelles.

Les preuves du potentiel de l'IA dans la reproduction des méthodes de récupération sont de plus en plus nombreuses (Zahavy et al., 2018; Gianani et al., 2024). Par exemple, Zahavy (Zahavy et al., 2018) ont démontré qu'un RN pouvait reconstruire des impulsions à partir d'une trace GSH FROG, avec des taux d'erreur comparables à ceux des méthodes de récupération conventionnelles. Leur méthode élimine le besoin de prétraitement et réduit considérablement le temps de calcul en permettant l'extraction des données en temps réel. Malgré ces avantages, ces algorithmes nécessitent des processus de formation complexes et gourmands en ressources. En outre, les travaux de Zahavy étaient axés sur les traces GSH FROG, qui impliquent des méthodes de récupération plus simples en raison des propriétés d'auto-génération du GSH FROG. L'extension de cette méthodologie à des RTFs plus complexes, comme FROSt, reste un défi important.

Le but de ce travail est de développer un algorithme entraîné capable de récupérer les profils temporels (amplitude et phase) d'une impulsion ultracourte à partir de sa trace FROSt avec une précision comparable aux méthodes traditionnelles de ptychographie. Inspiré par Zahavy (Zahavy et al., 2018), ce travail implique la création d'un ensemble de données simulées, la définition d'une procédure d'apprentissage et l'établissement d'étapes de validation. Dans cette section, nous avons

présenté une version abrégée de l'ensemble du texte. Elle est structurée comme suit : la Section 0.1 présente le contexte, les méthodes de caractérisation des impulsions et la théorie de FROSt. La Section 0.2 couvre le PRP, la récupération ptychographique et les outils d'IA utilisés. La Section 0.3 décrit la méthodologie, y compris la génération d'ensembles de données, l'architecture du RN, la formation et la validation. La Section 0.4 présente les résultats qui sont discuter dans la Section 0.5 et la Section 0.6 conclut sur les prochaines étapes.

0.2 Théorie

0.2.1 Le problème de la récupération de la phase

Le champ électrique d'une impulsion ultrarapide est une entité complexe où la phase — à la fois temporelle, $\phi(t)$, et spectrale, $\varphi(\omega)$ — joue un rôle crucial. La relation entre les propriétés temporelles et spectrales de l'impulsion est grandement inversement proportionnelle, la largeur de bande spectrale étant inversement liée à la durée temporelle. Cependant, la phase introduit une complexité supplémentaire, qui peut être comprise grâce au principe d'incertitude temps-fréquence. Ce principe stipule que les propriétés temporelles et spectrales ne peuvent pas être résolus simultanément avec des précisions équivalentes. Plus précisément, il affirme qu'il existe une incertitude minimale lorsqu'on relie la durée temporelle, Δt, et la largeur de bande spectrale, $\Delta\omega$, d'une impulsion, représentée par l'inégalité : $\Delta t\,\Delta\omega \geq PTF$. Le produit temp-fréquence (PTF) définit la forme de l'impulsion (l'impulsion Gaussienne ayant un $PTF = 0,441$ (Diels et al., 2006)), et le cas optimal d'une impulsion à sa durée la plus courte possible remplissant l'égalité : $\Delta t\,\Delta\omega = PTF$. Lorsque des phases plus complexes sont introduites, l'impulsion s'élargit et le principe d'incertitude ne peut pas être résolu exactement.

L'importance de visualiser à la fois les profils de temps et de fréquence apparaît clairement lorsque l'on considère l'incertitude entre les propriétés temporelles et spectrales. Une RTF en 2D, telle qu'un spectrogramme, est utile pour capturer les deux profils. Ces représentations peuvent être générées par calcul (grâce à la TF à court terme) ou par des méthodes expérimentales (telles que FROG et FROSt). Dans tous les cas, la représentation 2D capture l'évolution temporelle du spectre, fournissant une vue de la manière dont le contenu en fréquence de l'impulsion change au fil du temps. Ceci est intrinsèquement important, puisque la phase d'une impulsion modifiera la RTF pour deux représentations 1D identiques. Ainsi, les champs électriques peuvent être classés en deux types d'impulsions : la transformée de Fourier Limitée (TFL) et les impulsions chirpées. La première impulsion, avec une phase temporelle nulle, $\phi(t) = 0$, a la largeur de bande spectrale la plus étroite, adhérant à la forme la plus simple du principe d'incertitude (où il s'agit d'une durée minimum temporelle pour la largeur de bande spectrale donnée). En revanche, les impulsions chirpées ont la même largeur de bande spectrale, mais leur profil temporel est élargi en raison de l'introduction d'une phase dépendant du temps. La phase a un impact significatif sur les impulsions ultrarapides, car l'inclusion de la phase peut modifier considérablement le profil temporel. Ainsi, pour caractériser pleinement une impulsion, sa phase doit également être décrite.

Dans les installations expérimentales, seule l'intensité d'une impulsion ultrarapide peut être mesurée, car le champ électrique oscille trop rapidement pour être observé directement. Il en résulte une RTF 2D à valeurs réelles, où les composantes réelles et imaginaires de l'impulsion doivent être récupérées. Cette récupération est un simple problème de Pythagore, où la magnitude et l'angle de l'impulsion sont nécessaires pour résoudre les composantes réelles et imaginaires. Dans la pratique, ce

problème peut devenir extrêmement complexe et nécessite la mise en œuvre de solutions numériques. Dans le cas de FROSt, nous appliquons la récupération ptychographique afin de déterminer la solution la plus probable à la fois pour l'impulsion complexe et pour le switch.

0.2.2 Ptychographie

La ptychographie spatiale consiste à reconstruire une impulsion 2D et un objet en mesurant les schémas d'interférence à différents décalages spatiaux. La procédure repose sur la convolution spatiale de l'impulsion et l'objet. La ptychographie spatiale peut facilement être transposée au domaine temporel, où une impulsion et un objet peuvent être résolus temporellement à l'aide de la même technique. L'approche est analogue aux principes de FROSt et repose sur la mesure de l'interaction entre une impulsion de sonde et le switch à différents délais temporels, τ. Cette représentation 2D peut ensuite être exploitée par le biais du processus de récupération ptychographique afin d'identifier les vecteurs d'entrée possibles. Pour ce faire, on se déplace de manière itérative dans les domaines de la fréquence et du temps, en appliquant des contraintes dans chacun d'eux pour assurer la convergence. La convergence réussie de cette méthode dépend de la résolution adéquate de la RTF. Cela dépend du chevauchement des informations entre les mesures, assurant une continuité suffisante pour une récupération de phase sans ambiguïté. En ptychographie temporelle, cette redondance est obtenue par une sélection minutieuse de la taille des pas de retard, τ, satisfaisant $\tau < 1/2\,\omega_{min}$. Cela garantit un chevauchement partiel entre les tranches temporelles successives, répondant ainsi aux exigences du théorème de Nyquist.

Processus de récupération. Nous présentons un bref aperçu du processus de récupération ptychographique, extrait de Leblanc (2016), et similare à ceux de Fienup (1982); Naganuma et al. (1989). L'objectif est de trouver la sonde, $E_P(t)$, et l'objet, $S(t-\tau)$, qui génèrent le champ électrique transmis, $E(t,\tau)$, avec un spectre, $\tilde{E}(\omega,\tau)$, qui correspond le mieux à l'intensité mesurée expérimentalement. Le processus en quatre étapes suivi à chaque itération est illustré à la Figure 2 et décrit pour l'itération $g = 0$.

La Figure 2 présente un seul cycle du processus de récupération. À l'étape (a), une estimation est faite du champ électrique de la sonde, $E_{P,g}(t)$, ou de l'objet, $S_g(t-\tau)$. Le champ électrique théorique résultant dans le temps, $E_g(t,\tau)$, est calculé. À l'étape (b), une TF est appliqué pour acquérir le spectre au niveau du plan de détection, $|\tilde{E}_g(\omega,\tau)|^2$, qui peut être écrit de manière équivalente comme $I_g(\omega,\tau)$. Pour faciliter la convergence, l'étape (c) utilise l'intensité expérimentale, $I_{XP}(\omega,\tau)$, pour obtenir la véritable amplitude spectrale, $\sqrt{I_{XP}(\omega,\tau)}$. Cette amplitude est combinée avec la composante de phase spectrale déterminée par cette itération pour générer un spectre modifié, $\tilde{E}_{g+1}(\omega,\tau)$. À l'étape (d), la TF inverse est pris pour récupérer le champ électrique modifié pour l'itération suivante, $E_{g+1}(t,\tau)$. Cependant, un problème se pose à ce stage, car la supposition suivante, $E_{g+1}(t,\tau)$, est une fonction de la sonde, $E_{P,g+1}(t)$, et de l'objet, $S_{g+1}(t-\tau)$. Ces deux quantités sont inconnues et renvoient à une variation de la problème fondamentale. Il est nécessaire de déterminer quelle sonde et quel objet génèrent le champ électrique, $E_{g+1}(t,\tau)$, ce que l'on obtient en appliquant ce que l'on appelle la contrainte ptychographique.

La contrainte ptychographique est fondamental pour les ptychographies et est donc bien établi dans d'autres travaux (Thibault et al., 2009; Leblanc, 2016; Bourassin-Bouchet et al., 2013; Lucchini et al., 2018). Pour déterminer les nouveaux profils, $E_{P,g+1}(t)$ et $S_{g+1}(t-\tau)$, on détermine la somme des deux profils qui se rapproche le plus du champ électrique de la dernière itération. Pour simplifier,

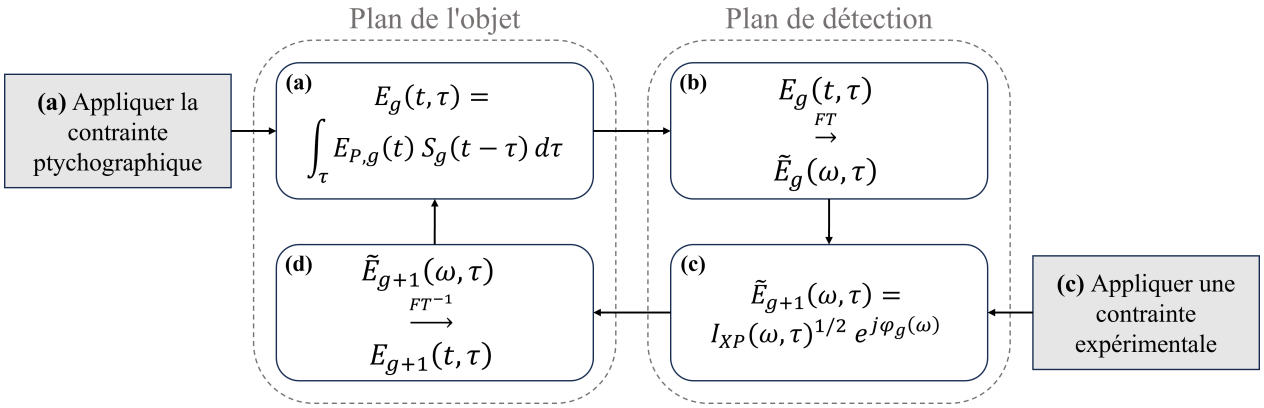

Figure 2 : Processus de la récupération ptychographique.

Les quatre étapes distinctes de la récupération ptychographique sont présentées. Pour l'itération initiale, $g = 0$, les entrées pour l'impulsion et l'objet, $E_{P,0}(t)$ et $S_0(t-\tau)$, sont choisies par hasard. À l'étape (a), le champ électrique résultant, $E_0(t,\tau)$, tel qu'il se produirait sur le plan de l'objet, est calculé à partir des suppositions initiales. À l'étape (b), la TF est utilisé pour acquérir le spectre correspondant sur le plan de détection, $\tilde{E}_0(\omega,\tau)$. À l'étape (c), une contrainte expérimentale est appliquée, où la phase de la supposition précédente est combinée à l'intensité recueillie expérimentalement pour améliorer la conversion. À l'étape (d), la TF inverse est acquis pour calculer le champ électrique au niveau du plan de l'objet pour l'itération suivante, $E_1(t,\tau)$. Lorsque le processus passe à l'itération suivante, $g = 1$, la contrainte ptychographique est appliquée. Ces étapes sont répétées de manière cyclique jusqu'à ce que la convergence soit atteinte, les sorties étant alors la meilleure estimation pour la sonde et l'objet.

dissons que $g = g + 1$ alors : $E_{g-1}(t,\tau) \simeq \int_\tau E_{P,g}(t)\, S_g(t-\tau)\, d\tau$. Nous déterminons l'étape suivante en identifiant analytiquement les entrées qui minimisent cette équation. Après avoir appliqué la contrainte ptychographique, nous aurons déterminé la prochaine estimation des vecteurs d'entrées, ce qui conduira à la prochaine itération du processus. En d'autres termes, à l'itération $g = 0$, la sonde et l'objet, $E_{P,0}(t)$ et $S_0(t-\tau)$, sont tous deux choisis au hasard pour obtenir une première estimation du spectre, $\tilde{E}_0(\omega,\tau)$. La sonde, $E_{P,0}(t)$, est ensuite combinée avec la sortie de l'étape (d), $E_1(t,\tau)$, pour déterminer l'entrée approximative pour $g = 1$ en utilisant la contrainte ptychographique. Par conséquent, pour $g = 1$, les entrées sont en fait $E_{P,0}(t)$ et $S_1(t-\tau)$ pour la sonde et l'objet, respectivement. De manière équivalente, une fois que $E_2(t,\tau)$ est déterminé à partir de $S_1(t-\tau)$ à l'étape (d) de $g = 1$, la contrainte ptychographique est appliquée pour produire $E_{P,2}(t)$. Cette opération est répétée jusqu'à convergence, chaque itération nécessitant l'approximation d'une seule des deux entrées, l'autre étant déterminée par la contrainte ptychographique. Le résultat final, après $g = G$ itérations, sera une approximation très proche des profils de la sonde et de l'objet. Comme pour tous les processus numériques, si cette erreur ne s'approche pas de zéro au cours de nombreuses itérations, l'algorithme est considéré comme divergent et une solution valide ne sera pas isolée. Cependant, s'il est exécuté avec succès, l'erreur finale sur la trace s'approchera de zéro et les profils de sortie, $E_{P,G}(t)$ et $S_G(t-\tau)$, seront considérés comme la solution.

0.2.3 Intelligence artificielle

L'objectif de ce travail est de reproduire le processus ptychographique à l'aide des techniques IA. Avant de plonger dans la méthodologie, il est essentiel de comprendre les opérations fondamentales des RNs. Les RNs peuvent être conceptualisés comme des couches de matrices effectuant des opérations mathématiques sur des données d'entrée. Ces couches, organisées en série ou en parallèle, imitent les équations mathématiques par le biais d'opérations d'algèbre linéaire. L'architecture d'un

RN fait référence à la disposition et à la fonction de ces couches. Chaque couche d'un RN effectue des opérations mathématiques spécifiques, les neurones étant les paramètres de ces opérations.

Les convolutions, fondamentales pour les RNs, fonctionnent comme des fonctions de fenêtre coulissante. En 1D, une convolution produit une série de sommes pondérées d'éléments d'entrée (où FROG et FROSt sont considérés comme des convolutions 1D). Cette opération s'étend à la 2D pour les données d'image, où les filtres (matrices) glissent sur les pixels de l'image pour produire des cartes de caractéristiques. Par exemple, l'application d'une matrice à noyau 3-par-3 à une entrée génère en sortie des cartes de caractéristiques influencées par les poids du noyau. Les cartes de caractéristiques peuvent être affinées davantage en utilisant plusieurs filtres et en organisant les couches convolutives en série. Cette disposition séquentielle permet aux couches successives d'extraire des détails plus fins de l'entrée. Des paramètres tels que le stride (taille du pas pour le mouvement du filtre) et le padding (pour contrôler les dimensions de la sortie) jouent un rôle essentiel dans les opérations convolutives. Ces paramètres garantissent que les sorties des couches successives sont compatibles pour la suite du traitement.

Les couches convolutives sont généralement suivies de couches d'aplatissement et d'élimination, avant de passer aux couches denses. L'utilisation d'une couche d'aplatissement permet de rassembler toutes les caractéristiques de chaque carte dans un seul canal, ce qui a pour effet d'égaliser tous les points de données. Souvent, une couche d'élimination est incluse après la couche d'aplatissement et avant les couches denses afin d'enlever un certain pourcentage d'éléments de la matrice. Le pourcentage de caractéristiques à supprimer, ainsi que les caractéristiques supprimées, sont des hyperparamètres qui modifient l'information dans le vecteur de sortie.

Le vecteur de sortie avec toutes les caractéristiques dans un seul canal est ensuite transmis à une couche dense, qui effectue simplement une transformation linéaire liée aux poids, W, de la couche. Dans une couche dense, les poids sont définis par la relation : $W(n) = A(n) + b$, où A est la magnitude et b le biais pour toute pixel n (Mallat, 2016; Goodfellow et al., 2016). Les couches denses reprennent les caractéristiques individuelles des cartes de caractéristiques aplaties et établissent des connexions linéaires entre elles. Les poids neuronaux peuvent ensuite être entraînés pour ajuster la contribution de chaque pixel à la sortie. Ces relations sont fondamentalement linéaires, ce qui limite la complexité des fonctions cartographiées, mais des non-linéarités peuvent être introduites par les couches d'activation.

Mathématiquement, une couche d'activation exécute l'opération suivante : $y_o = f(y_i)$, où y_i est l'entrée, y_o est la sortie et f est la fonction d'activation (Brunton et al., 2017). Cela s'applique aux fonctions d'activation non-linéaires, telles que l'unité linéaire rectifiée (ReLU) ($\text{ReLU}(n) = \max(0, n)$), sigmoïde ($\text{Sigmoïde}(n) = 1/(1 + e^{-n})$, et tanh $\left(\tanh(n) = (e^{2n} - 1)/(e^{2n} + 1)\right)$ (Brunton et al., 2017). Elles permettent au processus de formation d'extraire des relations non linéaires entre l'entrée et la sortie.

Une fois la fonction de chaque couche comprise, elles sont maintenant considérées dans le contexte d'un modèle plus large. Il a été établi que le fait de placer des couches convolutives en séries et en parallèles produisent des résultats différents. L'assemblage de ces couches dans un ordre particulier permet de disséquer une image en plusieurs parties, qui peuvent ensuite être reliées les unes aux autres par les couches denses. Cela doit être fait en tenant compte de la conformité de la matrice, du coût de calcul et de la taille minimale des éléments. De même, l'organisation des connexions, des interruptions et des neurones sur les couches denses modifiera les relations extraites par le modèle global. L'impact de cette organisation des couches a été largement étudié (Alzubaidi et al., 2021;

Mallat, 2016; LeCun et al., 2015; Szegedy et al., 2014) et ce travail s'appuiera fortement sur des architectures établies et réussies pour l'application donnée (Zahavy et al., 2018).

Hyperparamètres. Les paramètres d'apprentissage jouent un rôle crucial dans l'élaboration du comportement et des résultats d'un RN au cours du processus d'apprentissage (Goodfellow et al., 2016). Ces paramètres comprennent la fonction de perte, les métriques, l'optimiseur, le taux d'apprentissage, la taille du lot et le nombre d'époques. Bien que ces paramètres ne soient généralement pas au centre des préoccupations lors de la conception d'un RN, une sélection inappropriée peut avoir un impact négatif sur les performances du modèle.

0.3 Méthodologie

0.3.1 Génération de l'ensemble de données

La création d'un ensemble de données de traces expérimentales de FROSt pose deux problèmes importants. Tout d'abord, la collecte d'un ensemble de données suffisamment important nécessiterait beaucoup de ressources. Deuxièmement, l'étiquette, qui est le champ électrique de l'impulsion, est intrinsèquement inconnue en raison des limitations décrites dans le PRP. Pour relever ces défis, des ensembles de données simulées de champs électriques et leurs traces FROSt correspondantes sont utilisés pour entraîner le RN.

Les champs électriques sont générés à l'aide d'une impulsion TFL Gaussienne avec des paramètres définis, tels que l'amplitude, l'étalement temporel et le temps central. Ces impulsions sont modifiées par l'introduction de phases spectrales aléatoires, représentées par des polynômes, qui sont appliquées dans le domaine des fréquences et retransformées dans le domaine temporel pour produire des impulsions chirpées. Les fonctions de phases spectrales aléatoires sont générées à l'aide de l'Équation 3, où R_o est un coefficient d'ordre o généré aléatoirement au sein d'une fonction polynomiale. Trois paramètres ont été modifiés pour générer une variété de phases : l'ordre, o, le nombre de termes monomiaux, N_m, et l'amplitude de chaque coefficient, R_o.

$$\varphi(\omega) = \sum_{o\,=\,[O-N_m+1]}^{O} R_o\,\omega^o \tag{3}$$

Les fonctions spectrales ont été générées avec des ordres, o, de polynômes du 1er au 4ème degré (où O est l'ordre maximum d'un polynôme donné tel que $o \in [O - N_m + 1, O]$). D'abords, le champ électrique chirpé complexe, $E_C(t)$, est représenté par l'Équation 4, où il a une amplitude de $E_{0,\,C}(t)$, un étalement de σ_C, un centre temporel de $t_0 = 0$, et une phase aléatoire de $\phi(t)$ liée à $\varphi(\omega)$. La méthodologie décrite permet de générer des impulsions quasi infinies avec une grande variation dans les comportements de phase, ce qui permet d'obtenir un ensemble de données diverses et robustes d'impulsions chirpées.

$$E_C(t) = E_{0,\,C}(t)\, e^{-0.5\left(\frac{t}{\sigma_C}\right)^2}\, e^{j\phi(t)} \tag{4}$$

Les fonctions de switch, modélisées sous forme de courbes Gaussiennes intégrées, se rapprochent du comportement de l'inversion de population dans les matériaux. Ces switches sont intégrés pour

créer des courbes unilatérales et normalisées pour s'adapter aux contraintes expérimentales. Deux scénarios sont utilisés pour la formation : l'un avec une seule switch de durée fixe et l'autre avec une gamme de durées de 100 fs à 400 fs. Bien que le RN ne soit pas entraîné sur un large ensemble de switch, la diversité des durées ajoute de la complexité au processus de récupération. Les traces FROSt, $I(\omega, \tau)$, sont créées en combinant les champs électriques simulés et les fonctions des switchs. Les traces sont calculées à l'aide de l'Équation 2 avec le switch comme fenêtre. Ce calcul est effectué efficacement à l'aide d'opérations `tensorflow` dans une architecture appelée FROStNET. Le réseau met en œuvre l'équation théorique de FROSt par le biais d'opérations matricielles, ce qui permet de générer des traces de manière précise et efficace sur le plan informatique.

Assurance qualité. Une fois qu'un grand ensemble de données a été généré à l'aide de la procédure ci-dessus, il est important de s'assurer qu'il est de haute qualité. Pour ce faire, deux considérations principales sont prises en compte : les contraintes expérimentales et la suppression des ambiguïtés. Les contraintes expérimentales sont appliquées pour s'assurer que les impulsions générées décrivent des conditions physiquement réalistes tout en restant valides sur le plan programmatique. Il s'agit notamment de contraindre trois propriétés. Le retard de groupe et la dispersion de l'énergie sont limités pour garantir que l'impulsion se trouve dans la fenêtre d'échantillonnage. Pour la contrainte de fréquence instantanée, les fonctions de phase sont mises à l'échelle pour s'assurer qu'elles sont limitées à une région de $-\pi \leq \Delta\phi(t) \leq \pi$.

Les ambiguïtés sont des cas où deux traces différentes produisent des impulsions trivialement similaires. L'inclusion de ces ambiguïtés dans l'ensemble de données conduit à une divergence important pendant la procédure d'apprentissage. Il est nécessaire que chaque trace de l'ensemble de données ait un seul champ électrique correct à approcher (c'est-à-dire que le système a un seul minimum vers lequel on converge). Un ensemble de données qui comprend de multiples traces identiques, chacune avec des étiquettes nettement différentes, entraînera l'instabilité et la divergence. Pour minimizer l'effet, les impulsions liées par un déphasage constant, un décalage temporel constant ou une inversion temporelle sont supprimées de l'ensemble de données.

0.3.2 Architecture multirésolution

Le modèle utilisé pour ce travail est directement inspiré de l'architecture utilisée par Zahavy (Zahavy et al., 2018). Cette architecture utilise des couches convolutives 2D, suivies de couches denses, toutes dotées de fonctions d'activation ReLU ($\mathrm{ReLU}(n) = \max(0, n)$) (Zahavy et al., 2018; Szegedy et al., 2014). Il est nommé multirésolution (MR) puisqu'il utilise plusieurs taille de filtres dans les convolutions.

Un seul bloc MR contient trois couches convolutives parallélisées qui produisent des cartes de caractéristiques indépendantes à partir de leur entrée. Ces cartes sont concaténées et suivies d'une couche convolutive unique qui divise par deux la taille de la carte de caractéristiques mais produit deux fois plus de cartes. Ce bloc est mis à l'échelle et sérialisé pour l'architecture globale présentée dans la Figure 3. La Figure 3 montre comment une trace FROSt d'entrée est déconvoluée par une série de blocs MR avant d'être recombinée pour extraire des approximations à la fois pour le champ électrique et le switch. Les deux sorties sont nécessaires pour garantir que l'IA émule le comportement de la ptychographie, qui peut récupérer les profils du switch et de l'impulsion.

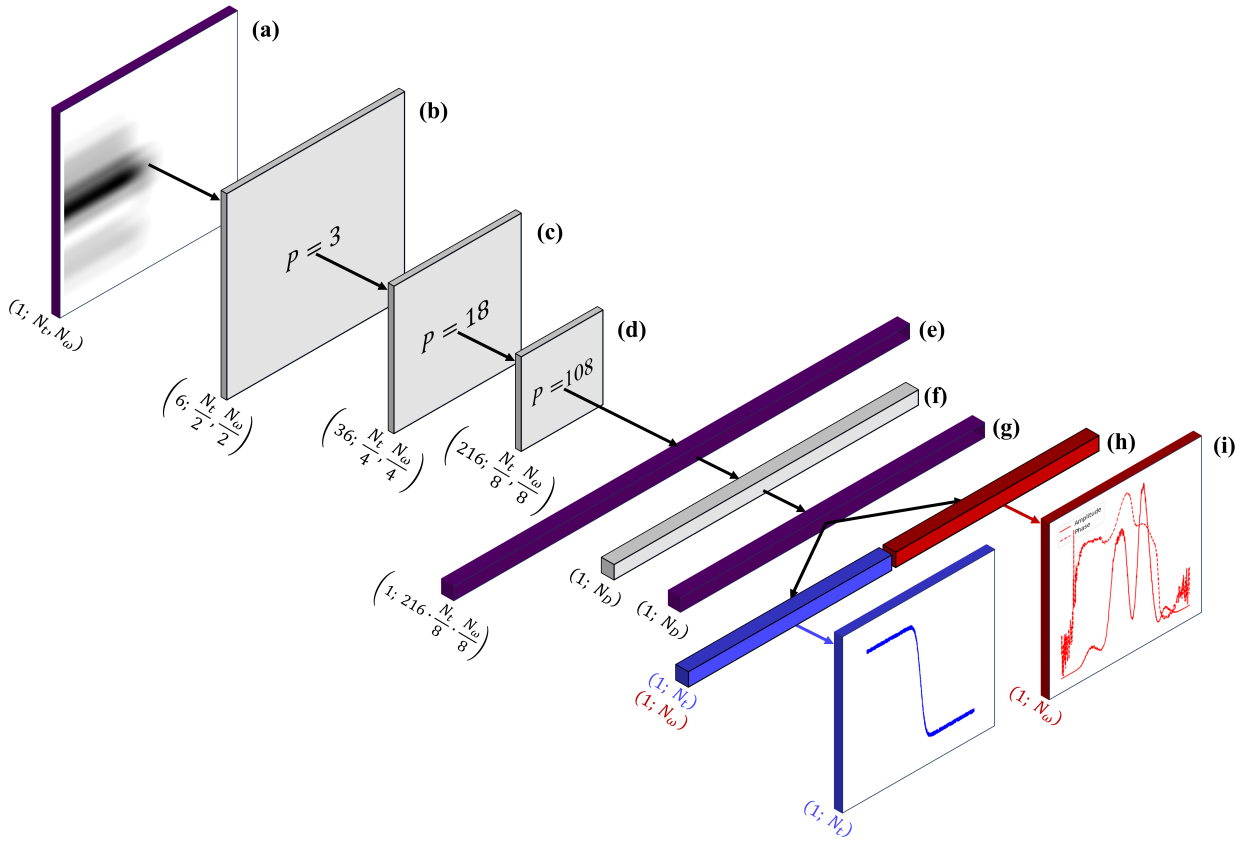

Figure 3 : Architecture multirésolution globale.

L'architecture globale, qui utilise trois blocs MR sérialisés, est présentée. Les panneaux bordés de violet représentent les entrées et les sorties des couches. Les blocs MR sont représentés en gris et sont désignés par le nombre de filtres MR, P, qu'ils contiennent. La forme de la sortie de chaque bloc MR est indiquée sous chaque couche. À l'entrée (a), une trace en noir et blanc de $(M_{(a)} = 1; N_t = 128, N_\omega = 128)$ est représentée par un panneau violet. Le premier bloc MR en (b) est exactement comme présenté dans la Figure 3.5 et produit six cartes de caractéristiques à la moitié de la taille de l'entrée : $\left(M_{(b)} = 6; N_t/2 = 64, N_\omega/2 = 64\right)$. En (c), il est transmis à un bloc MR mis à l'échelle, avec $P = 18$ filtres. La sortie de (c) est une matrice de 36 cartes de caractéristiques, chacune représentant un quart de la taille de l'entrée : $\left(M_{(c)} = 36; N_t/4 = 32, N_\omega/4 = 32\right)$. Ce bloc est suivi d'un troisième bloc MR en (d), mis à l'échelle pour un total de $P = 108$ filtres. La sortie de (d) est constituée de 216 cartes de caractéristiques, chacune représentant 1/8e de la taille de l'image d'entrée : $\left(M_{(d)} = 216; N_t/8 = 16, N_\omega/8 = 16\right)$. Ce vecteur est aplati pour produire le vecteur violet de la forme $\left(M_{(e)} = 1; 216 \times N_t/8 \times N_\omega/8\right)$ en (e). En (f), il est transmis à la première couche dense avec $\left(M_{(f)} = 1; N_D\right)$ neurones, qui produit le vecteur violet de $\left(M_{(g)} = 1; N_D\right)$ en (g). Ce vecteur est décomposé par les couches denses parallèles finales, en (h), qui produisent deux vecteurs de $\left(M_{(h)} = 1; N_t = 128\right)$ et $\left(M_{(h)} = 1; N_\omega = 128\right)$ points chacun. Ces vecteurs sont des approximations pour l'interrupteur (en bleu) et l'impulsion (en rouge), comme le montrent les panneaux de sortie (i). Ce modèle a été généré à l'aide de `tensorflow` et est inclus dans la Figure 7.6.

0.3.3 Procédure de formation

La méthodologie de formation a utilisé une procédure inspirée de Zahavy (Zahavy et al., 2018). L'ensemble de données se compose de $N = 2^{16}$ étiquettes et images appariées, où chaque étiquette comprend deux vecteurs : le champ électrique, $E_{P,\,L}(t)$, et le switch, $S_L(t-\tau)$, qui sont utilisés pour générer la trace simulée, $I_L(\omega, \tau)$.

Tout d'abord, une procédure d'apprentissage supervisé (AS) est testée. Les erreurs, ∇E_P et ∇S, ont été calculées à l'aide d'une fonction de perte d'erreur absolue moyenne (EAM), et la rétropropagation a optimisé les poids du réseau de manière itérative. L'ajout de bruit blanc Gaussienne (BBG) aux données d'apprentissage a permis d'évaluer la robustesse du modèle. Des cas d'essai ont été utilisés pour explorer divers résultats et fonctions de perte, notamment la phase spectrale $\varphi(\omega)$, le spectre $\tilde{E}(\omega)$, et le champ électrique $E(t)$, certains cas appliquant une contrainte expérimentale similaire à la ptychographie pour améliorer la prédiction de la phase.

L'apprentissage non-supervisé (ANS) a étendu l'AS en calculant une erreur sur la trace FROSt par l'intermédiaire de FROStNET. Les besoins mettent en œuvre la relation mathématique entre les erreurs des champs électriques, du switch et de la trace. L'effet du BBG ajouté et de divers ensembles de données de formation a été testé sur la procédure ANS. L'ensemble du processus a été optimisé à l'aide d'un objectif `Optuna`, conçu pour trouver les hyperparamètres qui minimisent l'erreur de champ électrique experimentale. Cette approche globale a permis d'équilibrer les stratégies AS et ANS afin d'obtenir une reconstruction précise de l'amplitude et la phase du champ électrique temporel.

0.3.4 Validation expérimentale

Les RNs entraînés sont évalués par rapport à la méthode standard de récupération ptychographique décrite dans la Section 0.2. Cette mise en œuvre, suivant Leblanc (2016), est intégrée dans PtyChoPy, une interface utilisateur combinant Python et C pour les algorithmes itératifs.

PtyChoPy comprend des étapes de prétraitement telles que le découpage temporel, l'élimination de l'arrière-plan et des valeurs aberrantes, le filtrage spectral et temporel et l'interpolation cubique pour préparer les traces à la récupération. La récupération utilise un hybride de méthodes ptychographiques, alternant entre les algorithmes de réduction d'erreur et d'entrée-sortie hybride pour une convergence optimisée (Leblanc, 2016). Un ensemble de données expérimentales de 77 traces, avec un rapport signal-bruit (RSB) moyen de 28.93 dB, a été traité à l'aide de PtyChoPy. La récupération ptychographique, traitée comme l'étiquette pour les traces experimentales, a atteint une erreur moyenne de 3,55 %.

0.4 Résultats

Nous démontrons le succès général de la méthodologie en montrant une prédiction expérimentale réussie dans la Figure 4. La prédiction de la Figure 4 présente une erreur de 7,88 %, 9,74 %, et 46,45 % pour le switch, le champ électrique, et la trace, respectivement. Bien que la trace non traitée soit fournie, on peut voir que le RN récupère presque exactement l'enveloppe temporelle. Cinq autres cas de traces expérimentales sont présentés à l'annexe B. Ils présentent des schémas similaires à

celui illustré ici, où le champ électrique des impulsions présente une forte concordance alors que le profil du switch ne l'est pas.

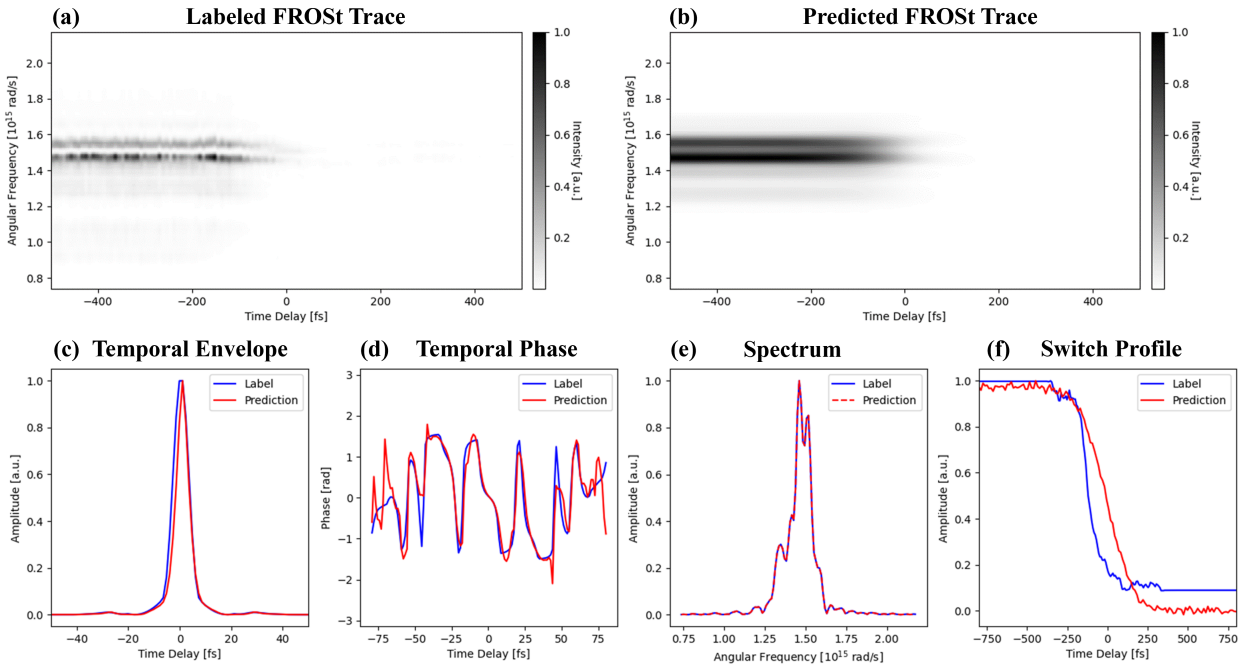

Figure 4 : Prédiction IA générée sur une trace expérimentale.

Une trace expérimentale brute de FROSt est présentée en (a). Cette trace n'a subi aucun filtrage ou prétraitement avant la génération d'une prédiction. La trace reconstruite, générée à partir des prédictions du RN, est représentée en (b). Les vecteurs étiquetés (en bleu) et prédits (en rouge) pour l'enveloppe et la phase de l'impulsion sont représentés respectivement en (c) et (d). Le spectre du champ électrique étiqueté et prédit est présenté en (e). L'amplitude temporelle du switch étiqueté et prédit est présentée en (f). Les deux switches sont normalisés conjointement selon leurs minima et maxima communs. Dans ce cas, les erreurs de reconstruction du switch, de l'impulsion, et de la trace sont respectivement de 7,88 %, 9,74 %, et 46,45 %.

0.4.1 Optimization de DFROStNET

L'ajout du BBG et la procédure ANS ont été évalués sur le cas d'essai le plus réussi L1-XP-W sur trois ensembles de données. Ceux-ci sont les ensembles de données à switch unique (SU) et à switch multiples (SM), tandis que le troisième est un ensemble de données augmenté (AUG) généré à partir de prédictions. Le meilleur cas a ensuite été optimisé grâce à l'étude `Optuna`. Nous comparons le RN optimisé aux autres variantes du cas d'essai L1-XP-W, comme le montre la Figure 5. Le cas DFROStNET est la version optimizé de C-SM. Après 50 essais de l'étude `Optuna`, les paramètres du Tableau 1 ont été déterminés pour produire le meilleur RN entraîné.

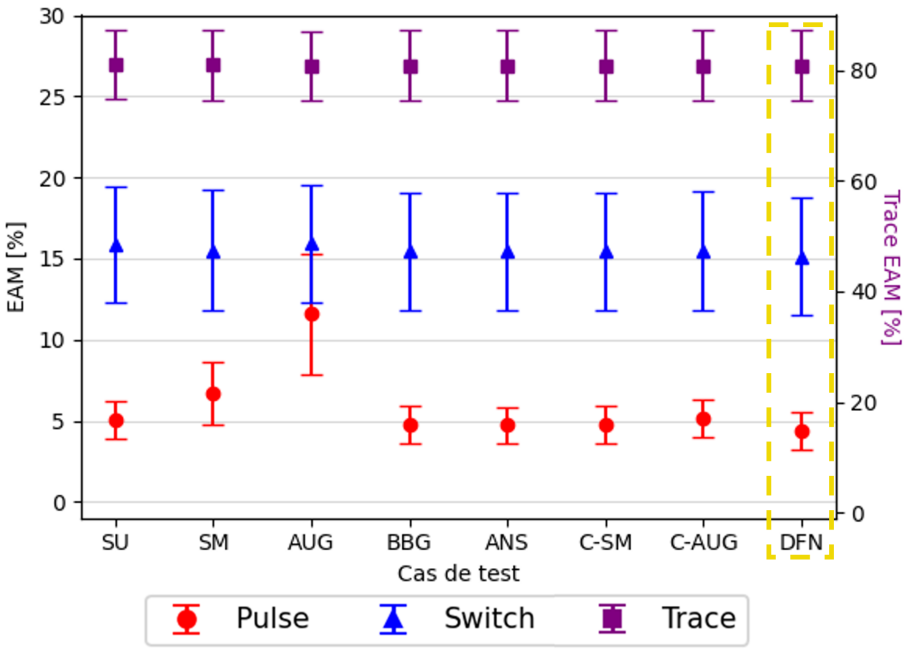

Figure 5 : Comparaison des variantes de L1-XP-W.

L'erreur de validation pour l'impulsion (en rouge, cercles), du switch (en bleu, triangles), et la trace (en violet, carrés) est montrée pour les quatre cas : RN formé sur un ensemble de données à switch unique (SU), un ensemble de données à switchs multiples (SM), l'ensemble de données augmenté (AUG), l'ensemble de données avec 3.5 dB de BBG (BBG), la procédure ANS avec une hybridation, Λ de 0,45 (ANS), ainsi que la combinaison de BBG et ANS sur les ensembles de données complexes (identifiés comme C-SM et C-AUG). La formation optimisée, DFROStNET (DFN), est représentée, mise en évidence par la boîte dorée en pointillés. Les barres d'erreur représentent le EAM sur l'ensemble de validation pour chaque métrique.

Table 1 : Paramètres optimisés par l'étude `Optuna`.

Le code de cet objectif a été inclus dans la Figure 7.9. Meilleur cas montre les paramètres de formation utilisés pour la procédure de formation optimisée, nommée DFROStNET.

Paramètre	Min	Max	DFROStNET
Taux d'abandon	**10 %**	**50 %**	**21 %**
Taux d'apprentissage	$1e-5$	$1e-1$	$1.78e-3$
Tailles de lot	**16**	**128**	**32**
Neurones denses, N_D	**256**	**2048**	**575**
Couches denses	**1**	**5**	**5**
BBG	5 dB	15 dB	7 dB
Hybridation	**0.1**	**0.5**	**0.17**

0.5 Discussion

Nous commençons par examiner les cas de test étudiés dans la Section 0.4. Nous nous concentrons tout d'abord sur les résultats de la formation à une seule switch. L'utilisation de la phase spectrale enveloppée (cas de test L1-XP-W où la phase est limitée entre $[-\pi, \pi]$) réduit considérablement les erreurs par rapport à la phase non enveloppée (L1-XP, où la phase est entre $(-\infty, \infty)$). La phase enveloppée atténue l'ambiguïté découlant des décalages de phase par des multiples de π, que les phases non enveloppées ne parviennent pas à traiter, ce qui entraîne des valeurs de perte plus élevées. Cet ajustement s'est avéré crucial pour la précision de l'entraînement et de la récupération. Il est également évident que l'application de la contrainte expérimentale (CXP) a un impact positif significatif sur les prédictions. La CXP a notamment permis de réduire les erreurs de reconstruction et d'améliorer la précision sur l'ensemble des données. Par exemple, les erreurs sont passées de 80 % à 20 % pour L2 et L3 avec la CXP. Si l'on compare les variables de perte, L1-XP-W et L3-XP ont surpassé L2, en produisant des solutions plus fiables et moins erronées, en particulier pour la récupération du champ électrique. Étant donné que L1-XP-W nécessite moins de calculs pendant la procédure d'apprentissage tout en restant aussi compétitif que L3-XP, il est privilégié.

Nous examinons maintenant l'impact des différents ensembles de données. L'entraînement sur un ensemble de données à un seul switch a entraîné une mauvaise reconstruction des switch en raison d'une diversité limitée, bien que la récupération des impulsions soit restée robuste. En revanche, l'ensemble de données SM (300 switch) a modifié les comportements d'entraînement. Alors que certains cas de test (par exemple, L1-XP) ont montré une meilleure récupération du champ électrique, d'autres (par exemple, L3-XP) ont vu leurs performances réduites, ce qui suggère une sensibilité à la variabilité du profil du switch. L'ensemble de données augmenté, combinant les données simulées avec les prédictions générées par le RN, a augmenté la taille de l'ensemble de données de manière significative mais a introduit des risques d'effondrement de mode (Shumailov et al., 2024). Les prédictions de switch dégradées issues de la formation augmentée s'écartent des comportements de switch naturels, ce qui affecte la précision. En dépit d'une génération prudente visant à éviter l'accumulation d'erreurs, la validation expérimentale a montré des augmentations mineures de l'erreur, ce qui indique que les avantages de l'augmentation sont limités. Le surajustement et l'effondrement de mode ont été notés comme des défis importants pour les ensembles de données augmentés.

Dans l'ensemble, les cas d'essai préliminaires ont donné des résultats prometteurs, tant pour les traces simulées que pour ceux expérimentales. L1-XP-W a été isolé comme étant le cas d'essai le plus précis et le moins gourmand en ressources. Il a été étudié plus en détail en appliquant le BBG et la procédure ANS, qui vont maintenant être examinés. L'examen des résultats de la procédure L1-XP-W entraînée par un seul switch dans la Figure 5 (SU) montre que la méthodologie proposée a permis de récupérer l'amplitude temporelle et la phase de l'impulsion avec un minimum d'erreur de 5,02 %. Cependant, d'autres améliorations ont été obtenues grâce à l'application de techniques utilisées pour réduire l'écart entre la simulation et la réalité.

Nous commençons par examiner les effets du BBG ajouté. L'erreur du champ électrique augmente rapidement d'un minimum de 4,73 % à un maximum de 10,46 % sur une plage allant entre 3.42 et 20 dB. De même, l'erreur de récupération du switch augmente de 15,44 % à 16,47 % sur presque la même plage. Fait intéressant, les deux erreurs de récupération diminuent à nouveau après cette région. Il est important de noter que le minimum a surpassé le minimum de formation sans bruit de 5,02 %. Bien que l'ajout de BBG à la formation d'un switch unique ait eu peu d'effet, il a eu un fort impact positif sur les procédures de formation avec les données SM et AUG (comme

le montre la Figure 5). Globalement, l'ajout de BBG a amélioré les formations SM et AUG, mais n'a pas significativement amélioré les prédictions de switch.

Nous examinons maintenant la procédure ANS. De manière similaire au BBG, il n'y a eu aucun impact sur les prédictions simulées. Fait intéressant, l'erreur du champ électrique approche d'un minimum de 4,72 % en utilisant ANS, ce qui surpasse la procédure AS. De cela, nous pouvons voir que l'hybridation des procédures produit une procédure de formation améliorée. Cela a été validé davantage dans la Figure 5, où le RN formé sur des données complexes avec les procédures combinées BBG et ANS (C-MS et C-AUG) a donné des performances équivalentes avec les données à switch unique. Une dernière étape d'optimisation a été réalisée en utilisant l'objectif `Optuna` pour produire le meilleur cas présenté dans le Tableau 1. Les erreurs minimales de 15,11 %, 4,39 %, et 80,92 % pour le switch, le champ électrique, et la trace FROSt sont obtenues avec les paramètres du meilleur cas.

0.5.1 Switch : Un autre problème de récupération de phase

DFROStNET montre une faible erreur dans la récupération du champ électrique, ce qui le rend adapté aux applications prévues. Cependant, il rencontre des difficultés avec la récupération du switch, générant des erreurs supérieures à 15 %, ce qui entraîne une erreur importante sur la trace récupéré. Cela est attribué à la variation limitée des switches dans les données. Bien que DFROStNET atteigne l'objectif principal de récupération du champ électrique, son incapacité à récupérer correctement le switch l'empêche de répliquer pleinement la fonctionnalité de la ptychographie. En ptychographie, récupérer à la fois le champ électrique et le switch permet de valider les résultats en comparant la trace reconstruite à la trace expérimentale, une étape qui est absente dans DFROStNET.

Cette limitation peut être partiellement atténuée en intégrant DFROStNET avec la contrainte ptychographique. En appliquant cette contrainte, le profil du switch peut être dérivé en utilisant le même processus que dans la Section 0.2, bien que le résultat représente l'une des nombreuses solutions possibles en raison de l'absence d'informations complètes sur la phase. Malgré cette imperfection, la méthode permet de récupérer l'intensité du switch, ce qui suffit pour les applications actuelles. La contrainte ptychographique améliore les capacités de DFROStNET, le rendant plus comparable à la ptychographie en récupérant l'intensité du switch.

Bien que la ptychographie bénéficie toujours d'une étape de validation grâce à son processus de double récupération, DFROStNET offre des prédictions fiables du champ électrique en temps réel avec un traitement minimal. La Figure 6 illustre les différences : la ptychographie nécessite un prétraitement extensif et des calculs itératifs, tandis que DFROStNET prédit directement le champ électrique et approxime le switch en utilisant la trace expérimentale et les prédictions de l'IA. Bien que la ptychographie reste supérieure en validation et exhaustivité, DFROStNET offre rapidité et simplicité, présentant des avantages uniques dans des scénarios expérimentaux spécifiques.

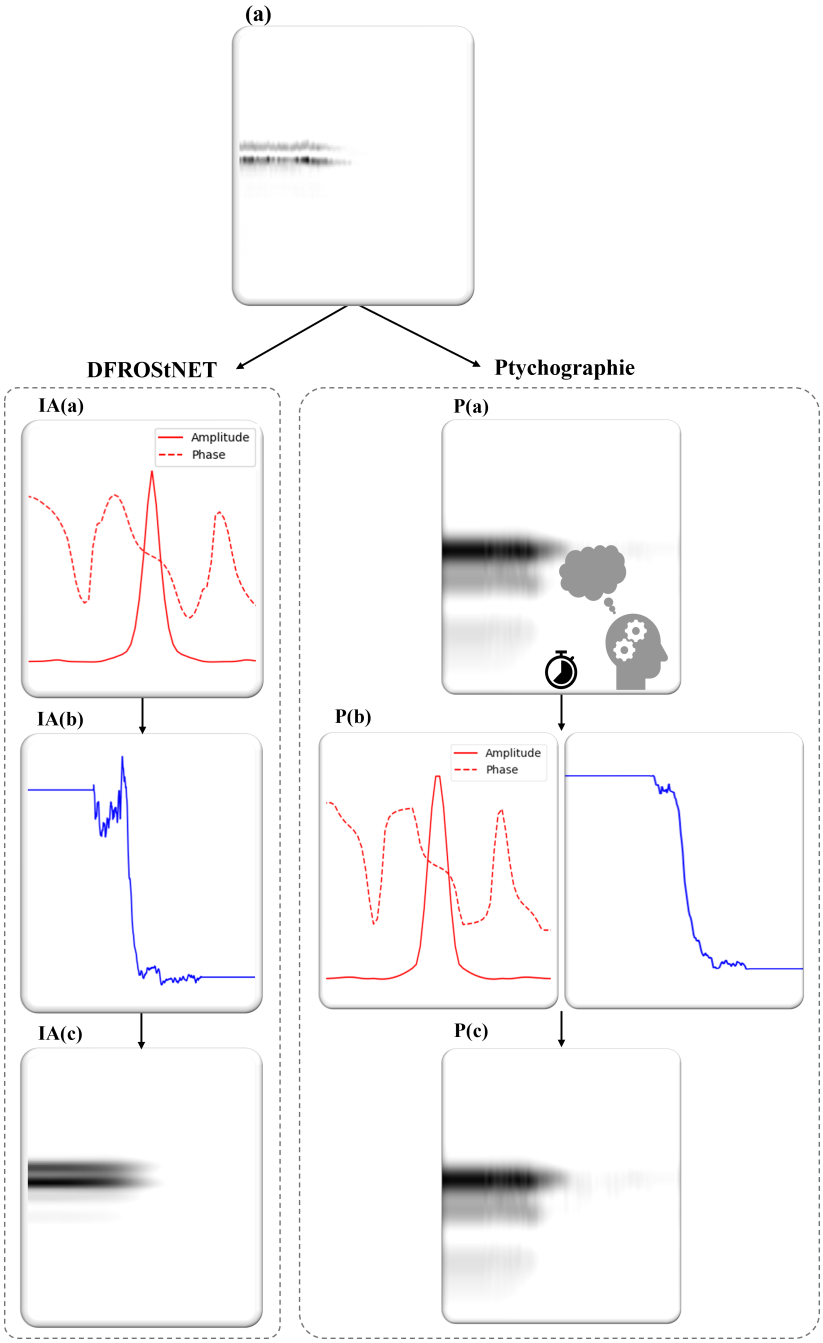

Figure 6 : Comparaison des processus DFROStNET et ptychographique.

Les procédures utilisées pour l'IA (IA(a) - (c)) et pour le ptychographique (P(a) - (c)) sont montrées. En (a), la trace expérimentale brute de FROSt est présentée. Pour la procédure IA, la trace est directement entrée dans DFROStNET, qui produit une prédiction fiable du champ électrique, comme montré en IA(a). En appliquant la contrainte ptychographique, le switch en IA(b) est calculé. À partir du champ électrique et du switch, la trace FROSt récupérée en IA(c) est générée à l'aide de FROStNET. Parallèlement, pour le processus ptychographique, la trace expérimentale subit un prétraitement pour récupérer la trace traitée en P(a). Une intervention humaine et du temps sont nécessaires pour obtenir cette trace traitée. Celle-ci est entrée dans la récupération ptychographique qui, après un certain temps, génère le champ électrique et le switch en P(b). Ceux-ci peuvent être recombinés à l'aide de FROStNET pour générer la trace FROSt récupérée en P(c). Les deux processus sont comparés pour montrer les étapes et les entrées supplémentaires nécessaires pour la méthode ptychographique.

0.6 Conclusions et prochaines étapes

Ce travail a permis de développer avec succès une méthodologie permettant à un RN, nommée DFROStNET, de récupérer des informations complex sur le champ électrique à partir d'une trace FROSt, atteignant une erreur de récupération de seulement 4,39 %. Bien que cette valeur soit relativement élevée comparativement aux erreurs typiquement rapportées pour les reconstructions GSH FROG (Zahavy et al., 2018), elle demeure cohérente avec les performances habituellement observées pour les approches FROSt (Bourassin-Bouchet et al., 2013; Leblanc et al., 2019, 2021). Par ailleurs, cette erreur est obtenue en temps réel, sans étapes de prétraitement, ce qui constitue un compromis favorable entre précision et rapidité. La méthodologie a démontré sa capacité à reproduire fidèlement les champs électriques simulés, en exploitant une architecture multi-résolution pour déconvoluer les traces FROSt. Une procédure d'entraînement robuste, fondée sur un apprentissage non supervisé, permet de maintenir une erreur faible tout en s'adaptant à des ensembles de données plus complexes. Enfin, DFROStNET offre des avantages distincts par rapport aux méthodes ptychographiques classiques, notamment des prédictions en temps réel et une indépendance vis-à-vis des biais de l'opérateur, positionnant cette approche comme une alternative rapide, impartiale et fiable pour la récupération du champ électrique à partir de mesures FROSt.

La méthodologie a efficacement imité les champs électriques réels, en exploitant une architecture multi-résolution pour déconvoluer les traces FROSt. Une procédure d'entraînement robuste qui implémente l'apprentissage non supervisé peut maintenir cette faible erreur tout en s'entraînant sur des ensembles de données plus complexes. De plus, DFROStNET offre des avantages uniques par rapport à la ptychographie, notamment des prédictions en temps réel sans étapes de prétraitement, éliminant ainsi le biais du chercheur et l'intervention humaine. Cela le rend pratique pour l'acquisition simultanée de données et la récupération du champ électrique, positionnant DFROStNET comme une alternative rapide, impartiale et fiable, comparatif aux méthodes ptychographiques.

DFROStNET montre un potentiel prometteur pour la récupération en temps réel du champ électrique d'une impulsion à partir de la trace FROSt non traitée, ce qui est déjà précieux pour les applications expérimentales. Bien que DFROStNET offre des avantages considérables par rapport aux méthodes traditionnelles, plusieurs axes d'amélioration et de développement futur demeurent. Deux domaines principaux sont brièvement abordés : l'amélioration de l'ensemble de données des switches et l'implémentation expérimentale.

0.6.1 Ensemble de données de switch

Une limitation clé est la précision de la récupération du switch, car DFROStNET, entraîné uniquement sur des données de champ électrique, peine à récupérer les profils de switch. Pour y remédier, un ensemble de données plus diversifié de switches, reflétant les dynamiques matérielles réalistes, est nécessaire. Ces données devraient modéliser les complexités de l'inversion de population, influencée par des variables expérimentales telles que l'intensité de la pompe, le type de matériau et l'épaisseur (Satou et al., 2011; Amstrup et al., 1993; Gornushkin et al., 2017). Générer ces données de switches de haute qualité, potentiellement à l'échelle de 65 536 switches uniques, permettrait un processus de récupération plus robuste. De plus, l'utilisation des réseaux antagonistes génératifss (RAGs) pourraient améliorer davantage la procédure d'entraînement. Les RAGs, composés d'un générateur et d'un discriminateur travaillant ensemble, pourraient remplacer les données simulés par une génération de données plus réaliste (Goodfellow et al., 2016). Cette approche

pourrait améliorer la robustesse et la stabilité du système par rapport à la méthode actuelle des données augmenté. Cependant, l'entraînement sur des données exponentiellement plus grands, tels que la combinaison de switches et de champs électriques, augmenterait considérablement les exigences computationnelles. Les temps d'entraînement pourraient grimper à des heures, nécessitant potentiellement l'utilisation de processeurs graphiques ou de processeurs tensoriels avancés pour un calcul efficace. Malgré ces défis, les améliorations dans la génération de données et les ressources computationnelles pourraient ouvrir la voie à un affinage et une évolutivité futurs de DFROStNET.

0.6.2 Implémentation expérimentale

Pour implémenter DFROStNET expérimentalement, plusieurs étapes doivent être franchies, notamment l'intégration dans une interface utilisateur et des améliorations potentielles des méthodologies expérimentales.

La première étape de l'implémentation consiste à l'intégrer dans la configuration existante de collecte de données expérimentales. Cela implique l'intégration du RN dans le processus de collecte de données, permettant la récupération en temps réel de l'impulsion et de l'approximation du switch directement à partir de la trace FROSt acquise, comme illustré dans le flux de travail expérimental dans Figure 6. Cette intégration élimine le besoin de prétraitement intensif, rationalisant le processus pour des prédictions immédiates en temps réel. Les chercheurs bénéficieraient de retours instantanés, leur permettant d'ajuster les paramètres expérimentaux, à l'instar des systèmes de récupération FROG existants (DeLong, 2009), ce qui améliorerait l'efficacité globale de la procédure expérimentale. Une fois cela réalisé, associer l'interface utilisateur avec l'acquisition de données en un seul tir (Brizard et al., 2020; Louisy et al., 2017; Kane et al., 1993) créerait une configuration expérimentale extrêmement puissante. La technique de FROSt en un seul tir a été réalisée dans Brizard et al. (2020) et permet une caractérisation plus rapide et plus précise des systèmes optiques. L'intégration de l'acquisition de FROSt en un seul tir avec la récupération en temps réel permettrait une caractérisation immédiate et complète du système optique. Cela permettrait de positionner davantage FROSt comme une alternative compétitive à FROG, en particulier dans les cas impliquant des impulsions de faible intensité ou fortement chirpées.

Récupération expérimentale du switch. Avec la configuration expérimentale proposée, nous pouvons théoriser une méthodologie expérimentale pour récupérer le profil complexe exact du switch. Nous considérons le cas où à la fois une sonde et un switch sont inconnus. Deux traces FROSt sont acquises à partir de ce système. La première est réalisée comme nous exécutons traditionnellement FROSt. Pour la deuxième, une dispersion constante inconnue est ajoutée au système avec un milieu dispersif (par exemple, du verre). À partir de ces deux mesures, nous acquérons la trace, $I_1(\omega,\tau)$, et sa partenaire modulée, $I_2(\omega,\tau)$. Les deux sont des fonctions de la sonde, $E_P(t)$, et du switch, $S(t-\tau)$, tandis que seul $I_2(\omega,\tau)$ dépend du milieu dispersif, $\tilde{D}(\omega)$.

$$I_1(\omega,\tau) = \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt \right|^2$$
$$I_2(\omega,\tau) = \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt\, \tilde{D}(\omega) \right|^2$$

À partir de ces acquisitions, nous créons un système d'équations qui peut être résolu pour identifier le profil complexe du switch. Nous pouvons utiliser le système d'équations pour résoudre à la fois pour le milieu dispersif, $\tilde{D}(\omega)$, et le switch, $S(t-\tau)$, en utilisant les arrangements ci-dessous.

$$\tilde{D}(\omega) = \left(\frac{I_2(\omega)}{I_1(\omega)}\right)^{1/2}$$

$$I_2(\omega,\tau) - I_1(\omega,\tau) = \left|\int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt\, \tilde{D}(\omega)\right|^2 - \left|\int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt\right|^2$$

Bien que nous puissions résoudre directement pour le milieu dispersif, nous pouvons mettre en œuvre une récupération itérative simple pour minimiser la deuxième équation. Le profil du switch qui atteint ce minimum est alors considéré comme la solution analytique du switch complexe. Parce que nous avons ajouté une redondance supplémentaire (c'est-à-dire l'ajout de la fonction de dispersion, $\tilde{D}(\omega)$, dans l'une des deux traces), nous pouvons récupérer le profil complexe du switch, tout comme dans la ptychographie.

Nous formulons l'hypothèse que cette procédure expérimentale permettrait de récupérer de manière adéquate et précise le profil complexe du switch. La principale limitation de ce processus proposé est le temps nécessaire pour effectuer deux scans FROSt, qui pourrait être minimisé avec l'intégration de la technique de FROSt en un seul tir mentionnée ci-dessus (Brizard et al., 2020). La détection en un seul tir rationaliserait la collecte des traces FROSt, tandis que la combinaison de DFROStNET et de cette simple récupération numérique permettrait la récupération précise et en temps réel à la fois de l'impulsion et du switch.

L'intégration de DFROStNET et du processus de récupération du switch fournit un outil avancé pour étudier les dynamiques des matériaux, en particulier l'inversion de population induite par des vaiseaux de haute intensité. FROSt permet aux chercheurs de caractériser les dynamiques de divers matériaux sous des conditions de photo-excitation en analysant le profil du switch récupéré. Cette méthode offre une approche plus fiable et reproductible que la ptychographie, minimisant la variabilité et améliorant la précision des études matérielles. La nouvelle approche pourrait révolutionner des domaines tels que la caractérisation des lasers ultrarapides et la science des matériaux en permettant des expériences plus précises et accessibles.

En conclusion, l'implémentation expérimentale de DFROStNET, associée aux méthodologies proposées, représente une amélioration significative dans la caractérisation en temps réel des systèmes optiques. En combinant des techniques de récupération avancées avec des configurations expérimentales améliorées, ce travail ouvre la voie à des percées dans la caractérisation des lasers ultrarapides, les dynamiques des matériaux et l'application de l'IA dans ces domaines.

TABLE OF CONTENTS

ACKNOWLEDGMENTS ..... **iii**

ABSTRACT ..... **v**

RÉSUMÉ ..... **vii**

SYNOPSIS ..... **ix**

0.1 Introduction ix

0.1.1 Techniques de characterization x

0.1.2 Frequency-resolved optical switching xi

0.1.3 Intelligence artificielle xii

0.2 Théorie xiii

0.2.1 Le problème de la récupération de la phase xiii

0.2.2 Ptychographie xiv

0.2.3 Intelligence artificielle xv

0.3 Méthodologie xvii

0.3.1 Génération de l'ensemble de données xvii

0.3.2 Architecture multirésolution xviii

0.3.3 Procédure de formation xx

0.3.4 Validation expérimentale xx

0.4 Résultats xx

0.4.1 Optimization de DFROStNET xxi

0.5 Discussion xxiii

0.5.1 Switch : Un autre problème de récupération de phase xxiv

0.6 Conclusions et prochaines étapes xxvi

0.6.1 Ensemble de données de switch xxvi

0.6.2 Implémentation expérimentale xxvii

TABLE OF CONTENTS ..... **xxix**

LIST OF FIGURES ..... **xxxi**

LIST OF TABLES ..... **xxxv**

LISTE D'ACRONYMES FRANÇAIS ..... **xxxvii**

LIST OF ENGLISH ACRONYMS ..... **xxxix**

1 INTRODUCTION 1
1.1 General Background 1
1.1.1 Sampling Theory 3
1.2 Characterization Techniques 4
1.2.1 Characterization in the Time Domain 4
1.2.2 Characterization in the Spectral Domain 7
1.3 Frequency-Resolved Optical Switching 10
1.4 Artificial Intelligence 14
1.4.1 Learning Procedures 15
1.5 In this work 19
2 THEORY 21
2.1 The Phase Retrieval Problem 21
2.2 Ptychography 25
2.2.1 Spatial and Temporal Ptychography 25
2.2.2 Numerical Recovery Procedure 28
2.3 Artificial Intelligence 32
2.3.1 Building a Neural Network 32
2.3.2 Training Parameters 36
3 METHODOLOGY 39
3.1 Dataset Generation 39
3.1.1 Random Phase Generation 42
3.1.2 Dataset Quality Assurance 43
3.2 Multi-Resolution DNNs 46
3.3 Training Procedure 48
3.3.1 Supervised Learning Procedure 49
3.3.2 Unsupervised Learning Procedure 51
3.4 Experimental Validation 52
4 RESULTS 55
4.1 Example Predictions 55
4.2 Test Cases 57
4.3 Parameter Optimization 60
4.3.1 White Gaussian Noise 60
4.3.2 Unsupervised Learning Hybridization 61

4.3.3 Hyperparameters 62
5 DISCUSSION **65**
5.1 Evaluating DFROStNET 65
5.1.1 Test Cases 65
5.1.2 Training Datasets 66
5.2 Overcoming the Sim-to-Real Gap 69
5.3 Switch Recovery : Another Phase Retrieval Problem 70
6 CONCLUSIONS AND NEXT STEPS **75**
6.1 In This Work : DFROStNET 75
6.2 Developing Training Datasets 76
6.3 Experimental Implementations 77
BIBLIOGRAPHY **81**
7 APPENDICES **87**
7.1 Appendix A : Codes 87
7.1.1 Data Generation 87
7.1.2 Architecture 88
7.1.3 Training 90
7.1.4 Ptychography 93
7.2 Appendix B : Further Predictions 95

LIST OF FIGURES

Figure 1 Théorie Principal de FROSt. xi
Figure 2 Processus de la récupération ptychographique. xv
Figure 3 Architecture multirésolution globale. xix
Figure 4 Prédiction IA générée sur une trace expérimentale. xxi
Figure 5 Comparaison des variantes de L1-XP-W. xxii
Figure 6 Comparaison des processus DFROStNET et ptychographique. xxv

Figure 1.1 General Experimental Pump-Probe Set-up. 3
Figure 1.2 Experimental Michelson Interferometer Set-up. 5
Figure 1.3 Experimental Electro-Optical Sampling Set-up. 7
Figure 1.4 Experimental SHG FROG Set-up. 9
Figure 1.5 Experimental FROSt Set-up. 11
Figure 1.6 Theoretical Principle of FROSt. 12
Figure 1.7 Feature Maps of a Dog. 16
Figure 1.8 Feature Maps of a Dog, a Cat, and a Cup. 17

Figure 2.1 STFT Spectrograms Demonstrating the Uncertainty Principle. 23
Figure 2.2 Visualization of Spatial Ptychography. 25
Figure 2.3 Iterative Ptychographic Recovery Process. 28
Figure 2.4 Serialized Convolutions of a FROSt TFR. 34

Figure 3.1 FROStNET Architecture. 41
Figure 3.2 Temporal and Spectral Profiles of an FTL and a Chirped Pulse. 42
Figure 3.3 Electric Fields Exhibiting Monomial Phases. 43
Figure 3.4 Examples of Time-Space Ambiguities. 45
Figure 3.5 Multi-Resolution Block Architecture. 47
Figure 3.6 Global Multi-Resolution Architecture. 48
Figure 3.7 Supervised Learning Training Procedure. 49
Figure 3.8 Unsupervised Learning Training Procedure. 51
Figure 3.9 PtyChoPy's Inputs and Outputs. 54

Figure 4.1 Prediction Generated on Simulated Data. 56

Figure 4.2 Prediction Generated on Experimental Trace. 57
Figure 4.3 Evaluation of Test Cases on Validation Sets. 59
Figure 4.4 Evaluation of WGN on Training Validation Sets. 61
Figure 4.5 Evaluation of UL on Training Validation Sets. 62
Figure 4.6 Comparing Variants of L1-XP-W on Experimental Validation Set. 63

Figure 5.1 Degenerating Switch Predictions. 68
Figure 5.2 Comparison of DFROStNET and Ptychographic Processes. 73

Figure 7.1 Generating an FTL Pulse. 87
Figure 7.2 Applying Spectral Phase to FTL Pulse. 88
Figure 7.3 Generating the Switch Function. 88
Figure 7.4 FROStNET Architecture. 89
Figure 7.5 Multi-Resolution Block Architecture. 89
Figure 7.6 Global Multi-Resolution Model Architecture. 90
Figure 7.7 Training Loop. 91
Figure 7.8 Experimental Validation. 92
Figure 7.9 `Optuna` Objective. 92
Figure 7.10 Trace Pre-Processing. 93
Figure 7.11 Application of Ptychographic Constraint. 94
Figure 7.12 Example 1 of Experimental Prediction by DFROStNET. 95
Figure 7.13 Example 2 of Experimental Prediction by DFROStNET. 96
Figure 7.14 Example 3 of Experimental Prediction by DFROStNET. 97
Figure 7.15 Example 4 of Experimental Prediction by DFROStNET. 98
Figure 7.16 Example 5 of Experimental Prediction by DFROStNET. 99

LIST OF TABLES

Table 1 Paramètres optimisés par l'étude Optuna. xxii

Table 3.1 Training Test Cases. 51
Table 3.2 Hyperparameters for Optuna Study. 52

Table 4.1 Training Time Per Epoch. 58
Table 4.2 Optimized Parameters from Optuna Study. 63

LISTE D'ACRONYMES FRANÇAIS

ANS	apprentissage non-supervisé
AS	apprentissage supervisé
AUG	augmenté
BBG	bruit blanc Gaussienne
CXP	contrainte expérimentale
GSH	génération de la seconde harmonique
IA	intelligence artificielle
MR	multirésolution
PRP	problème de la récupération de phase
PTF	produit temp-fréquence
RAG	réseaux antagonistes génératifs
ReLU	unité linéaire rectifiée
RN	réseaux de neurones
RSB	rapport signal-bruit
RTF	représentation temps-fréquence
SM	switch multiples
SU	switch unique
TF	transformée de Fourier
TFL	transformée de Fourier Limitée

LIST OF ENGLISH ACRONYMS

AI	artificial intelligence
AUG	augmented
CCD	charge-coupled device
CRAB	Complete Reconstruction of Attosecond Bursts
d-scan	dispersive scanning
DNN	deep neural network
EAM	erreur absolue moyenne
EOS	electro-optical sampling
ER	error reduction
FROG	Frequency-Resolved Optical Gating
FROSt	Frequency-Resolved Optical Switching
FT	Fourier transform
FTL	Fourier Transform-Limited
FWHM	full-width half-maximum
GAN	generative adversarial network
Ge	germanium
GPU	graphical processing unit
GRENOUILLE	GRating-Eliminated No-nonsense Observation of Ultrafast Incident Laser Light E-Field
HIO	hybrid input-output
MAE	mean absolute error
MR	multi-resolution
MS	multi-switch
NN	neural network
PRP	phase retrieval problem
ReLU	rectified linear unit
SHG	second harmonic generation
Si	silicon
SL	supervised learning
SNR	signal-to-noise ratio
SPIDER	Spectral Phase Interferometry for Direct Electric-field Reconstruction
SS	single-switch
STFT	Short-Time Fourier Transform
TADPOLE	Temporal Analysis by Dispersing a Pair Of Light E-fields
TBP	time-bandwidth product
TFR	time-frequency representation
THG	third harmonic generation
TPU	tensor processing unit
UI	user interface
UL	unsupervised learning
WGN	white Gaussian noise
XFROG	Cross-Correlation FROG
XPC	experimental constraint
ZnSe	zinc selenide

1 INTRODUCTION

1.1 General Background

The development of lasers has had a transformative and far-reaching impact on technology over the past 60 years, particularly in fields such as telecommunications, medicine, and material science (Knox, 2000; Strickland et al., 1985; Diddams et al., 2000; Fermann et al., 1997; Anderson et al., 1995; Sprangle et al., 1988; Sugioka et al., 2014). In telecommunications, the advent of fiber optics revolutionized global communication by enabling reliable, high-speed data transmission across vast distances (Pogna et al., 2021; Foster et al., 2008; Martínez et al., 2010; Sinelnik et al., 2024; Wada, 2004). Similarly, in the medical industry, lasers have become indispensable tools for creating non-invasive diagnostic techniques such as imaging systems, as well as for performing high-precision surgical procedures with improved patient recovery (Hebden et al., 1995; Yun et al., 2017; Hoy et al., 2013; Huang et al., 2023; Juhasz et al., 1999; Errico et al., 2015). In material science, lasers have revolutionized both manufacturing processes and characterization methods by enabling the creation of advanced materials and providing precise analytical tools to investigate microscopic material properties (Rosker et al., 1987; Cireasa et al., 2015; Niikura et al., 2002; Vodungbo et al., 2012; von Korff Schmising et al., 2014). The versatile application of lasers has undeniably shaped the modern technological landscape, fostering innovations in nearly every scientific field.

These diverse applications demand lasers with highly specialized properties that can be precisely controlled. Specifically, ultrafast lasers with well-characterized, short temporal pulse are advantageous across various fields. In telecommunications, shorter laser pulses enable higher data transmission rates by increasing the bandwidth and speed of optical communication systems (Knox, 2000; Martínez et al., 2010). In the medical industry, ultrafast lasers allow for advanced imaging techniques, such as optical coherence tomography (Yun et al., 2017; Hebden et al., 1995), and enable high-precision surgeries, reducing peripheral tissue damage and improving patient outcomes (Juhasz et al., 1999; Hoy et al., 2013). For studying material dynamics, the temporal resolution of the laser dictates the timescales of dynamic processes that can be observed. Consequently, shorter laser pulses facilitate the study of faster and more intricate material dynamics, such as electron movements and phonon interactions, which are otherwise inaccessible (Cireasa et al., 2015; Vodungbo et al., 2012). Ultrafast lasers, with femtosecond (fs) pulse durations (on the order of 1×10^{-15} s), are therefore indispensable tools in these applications, providing unparalleled temporal and spatial resolution, enabling a broad spectrum of innovations (Saleh et al., 2019; Born et al., 2019).

Before a laser can be used for these applications, certain behaviors must first be fully stabilized and characterized. This is essential to ensure that procedures are precise, controlled, reproducible, and safe. For instance, in telecommunications, poorly characterized laser pulses can lead to signal distortion and data loss, reducing the efficiency and reliability of communication systems (Knox,

2000; Wada, 2004). In the medical field, an unstable or imprecisely characterized laser could result in imaging artifacts, leading to misdiagnosis or unintended damage during surgeries (Juhasz et al., 1999; Hebden et al., 1995). When studying material dynamics, a poorly characterized laser may misrepresent the time scales of dynamic processes, leading to inaccurate observations and flawed conclusions (Rosker et al., 1987; Niikura et al., 2002). Proper stabilization and characterization of ultrafast lasers are therefore crucial to avoid these risks and maximize the benefits across all applications.

As laser development continues to push the cutting-edge of research, the characterization techniques used must keep pace. In the case of slow lasers, with oscillations between micro and radio frequencies, there are few experimental limitations to characterization (Oppenheim et al., 2007; Hecht, 2017; Saleh et al., 2019). However, when dealing with an ultrafast laser that is already considered to be the fastest phenomena available, there are unique dilemmas that must be overcome to fully describe their behavior. These lasers operate within a timescale that is far faster than conventional measuring equipment, leaving few tools available for characterization (Diels et al., 2006; Born et al., 2019; Saleh et al., 2019). Moreover, ultrafast pulses have specific properties that impact their behavior in complex ways, making them very sensitive to effects such as dispersion, nonlinearities, and experimental conditions. As a result, the characterization techniques used must be robust enough to overcome these limitations. Standard characterization techniques will be discussed at length in Section 1.2.

An ultrafast pulse is simply an electromagnetic wave, which has two important properties to characterize : its amplitude and phase. The amplitude indicates the magnitude of the electric field, while the phase describes the position in a wave's cycle at a specific point in space and time. Theoretically, the equation of a wave (Equation 1.1) allows the identification and description of both parts of the pulse, in time, t, and angular frequency, ω. The electric field, $E(t)$, and its spectrum, $\tilde{E}(\omega - \omega_0)$, are related through the Fourier transform (FT), where ω_0 is the central angular frequency. The electric field is described by its temporal envelope, $E_0(t)$, and phase, $\phi(t)$. Similarly, the spectrum is described by the amplitude, $\tilde{E}_0(\omega - \omega_0)$, and spectral phase, $\varphi(\omega - \omega_0)$.

$$\begin{aligned} E(t) &= E_0(t)\, e^{j(-\omega_0 t + \phi(t))} \\ \tilde{E}(\omega) &= \tilde{E}_0(\omega - \omega_0)\, e^{-j\varphi(\omega - \omega_0)} \end{aligned} \tag{1.1}$$

Because ultrafast laser pulses are so temporally brief, it is impossible to experimentally measure the fast moving oscillations of the electric field, $E(t)$, or the spectrum, $\tilde{E}(\omega - \omega_0)$. Instead, characterization techniques focus on recovering the intensity of these quantities ; $|E(t)|^2$ in the time domain or $|\tilde{E}(\omega - \omega_0)|^2$ in the frequency domain (Diels et al., 2006). As a result, the imaginary component of the electric field, and therefore the exact amplitude and phase, is experimentally irretrievable (Trebino, 2002). This is known as the phase retrieval problem (PRP) which will be explained in detail in Section 2.1. While the PRP poses significant limitations characterizing ultrafast lasers, other experimental limitations are constraints to achieving this goal.

1.1.1 Sampling Theory

The first limitation in characterizing a laser system is the ability to accurately record a very fast event. While simple in theory, there are limitations to this in practice. It is understood that, generally, in order to record an event, a faster, periodic event must be used to sample it (Muybridge, 1899). According to Nyquist's Theorem, the necessary sampling rate, f_s, is at minimum twice as fast as the fastest event, f_{max}, being recorded : $f_s > 2\,f_{max}$ (Oppenheim et al., 2007; Saleh et al., 2019). Temporally, it is the inverse, where the sampling period, T_s, must be maximally half of the shortest period in the event, T_{min} : $T_s < 0.5\,T_{min} = 1/2\,f_{max}$. When the necessary sampling rate is not achieved, the signal will exhibit signs of aliasing (also known as sub-sampling). When aliasing occurs, the highest (or sub-sampled) frequencies in the signal will be overlapped with the accurately sampled, lower frequencies, causing inaccuracies in the resulting recording. To avoid aliasing, the Nyquist criterion must be satisfied. However, for the time scale in question for ultrafast lasers, satisfying the Nyquist criterion is impossible, as there are simply no events fast enough that can be used to sample the pulse with the necessary precision.

Overcoming this problem requires some experimental creativity. A very common process employs a self-sampling technique, where the pulse being characterized is used to sample itself. While this does not perfectly resolve the sampling problem described, the use of a pulse to measure itself is a fundamental technique used for characterization. Self-sampling is achieved experimentally using pump-probe interferences, which are the basis for most temporal characterization techniques. The general procedure for a pump-probe self-sampling experiment is presented in Figure 1.1.

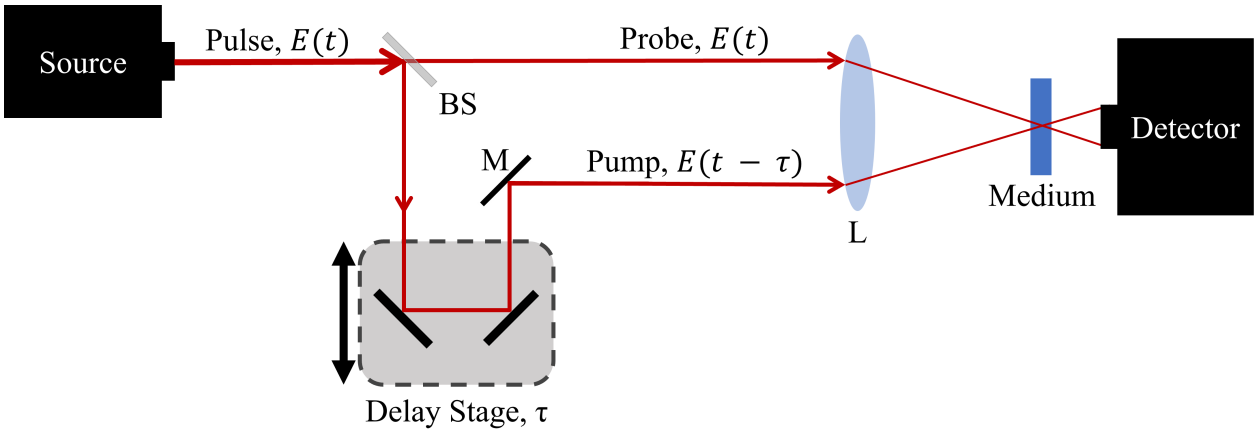

Figure 1.1 : General Experimental Pump-Probe Set-up.

The general process used in most pump-probe experiments is presented. An unknown pulse, $E(t)$, is split into pump and probe beams by the beamsplitter (BS). The pump pulse is subject to a variable delay, τ, using the delay stage. Mirrors (M) and lenses (L) are used to manipulate both the pump, $E(t-\tau)$, and probe, $E(t)$, pulses to overlap spatially in a medium. The overlap of the pump and probe in the medium generates a secondary effect (which can be linear or more complex) that is then recorded when the beam impinges on the detector. This process can be detected temporally (e.g., with a photodiode) or spectrally (e.g., with a spectrometer).

The case described in Figure 1.1 is a simple, generalized pump-probe experiment with a single pulse that is used to sample itself. The initial pulse, $E(t)$, is split into two beams, the pump and the probe, via a beamsplitter. The pump experiences a variable temporal offset, τ, using a delay stage before the two beams are spatially overlapped in a medium and impinge on a detector. A temporal sweep is performed using the delay stage to collect the measured intensity as a function of this pump-probe delay. Many variations of this basic setup have been developed for an array of applications which will be discussed in Section 1.2. These variations make pump-probe experiments extremely adaptable and powerful for advanced ultrafast characterization techniques.

1.2 Characterization Techniques

In the following Section, we discuss some techniques that are commonly used to characterize a variety of ultrafast pulses, which all apply self-sampling. These are first described in the time domain, before proceeding to spectral characterization.

1.2.1 Characterization in the Time Domain

Pump-probe experiments can be performed in the time domain using a photo-diode as the detector. Photo-diodes are semiconducting devices that convert optical energy into an electrical signal through the photo-excitation of charge carriers by the incident electric field, $E(t)$ (Liu, 2016; Born et al., 2019). The temporal response of these detectors, characterized by their rise and fall times, typically lies in the nanosecond (ns) regime (Liu, 2016; Saleh et al., 2019), although high-speed photo-diodes can reach response times on the order of tens of picoseconds (10 ps). When the optical waveform varies slowly compared to the detector response, the Nyquist criterion is met and the instantaneous electric field can be directly resolved (Oppenheim et al., 2007; Saleh et al., 2019). However, for ultrafast events, like fs pulses, the temporal resolution of the photo-diode is insufficient to adequately detect the electrical field oscillations. Consequently, the detector measures the time-averaged intensity of the incident field, rather than directly recording the electric field itself (Saleh et al., 2019; Liu, 2016). The exact value of the electric field cannot be recovered from this information. However, when paired with variations of the pump-probe experiment showed in Figure 1.1, extremely useful information can be recovered. Generally, these techniques advantageously induce some change in the pulse that modulates at a rate slower than the photo-diode reaction time, allowing these changes to then be accurately resolved by a relatively slow photodetector. Using the recorded intensity and the known change induced in the system, the electric field of the pulse can be mathematically recovered. Two powerful temporal characterization techniques are presented below.

Autocorrelation. Autocorrelation is a fundamental self-sampling technique where the generated pump-probe interference is measured (where the medium of choice is simply air). Among the most used experimental autocorrelations is the Michelson interferometer, which is presented in Figure 1.2 (Born et al., 2019; Hecht, 2017). In this simple pump-probe experiment, a pulse is split in two using a beamsplitter to produce a pump and probe, where the pump undergoes a variable temporal delay. The two beams are manipulated to superimpose and interfere as they co-propagate to the photodetector. The resulting measurement is the intensity shown in Equation 1.2, $I_{\mathrm{M}}(\tau)$, where the equation of a photo-diode is joined by an additional term, $+E(t-\tau)$, generated from the pump-probe interference (Born et al., 2019; Hecht, 2017). In each case, the additional term yields information on the spectrum and allows the estimation of the temporal pulse duration (Born et al., 2019; Hecht, 2017). This technique also has many variants, including intensity and interferometric autocorrelators, where the two beams overlap in a non-air medium (Naganuma et al., 1989).

$$I_{\mathrm{M}}(\tau) \propto \left| \int_t E(t) + E(t-\tau)\, dt \right|^2 \tag{1.2}$$

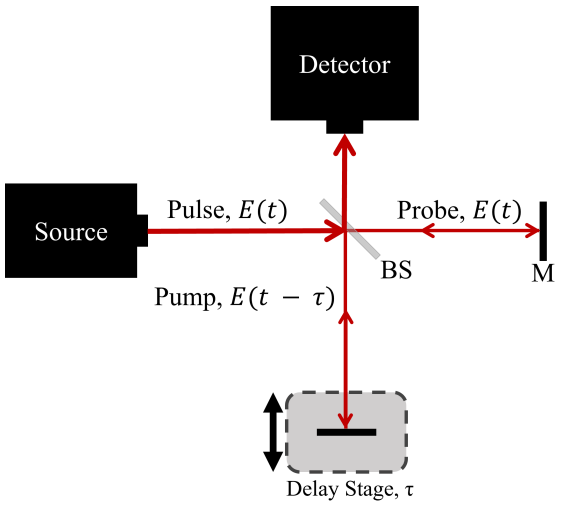

Figure 1.2 : Experimental Michelson Interferometer Set-up.

The experimental pump-probe set-up used for a simple Michelson interferometer is shown. The source pulse, $E(t)$, is split into the pump and probe beams using a beamsplitter (BS). The pump beam undergoes a variable temporal delay, τ, using the delay stage. The pump, $E(t-\tau)$, and probe, $E(t)$, beams are manipulated by mirrors (M) to overlap and co-propagate towards a photodetector. The measured intensity, $I_{\mathrm{M}}(\tau)$, is described by Equation 1.2.

Electro-Optical Sampling. Another commonly used technique is known as electro-optical sampling (EOS), where the Pockels effect is exploited to alter the polarization of a probe beam in the presence of a strong electric field (Hecht, 2017; Saleh et al., 2019). The experimental pump–probe set-up for EOS is shown in Figure 1.3. As with autocorrelation, the unknown pulse is divided into pump and probe beams by a beamsplitter, with the pump undergoing a variable temporal delay. The two beams are then directed and focused by flat and parabolic mirrors to spatially overlap

within a birefringent crystal. In this configuration, the pump field induces a transient birefringence in the crystal, modulating the polarization state of the probe through the Pockels effect. The probe subsequently passes through a quarter-wave plate, converting circular to linear polarization, and a Wollaston prism, which separates the orthogonal polarization components. The resulting intensities are detected by a balanced photodiode pair.

EOS, unlike the other intensity-based techniques discussed in this work, directly measures the temporal evolution of the electric field rather than its intensity envelope. This direct measurement yields a bipolar signal that preserves the field's sign and thus lifts the phase ambiguity inherent to self-referenced techniques. The detected intensity difference, $I_{\mathrm{EOS}}(\tau)$, is proportional to the instantaneous phase retardation, $\Delta\varphi$, which depends on the electric field of the pump pulse, $E(t-\tau)$, scaled by the electro-optic properties of the crystal (refractive index, thickness, and Pockels coefficient) (van der Valk et al., 2004; Carnio et al., 2024; Saleh et al., 2019).

$$I_{\mathrm{EOS}}(\tau) \propto \sin^2\left(\frac{\Delta\varphi}{2}\right) \tag{1.3}$$

However, EOS is not suitable for the type of ultrafast pulses we seek to characterize, particularly when these pulses are highly chirped. The electro-optic crystals used in EOS have a limited response bandwidth, and the pump and probe beams experience significant group-velocity mismatch, both of which restrict the temporal resolution to the ps range (van der Valk et al., 2004; Carnio et al., 2024). Moreover, the rapid oscillations of optical carrier fields far exceed the modulation bandwidth of the Pockels effect, causing the measurement to average over multiple optical cycles. As a result, while EOS is an important characterization technique, it is still primarily limited to terahertz (THz) and mid-infrared field detection, where the pulses occur on timescales that can be resolved.

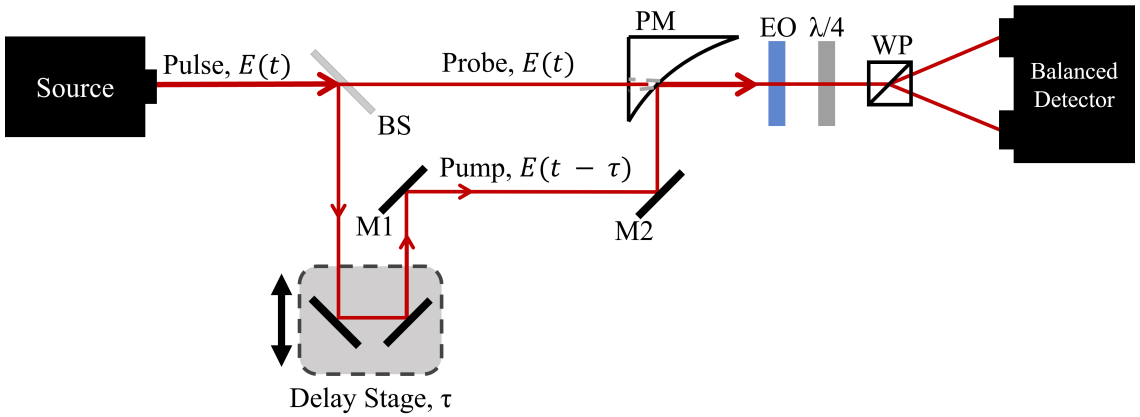

Figure 1.3 : Experimental Electro-Optical Sampling Set-up.

The experimental pump-probe set-up used for EOS is presented. As with other cases, the unknown pulse, $E(t)$, is split into pump and probe using a beamsplitter (BS). The pump undergoes a variable time delay, τ, at the delay stage. Flat (M) and parabolic (PM) mirrors are used to manipulate the pump and probe to overlap spatially on an electro-optical crystal (EO). The pump induces a change in refractive index in the birefringent crystal, which in turn modulates the polarization of the probe. The probe is then passed through a quarter wave plate ($\lambda/4$), which transforms the circular polarization to linear polarization. The linearly-polarized probe then propagates through a Wollaston prism (WP) which divides it by the polarization axes (fast and slow). Each beam then impinges on a different channel of a balanced photo-diode and the relative difference in polarization is measured. The recorded intensity, $I_{\mathrm{EOS}}(\tau)$, is described in Equation 1.3.

Other nonlinear effects can equivalently be used in the place of polarization to produce a resolvable change that provides mathematically useful information. For example, second harmonic generation (SHG) is commonly used to generate a third beam at double the fundamental frequency of the pulse, which is then measured. For many cases, these techniques can adequately recover the intensity of the pulse. However, because it is the intensity being measured, we still cannot identify the imaginary component. As a result, these temporal techniques remain inefficient for the full characterization of ultrashort pulses. Thankfully, applying the same self-sampling theory to the spectral domain can provide unique benefits that can remove this limitation.

1.2.2 Characterization in the Spectral Domain

Pump-probe experiments can also be executed in the spectral domain, instead of temporally. This can be advantageous given the inverse relation between temporal duration and spectral bandwidth. Since we are dealing with ultrafast pulses, these will generally have a spectral bandwidth broader than their temporal duration. As a result of this inverse relation, ultrafast pulses that are difficult to sample temporally have spectra that can easily be resolved. A more lengthy discussion on the topic will be in Section 2.1.

In the spectral domain, the photo-diode is replaced by a spectrometer that measures the spectral intensity of each wavelength. The spectrometer detects the individual components of the spectrum with the help of a diffraction grating or prism. When the incoming pulse impinges on the grating, angular dispersion is induced, causing each wavelength to split spatially at a different angle. As a result, each frequency is resolved into its own spatial beam that can be detected by a charge-coupled device (CCD) camera. The measured quantity does not suffer from the same sampling concerns as the temporal intensity. Instead, the spectral resolution is limited by the dimensions of the spectrometer, making it entirely independent of the limited temporal duration of the pulse. This simple alteration to pump-probe experiments allows a series of spectral-domain variations of the technique to be generated.

Frequency-Resolved Optical Gating. Among the most powerful spectral characterization techniques available for ultrashort pulses is Frequency-Resolved Optical Gating (FROG). This technique was established in the 1990s (Trebino, 2002; Trebino et al., 1997) and builds off the pump-probe autocorrelation experiments discussed previously. The original FROG is similar to the general pump-probe experimental set-up presented in Figure 1.1, but requires that the pulses overlap in a nonlinear medium to produce a secondary effect that is then detected with a spectrometer. Many secondary effects can be exploited by FROG, including SHG, third harmonic generation (THG), polarization, and self-diffraction (Trebino, 2002; Trebino et al., 2021).

The most common is SHG FROG, which is pictured in Figure 1.4, where the two beams overlap in a nonlinear crystal to generate a second harmonic. As with other pump-probe experiments, the unknown pulse is split by a beamsplitter to generate the probe and the pump that undergoes a temporal delay. These are manipulated to overlap in a crystal with a large $\chi^{(2)}$ coefficient to produce a third beam, $E_{SH}(t)$, at twice the frequency (i.e., half the wavelength) of unknown pulse, which is then detected by a spectrometer. Simply put, the pump gates the probe, allowing it to be adequately temporally and spectrally resolved simultaneously. This process is described by Equation 1.4, where the measured intensity is proportional to the second harmonic generated by the gated pulse. Since detection was performed in the spectral domain, the measured quantity is a convolution between the electric field being characterized, $E(t)$, and its gating function, $E(t-\tau)$. As a result, the intensity, $I_{\text{SHG FROG}}(\omega,\tau)$, is a function of two dimensions (the angular frequency, ω, and temporal delay, τ). Because of this, the full electric field (amplitude and phase) can be recovered from Equation 1.4 by means of numerical post-processing tools (DeLong et al., 1994).

$$I_{\text{SHG FROG}}(\omega,\tau) \propto \left| \int_t E(t)\, E(t-\tau)\, e^{-j\omega t}\, dt \right|^2 \tag{1.4}$$

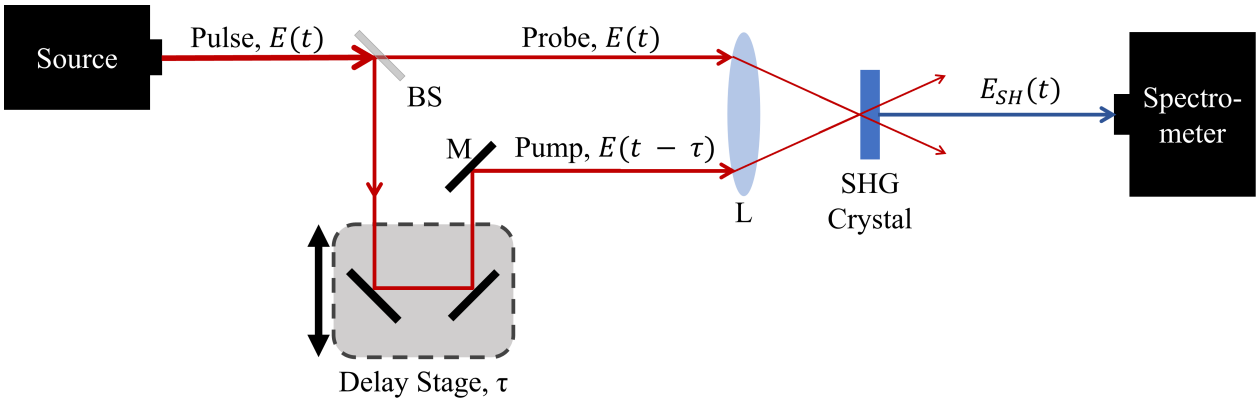

Figure 1.4 : Experimental SHG FROG Set-up.

The experimental set-up for SHG FROG is shown. The source pulse, $E(t)$, is split using a beamsplitter (BS) to produce the pump and probe. The pump undergoes a variable temporal delay, τ, with the delay line. The pump, $E(t-\tau)$, and probe, $E(t)$, are then manipulated by mirrors (M) and lenses (L) to overlap spatially in the SHG crystal. The second harmonic, $E_{SH}(t)$, is generated from the crystal and detected by the spectrometer, resulting in the intensity described by Equation 1.4.

As mentioned, many nonlinearities can be exploited with this experimental set-up to produce variants of FROG. In some variations, such as the Cross-Correlation FROG (XFROG), the pulse is instead gated by a previously-characterized pulse. XFROG is particularly useful when a shorter reference pulse is available to gate the probe under investigation (Trebino et al., 2021). Variants like XFROG were created to characterize a variety of pulses because FROG is considered such a fundamentally important experimental procedure for characterizing ultrafast pulses. FROG is useful because the temporal and spectral resolutions are rendered independent of each other. The spectral resolution depends primarily on the dimensions of the spectrometer, while the temporal resolution is primarily limited by the step size used for the temporal sweep. While other experimental conditions (such as crystal width, laser intensity, etc.) will impact the minimum achievable resolutions, neither the temporal or spectral resolutions are fundamentally limited by the temporal duration of the pulse in question. As a result, the entire complex temporal profile of an ultrashort pulse can be accurately characterized using these techniques.

While many variations of FROG exist, there are also techniques that use similar ideas for more complex system characterization. Dispersive scanning (d-scan) can characterize pulses temporally and spatially by introducing a variable dispersion in the pulse before performing SHG and recording the results (Louisy et al., 2017; Miranda et al., 2012). In place of a spatio-temporal scan, the amount of dispersion is varied through the thickness of the dispersion medium used. The result is a pulse characterized both spectrally and temporally, thanks to the SHG, as well as spatially, due to the induced dispersion (Miranda et al., 2012). More important techniques include GRating-Eliminated No-nonsense Observation of Ultrafast Incident Laser Light E-Field (GRENOUILLE) (O'Shea et al., 2004; Trebino et al., 2021), Complete Reconstruction of Attosecond Bursts (CRAB) (Mairesse et al., 2005; Lucchini et al., 2018), and Temporal Analysis by Dispersing a Pair Of Light

E-fields (TADPOLE) (Trebino et al., 2021; Bowlan et al., 2007). Each carries unique benefits for different applications, making a rich and versatile set of techniques capable of characterizing a variety of pulses.

While these techniques offer important information on ultrafast pulses, they carry limitations. There are experimental conditions that must be fulfilled to acquire the measurements (Trebino, 2002). For variants that use crystals for harmonic generation or polarization modulation, phase matching must be achieved. Depending on the working wavelength of the given pulse, it may be necessary for the crystals to be very thin to achieve the phase matching requirements. Moreover, nonlinear effects generated by these crystals only occur when considering relatively high-intensity pulses. For pulses at low-intensities, the necessary nonlinearities cannot be generated with sufficient intensity for detection. As a result, low-intensities pulses at certain wavelengths cannot be characterized experimentally by FROG. For pulses that can be characterized, FROG still requires the numerical methodologies to recover the electric field from Equation 1.4 DeLong et al. (1994). While FROG techniques have greatly improved ultrafast characterization, further improvements must still be made to both the experimental procedure and the recovery process.

Some other important techniques exist, like Spectral Phase Interferometry for Direct Electric-field Reconstruction (SPIDER), which is based on spectral shearing (Dorrer et al., 2001; Stibenz et al., 2006; Bourassin-Bouchet et al., 2013). SPIDER is uniquely able to directly measure the electrical field but the necessary experimental limitations cannot be met to achieve adequate resolution for very short, broadband, or strongly chirped pulses (Dorrer et al., 2001; Stibenz et al., 2006). Spectral autocorrelation is also a useful technique, where the set-up described in Figure 1.2 is replicated using a SHG crystal as a nonlinear medium and replacing the photodetector with a spectrometer to acquire a trace similar to a SHG FROG. Subsequently, the pulse amplitude and phase can be successfully retrieved using iterative processes like the Naganuma algorithm (Naganuma et al., 1989). Unfortunately, this technique is limited by the same set of experimental challenges when it comes to generating the second harmonic of ultrashort, highly chirped, or low-intensity pulses. While the collection of methodologies described above create a diverse toolkit of characterization techniques, there is still difficulty characterizing fs pulses due to experimental limitations.

1.3 Frequency-Resolved Optical Switching

To overcome the experimental limitations of the previously described techniques, methods such as Frequency-Resolved Optical Switching (FROSt) were developed (Leblanc et al., 2019, 2021; Brizard et al., 2020). FROSt is yet another variation on the traditional pump-probe experiment, this time developed from spatial microscopy techniques (detailed in Section 2.2). Importantly, FROSt relies on a material, known as the switch, to generate the temporal interference with the probe. The switch is typically a semiconductor whose transparency is transiently induced by an optical excitation. In other words, the semiconductor becomes partially transparent for a short duration

following excitation by a pump pulse, and this transient transparency window is exploited in the technique to temporally gate the probe. The experimental set-up used to collect a FROSt trace is shown in Figure 1.5.

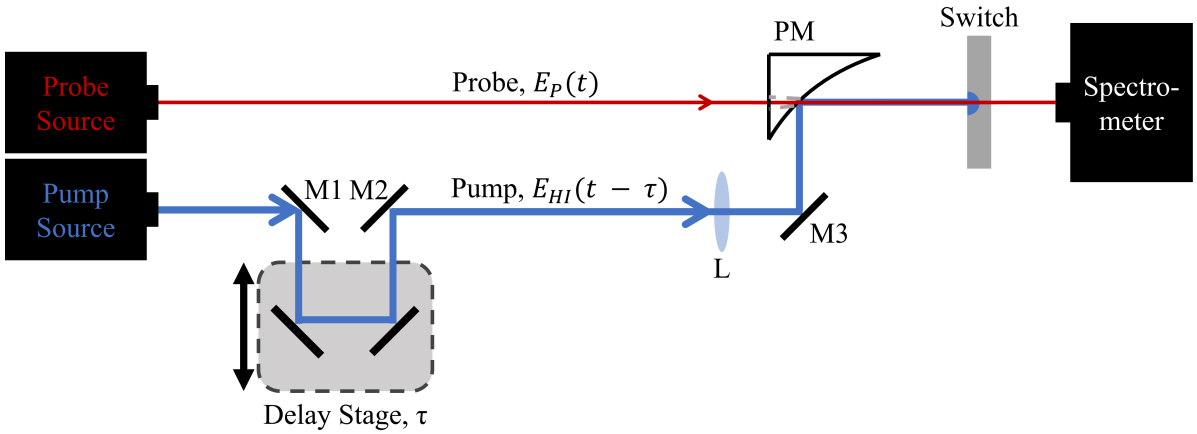

Figure 1.5 : Experimental FROSt Set-up.

The experimental pump-probe variation used to generate FROSt traces is presented. In this case, the pump and probe beams are generated from independent sources (marked as Pump and Probe Source, in blue and red, respectively). The high-intensity pump beam is subjected to a variable temporal delay, τ, using a delay stage. The pump beam, $E_{HI}(t-\tau)$, is manipulated by several mirrors (M1-3) and a lens (L) before overlapping spatially with the probe, $E_P(t)$, through a pinhole parabolic mirror (PM). The pump and probe are then co-propagated to impinge on the switch, where the pump causes the necessary population inversion to occur in the material (i.e., "flips the switch"). After the probe interacts with the photo-excited switch material, the resulting interference pattern is detected by a spectrometer. The full trace, $I(\omega,\tau)$, is generated by collecting this interference while sweeping the variable τ with the delay stage and is described by Equation 1.5.

The switch is a key element in FROSt. The selected material is optically transparent under equilibrium (i.e., in dark conditions, when not excited by a beam), but becomes opaque when excited by a high-intensity beam. Simply, the switch is a semiconductor that effectively turns off the probe when it has been excited by the pump. Zinc selenide (ZnSe), silicon (Si), and germanium (Ge) are common candidates, depending on the exact properties of the pulse undergoing characterization (Leblanc et al., 2019).

The switch function, $S(t-\tau)$, is simply the mathematical description of the transmitted intensity of the probe at various temporal delays. Theoretically, $S(t)$ will have a minimum of 0 (where the probe is fully absorbed) and a maximum of 1 (where the probe is fully transmitted). Each material will have a unique switching function which will depend on the maximum flux of the pump beam. It has been proven that these effects do not alter the accuracy of FROSt, as long as the material can effectively switch for the frequency range of interest. Similarly, the properties of the high-intensity pump have little effect on the resulting traces (assuming a necessary intensity to induce photo-excitation) (Leblanc et al., 2019). The selection of a switch material is considered an important experimental step to achieve an accurate characterization.

Unlike many pump-probe experiments, the pump, $E_{HI}(t-\tau)$, and probe, $E_P(t)$, are generated independently in FROSt. The pulse undergoing characterization is the probe, $E_P(t)$, shown in red in Figure 1.5. Meanwhile, the pump, $E_{HI}(t)$, is a secondary high-intensity beam used to generate the switching effect, shown in blue in Figure 1.5. The pump is subject to a variable temporal delay, τ, before propagating through the switch, which alters the excitation state of the material. The probe is then propagated through the switch at various states of excitation as generated by the delayed pump, $E_{HI}(t-\tau)$. The switch at each delay, $S(t-\tau)$, acts as a gating function that filters the probe before detection by the spectrometer.

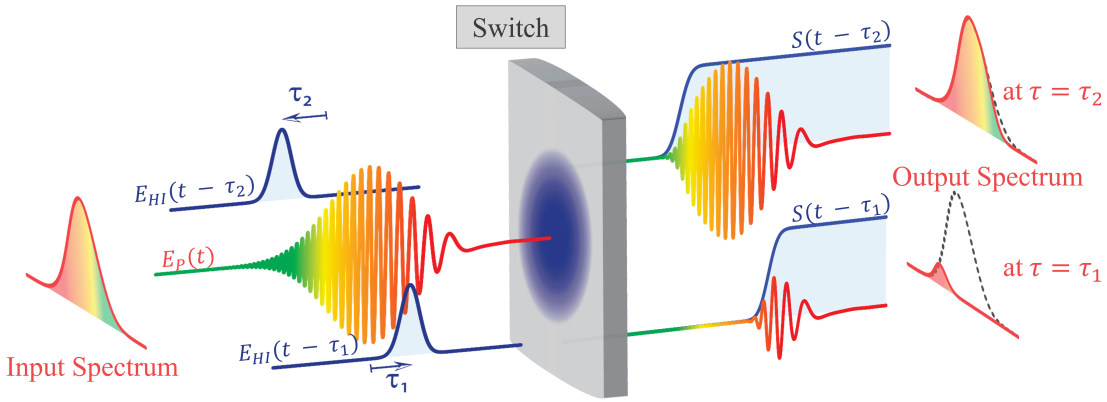

Figure 1.6 : Theoretical Principle of FROSt.

A solid switch material (Switch) is photo-excited by the high-intensity pulse, $E_{HI}(t-\tau)$, at several temporal delays, τ. The resulting population inversion in the switch generates the sliding window, $S(t-\tau)$, that is convolved with the probe pulse, $E_P(t)$. The convolved pulse is measured at the output by a spectrometer at many values of temporal delay, τ. At $\tau=\tau_1$, the material is almost entirely opaque and only a small portion of the probe spectrum is recorded. At $\tau=\tau_2$, the material is transparent, resulting in the entire spectrum being recorded. Reproduced from Leblanc et al. (2019).

The FROSt process is visualized in Figure 1.6, where a probe undergoing characterization interacts with the switch at two distinct states of excitation. At $\tau=\tau_1$, the high-intensity pump, $E_{HI}(t-\tau)$, interacts with the switch much before the probe, $E_P(t)$. In this case, the switch is off (a transmission of nearly 0% across the spectrum) and only a very small portion of the probe is propagated. At $\tau=\tau_2$, the pump incidents the switch after the probe, turning the switch on (a transmission of 100%) and propagating the entire probe. As a result, the output spectrum is the convolution between the switch, $S(t)$, and the probe, $E_P(t)$, where the temporal delay, τ, is a dependent variable.

The excitation state of the switch is controlled through the temporal delay, τ, between the pump and probe beams. By collecting successive spectra at each value of τ, a 2D trace, known as a time-frequency representation (TFR), is acquired (more on TFRs in Section 2.1). The TFR shows the evolution of the probe's spectrum, $\tilde{E}_P(\omega)$, as a function of the time delay, τ. The trace is described mathematically by Equation 1.5, where $I(\omega,\tau)$ is the intensity of the transmitted beam, $E_P(t)$ is the electric field of the probe, and $S(t-\tau)$ is the temporally delayed switch profile. As

in the case of FROG, this 2D representation carries the necessary information to determine the numerical solutions for the temporal profiles of both the probe and switch functions (Leblanc et al., 2019, 2021).

$$I(\omega, \tau) = \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt \right|^2 \tag{1.5}$$

FROSt can provide some unique experimental benefits when compared to alternative techniques, as discussed in Section 1.2. Notably, FROSt does not rely on nonlinear effects produced in crystals, meaning it is viable for low-intensity probes, has no phase matching limitations, and is polarization independent. Moreover, switch materials are traditionally inexpensive when compared to nonlinear crystals. As a result, FROSt allows a wide range of wavelengths and bandwidths to be characterized experimentally at low-intensities using a relatively low-cost, simple experimental method (Leblanc et al., 2021). Broadly, FROSt can accurately characterize beams where other techniques, such as FROG, are limited.

Unlike FROG, FROSt can also be applied for two distinct experimental cases. In this work, we focus primarily on its use as a characterization technique for an unknown pulse. In this case, the population inversion of the switch is simply a tool used to generate the necessary interference. However, in the case where the probe beam is already well-characterized, FROSt can equivalently be used to study the material properties of the switch as it undergoes photo-excitation. This is because ptychographic methods (described in the following paragraph) recover the profiles for both the probe and switch from the FROSt TFR. While not the focus of this work, this is an important experimental distinction between FROSt and FROG (which simply convolves the pulse with itself, instead of an external switch). As a result, the recovery methodologies applied to FROSt and FROG are different, though rooted in the same theory. This makes FROSt a more versatile experiment, but generates added complexities in the recovery process.

Additionally, FROSt does suffer from some experimental limitations. The switch material must be carefully selected to fit the provided description for the working wavelength of the probe. There is also a limitation imposed by the repetition rate of the laser used to pump the material, as it is necessary for the switch to relax between consecutive excitations. While these experimental limitations can be overcome easily, the dependence on ptychographic recovery methodologies to acquire the probe and switch profiles remains a significant limitation.

Ptychographic Recovery. The numerical recovery process referenced is known as ptychography (this is described in more detail in Section 2.2). Briefly, ptychography refers to a class of experiments that exploit redundancies in data collection to derive additional information through recursive numerical processes. Spatial ptychography is used to resolve information in space, whereas its descendant, temporal ptychography, does this in time. Simply put, ptychographic experiments collect interference patterns, while varying a spatial or temporal parameter. Given the known variations of that parameter, the redundancies between successive interference patterns can be used to

reconstruct the system variables. FROG and FROSt are both examples of temporal ptychographic experiments, where the temporal offset between pump and probe is varied to generate a 2D TFR with sufficient redundancies. Once the trace is acquired experimentally, the ptychographic recovery method is necessary to reconstruct the system variables (in the case of FROSt, these are the probe and switch) that most likely produced the TFR.

This process carries its own limitations, as all numerical methods do. Ptychographic recovery cannot ensure the identification of the exact solution of the experimental case, only one with the lowest error given a limited number of iterations. As a result, the recovery will always carry a certain error, as the solution found may not be the most correct one. In some cases, the process may simply fail to find any solution due to stagnation or divergence between iterations. Such cases can be reduced through the application of constraints, which require a thorough understanding of the theoretical processes that underlie the specific experimental problem. The application of constraints will introduce complexities to the implementation but significantly reduces the risk of stagnation and computational time.

Additionally, pre-processing the experimental traces, using tools such as spectral or temporal filtering and slicing, significantly improves the numerical process. However, these steps, while commonly used in signal processing, can vary the outcome of the solution determined and introduce bias from the researcher. Input parameters can lead to variations in the recovered solutions, which makes the entire process difficult to reproduce reliably. Broadly, the most significant limitation of FROSt is the computation time required and limited reliability of the determined solution. While experimental restrictions can be relaxed when using FROSt, there remain significant limitations on the accurate and timely recovery of the pulse, due to the numerical methodologies necessary.

1.4 Artificial Intelligence

For many numerical processes, like ptychographic recovery discussed above, recent advances in artificial intelligence (AI) may provide a valid alternative. Many modern AIs are neural networks (NNs) that, if effectively trained for the specific task, can generate an approximation of the mathematical solution with great accuracy in real time. When a NN is trained to identify a specific pattern (such as a mathematical relation), they can replicate the solving process achieved by numerical methods to produce a valid output without performing the actual computations. This can be conceptualized as an extension of statistical analyses, such as nonlinear regressions (Brunton et al., 2017; Goodfellow et al., 2016; Bishop, 2006). Regression analysis extracts a mathematical relation between a parameter and the system's dependent response to its variation, which can be described as the numerical mapping of the system's underlying mathematical operations. When a NN is well-trained for a specific mathematical relation, this mapping is extremely accurate and produces valid solutions (Li et al., 2025). In practice, these mappings are determined through a training process and can result in an AI that replicates the behavior of numerical methods with unique benefits.

There is evidence that supports the legitimacy of AIs trained to replicate phase recovery methods (Zahavy et al., 2018; Gianani et al., 2024). Literature has shown that AI learning algorithms can be effective in overcoming many of the limitations of ptychographic recovery. In Zahavy et al. (2018), a deep neural network (DNN) was developed to reconstruct pulses from a FROG trace as an alternative to ptychography. They created a procedure to train an AI that was able to recover the solution with an error rate comparable to the standard ptychographic and statistical analyses. They demonstrated that an AI could accurately recover pulse characteristics, even from experimental results with low signal-to-noise ratio (SNR). Their implementation showed many benefits to replacing the ptychographic method with an AI algorithm. The proposed algorithm makes pre-processing unnecessary, which improves reproducibility. It also improved the computation time required to extract the necessary information. Instead of allotting time for convergence on each individual problem, resources were invested to train an algorithm capable of extracting the same information from all traces in real time. However, these benefits are only achievable with an algorithm generated by a successful dynamic training process, which can be resource intensive and complex to execute. Moreover, Zahavy et al. (2018) was done on SHG FROG traces which has a simpler numerical recovery process, thanks to the self-gating nature of SHG FROG. Generating the equivalent methodology for other types of TFRs is not trivial.

1.4.1 Learning Procedures

Many training processes exist and will vary depending on application. These can broadly be grouped into two methods : supervised and unsupervised learning. These will be discussed in generalized terms before proceeding to a more specific discussion in Section 2.3.

A common method to train for applications such as image recognition is supervised learning (SL). SL uses a dataset, which is a large number of examples that is used to relate the image to a desired output from the AI. During the training process, this dataset is fed into a NN, which has parameters that are adjusted to reliably produce the desired output (Brunton et al., 2017; Goodfellow et al., 2016). This is repeated for all the examples in the dataset, until the NN's parameters generate an output that is accurate. Initially, the output from the NN is entirely random because the parameters are arbitrarily selected. However, as more examples from the dataset are presented, these parameters will quickly optimize to produce the desired response from various inputs (Li et al., 2025). Once this is achieved, a well-trained model will produce the correct answer for an image which was not included in the dataset. While the above is a relatively simplified description, this is an extremely powerful approach that can be used to map complex relations between inputs and outputs.

Now, we consider SL in a specific case, where an AI is being trained to identify images of dogs. The dataset will consist of many images, some containing dogs while others will not. Each image will be associated with a label to identify it (in this example, 0 for images that are not dogs and 1 for images that are dogs). To begin, an untrained NN with randomized parameters is used. When

an image from the dataset is fed into the NN, it is decomposed into feature maps. Feature maps can be conceptualized simply as filtered images, where the filter properties are determined by the NN parameters (Gonzalez et al., 2018). Some examples of feature maps, and the filters used to produce them, are presented in Figure 1.7. Each feature map will emphasize different pieces of information from the image and relates that to the label associated with the image (Gonzalez et al., 2018; Brunton et al., 2017). During the training process, the filter parameters are updated until the information extracted from the feature maps reliably produce the correct output (in this case, simply 0 if not a dog or 1 if a dog). For example, the edge feature map in Figure 1.7c emphasizes fur and whiskers in white, which should invariably be produced by any input image that contains a dog with fur and whiskers.

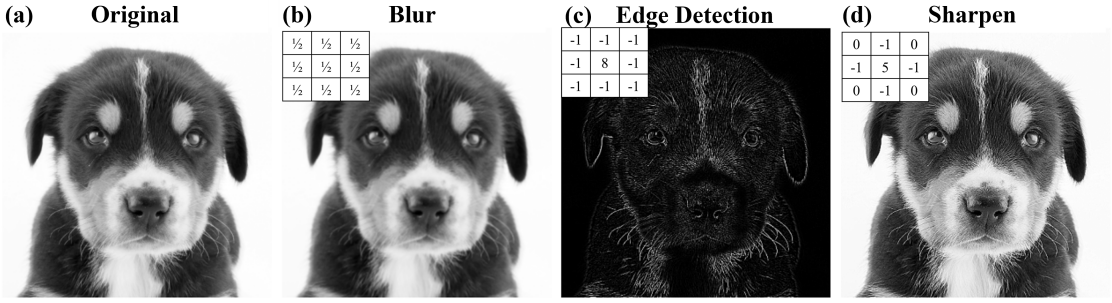

Figure 1.7 : Feature Maps of a Dog.

A black-and-white image of a dog, shown in (a), is convolved with three filters : blur (b), edge detection (c), and sharpen (d) (Gonzalez et al., 2018). The resulting feature maps generated by each filter are presented in (b) through (d), respectively. The parameters (kernels) for the filters are displayed as insets in the upper-left of each feature map. Mathematically, the feature maps are produced by convolving the initial image with the filter kernel, resulting in feature maps that contain different sets of information from the original image. (a) was acquired from Geographic (2025).

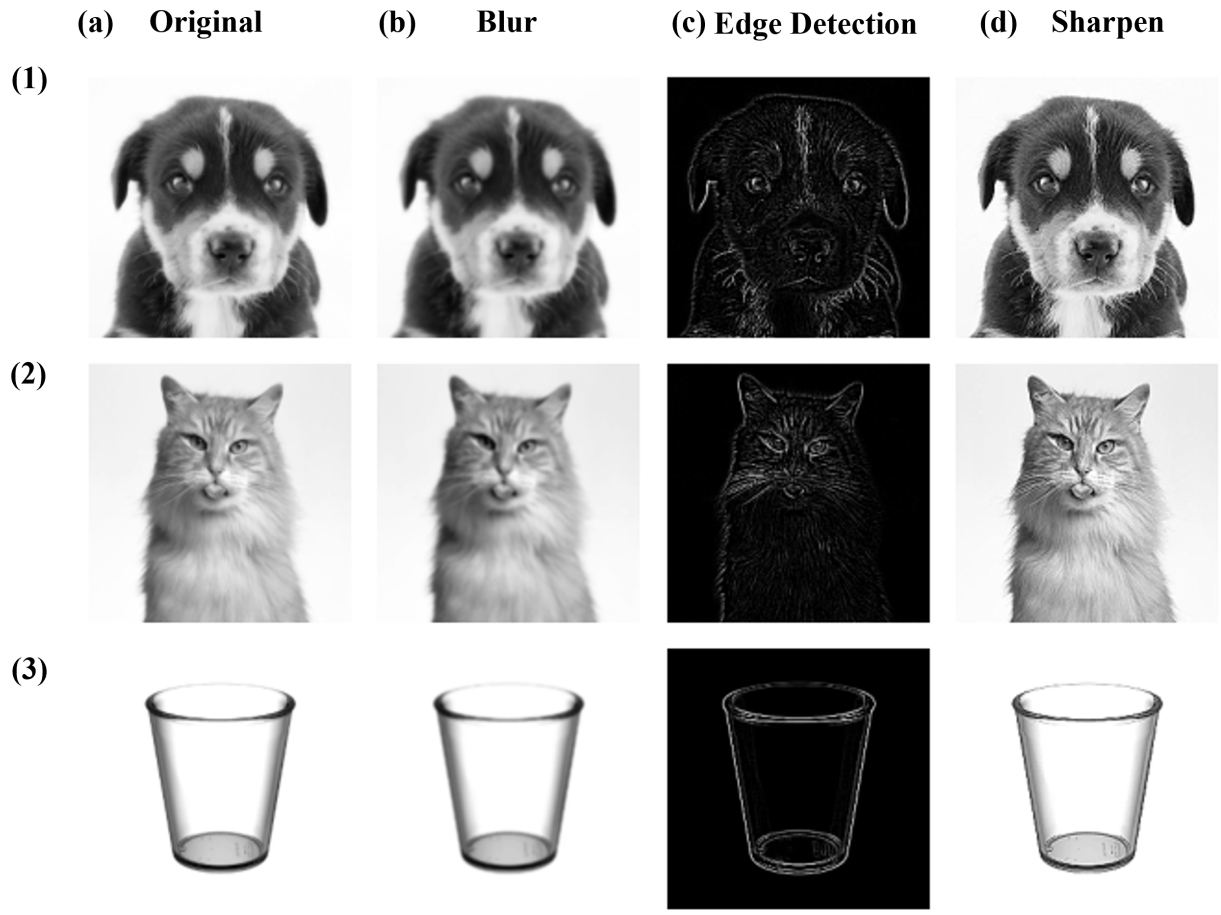

Figure 1.8 : Feature Maps of a Dog, a Cat, and a Cup.

Similarly to Figure 1.7, original images are shown in column (a) and convolutions with three filters (blurring, edge detection, and sharpening) are shown in columns (b) through (d), respectively. The dog and its feature maps from Figure 1.7 are shown in row (1). Row (2) shows a cat and its feature maps, while row (3) shows a cup and its feature maps. Visually, some similarities (such as fur, whiskers, and eyes/nose shape) are shared by the feature maps for the dog and cat, in rows (1) and (2). Meanwhile, the maps for the cup in row (3) are notably dissimilar. Images in (a) were acquired from Geographic (2025); American (2025); Hospequip (2025), respectively.

In Figure 1.8, the same NN is fed images of the dog, a cat, and a cup, where each input produces 3 feature maps (one generated by each filter applied). For a well-trained AI, an input such as the dog in Figure 1.8a will be decomposed and the NN will produce an output of 1 (true, this is a dog). Theoretically, for images that do not contain a dog, like those presented in rows 2 and 3 of Figure 1.8, the decompositions can be differentiated from that of a dog and the AI will output a 0 (false, this is not a dog). In practice, this demonstrates a significant fallibility of AI (Brunton et al., 2017). As is seen in Figure 1.8c, the feature maps for the dog and the cat are visually quite similar, while those of the dog and the cup are notably different. An AI can quite easily identify that the cup is different from the dog and output a 0, as is expected and desired. For the cat, however, an AI will likely identify the similarities in the feature maps and produce an output of 1 (true, this is a

dog). This demonstrates that AIs are prone to errors when processing inputs that are ambiguously similar, which creates instability in their behaviors. An effective training procedure will consider these ambiguous cases and reduce the error rate, though this still presents an important shortcoming of AI.

An important consideration to achieve an effective training protocol is the use of a high-quality dataset. For example, if a dataset contains images that are not accurately matched to their labels (e.g., an image of a dog labeled 0), the resulting training will be insufficient. These datasets must also be very large and provide many diverse examples. As a result, generating labeled datasets is very resource intensive, requiring the manual collection and tagging of each image. Acquiring high-quality datasets, however, are key to successfully applying SL training procedures.

Given the nature of datasets, it is often impossible or impractical to acquire them for certain applications. In these cases, they can be artificially generated through simulations. Simulated datasets can be used to train an algorithm but this exposes AI to a new susceptibility, known as the sim-to-real gap. The sim-to-real gap is a result of the discrepancies between simulations and real world cases. In a simulation, the parameters can be tightly modulated by the researcher, which is not the case in real world experiments. Moreover, many mathematical models that underlie simulations rely on simplifications of the true conditions that may not be entirely accurate in experimental settings. For example, any data collected experimentally will have some degree of noise (whether thermal, acoustic, or electrical) that is incredibly complex to reproduce in simulations (Ahmed et al., 2001; Middleton, 1999; Gravel et al., 2004; Scholten et al., 2003). While an AI trained on simulated data may perform excellently on simulated inputs, these discrepancies can result in poor behavior when faced with experimental data. The sim-to-real gap is a well-established limitation in robotics and AI that is studied extensively (Zhao et al., 2020; Chen et al., 2024; Shashua et al., 2021; Zhang et al., 2022; Liang et al., 2025).

The sim-to-real gap can be reduced by using a training protocol known as unsupervised learning (UL), which does not require a labeled dataset (Brunton et al., 2017; Gonzalez et al., 2018). UL training procedures can offer an alternative to SL that is able to overcome the discrepancies between simulated and experimental datasets while being trained on a more easily acquired dataset. UL still utilizes datasets, though these are much larger, cheaply acquired, and of low quality. UL relies on more complex statistical processes to identify similar and dissimilar patterns in these much larger datasets (Brunton et al., 2017). While this can be extremely beneficial in application, UL suffers from high risks of instability during training and is much less intuitive to conceptualize.

Returning to the earlier example of an AI identifying dogs, using UL will allow a NN to identify that the images of dogs and cups are different from one another, without having to relate them to the labels of 0 or 1 (Goodfellow et al., 2016). The process would be very similar, where an image from the dataset is fed into an untrained NN to extract feature maps. After consuming a large enough dataset, the AI may then be able to group them by similar and dissimilar features (Brunton et al., 2017). For example, the feature maps in Figure 1.8c can be grouped based on the ratio of

white to black pixels. In this case, the cat and dog would be grouped separately from the cup, as the presence of fur generates more edges and results in more white pixels. This is one extremely high-level and simplified example to demonstrate the pattern recognition necessary to achieve UL. Generally, UL can be very effective when there is an underlying, unchanging logic to the pattern that can be decoded from the feature maps. However, if this does not exist or cannot be identified, the training will be unstable and will simply diverge. As a result, SL is much easier to implement, though UL is generally more robust when facing the real-to-sim gap (Goodfellow et al., 2016; Chen et al., 2024).

1.5 In this work

The aim of this work is to generate a trained AI algorithm that is able to recover the temporal profiles (both amplitude and phase) of an ultrashort pulse from its FROSt trace with comparable accuracy to the traditional ptychography recovery methods. This requires the establishment of a simulated dataset, a training procedure, and the necessary validation steps. Zahavy et al. (2018), which was discussed previously, had similar goals using a SHG FROG trace in place of a FROSt. This work was closely inspired by their methodology and will be widely referenced.

The information will be presented as follows : Section 1 described the general problem, the current standardized methods for pulse characterization, and presented the theoretical and experimental principles of FROSt. The utility of applying AI to this problem was also introduced. Section 2 will provide relevant background information regarding the PRP, the standard ptychographic recovery process being replaced, and the AI tools necessary to achieve this. Section 3 describes the methodology followed for the entire work. This includes the process to generate simulated datasets for training, the DNN architecture used, the training procedure applied, and the validation process used to evaluate the quality of the resulting algorithm. Results are presented in Section 4, followed by a discussion in Section 5. Conclusions and next steps are presented in Section 6. Appendices are found in Section 7. These include the codes used to apply this methodology in Section 7.1 and some additional predictions in Section 7.2.

2 THEORY

In Section 2, we present the relevant theory that preempted this work and was applied in the methodology. Section 2.1 provides an in-depth description of ultrafast pulse characterization and the PRP. A detailed explanation of the FROSt ptychographic recovery process is presented in Section 2.2. Finally, some relevant machine learning fundamentals are detailed in Section 2.3. The information presented in this Section is necessary to support the subsequent methodology described in Section 3.

2.1 The Phase Retrieval Problem

The electric field of an ultrafast pulse is defined in Equation 1.1, where phase — temporally $\phi(t)$ and spectrally $\varphi(\omega)$ — is an important property that adds complexities to the relationship between the time and frequency domains. Broadly, temporal and spectral properties are inversely proportional. This was mentioned in Section 1.2 when discussing the advantages of spectral characterization for ultrafast pulses, since a narrow temporal duration will often produce a broader spectral bandwidth. However, this is not a strictly inversely proportional relationship, due to phase. The additional complexity of phase can be described mathematically by the Time-Frequency Uncertainty Principle, which states that temporal and spectral properties cannot simultaneously be mapped with full certainty. As a result, there is a minimum uncertainty when relating the temporal duration, Δt, to the spectral bandwidth, $\Delta\omega$, of a pulse. This is presented mathematically through an inequality : $\Delta t\,\Delta\omega \geq \text{TBP}$, where the time-bandwidth product (TBP) describes the pulse shape (for a Gaussian pulse, TBP = 0.441 (Diels et al., 2006)) (Boashash, 2016). In the optimal case, where the pulse is as brief as physically possible, the inequality can be simplified to $\Delta t\,\Delta\omega = \text{TBP}$. However, when more complex phases are introduced, the pulse will broaden such that the exact temporal duration, Δt, is unknown. In this case, the inequality must hold true but cannot be exactly solved for $\Delta\omega$. As a result, the Uncertainty Principle mathematically describes the relation between spectral bandwidth, temporal duration, and pulse shape.

Since it is established that the temporal shape of the pulse is directly linked to its spectral bandwidth and temporal duration, it becomes increasingly valuable to produce a 2D representation that presents both the time and frequency profiles simultaneously (Boashash, 2016). Many experimental processes, such as FROG and FROSt, have the goal of acquiring a TFR. A spectrogram is generated computationally using the Short-Time Fourier Transform (STFT) (Jaganathan et al., 2016; Boashash, 2016). STFT is a technique wherein a complex electric field, $E(t)$, is convolved with a windowing function, $W(t-\tau)$, to generate a 2-dimensional spectrogram, $I_{\text{STFT}}(\omega,\tau)$, as is presented in Equation 2.1.

$$I_{\mathrm{STFT}}(\omega,\tau) = \left| \int_t E(t)\, W(t-\tau)\, e^{-j\omega t}\, dt \right|^2 \tag{2.1}$$

Figure 2.1 demonstrates the utility of a TFR by showing the STFT of two pulses that share a temporal profile. Each subfigure shows the STFT spectrogram with its temporal profile along the x-axis and the spectral profile along the y-axis. While it is clear from the x-axes that the two pulses in Figure 2.1 share the same temporal duration, they differ dramatically in the spectral bandwidth. As has been explained by the Uncertainty Principle, this is possible due to the existence of phase. In Figure 2.1a, the pulse has a temporal phase of $\phi(t) = 0$. In this case, there is no offset between the frequencies and the entire bandwidth is present for the duration of the pulse. This is clearly noted by the horizontal trace generated in the spectrogram of Figure 2.1a. This type of pulse is known as a Fourier Transform-Limited (FTL) pulse, where the phase ($\phi(t) = 0$) produces the smallest possible duration, Δt, for a given bandwidth, $\Delta\omega$. This is the case where the Uncertainty Principle can be simplified to $\Delta t\, \Delta\omega = \mathrm{TBP}$. Meanwhile, in Figure 2.1b, the same temporal profile is presented with a much larger spectral bandwidth. If the equality of the Uncertainty Principle was always true, it would be expected that the broader spectrum would result in a more narrow time pulse. However, the existence of phase can account for this discrepancy. While more frequencies are presented in the pulse, the delay in the time of arrival forces the temporal profile to broaden. The time of arrival of each frequency is visualized by the sloping in the spectrogram of Figure 2.1b, where higher frequencies exist at the end of the pulse. This pulse is considered chirped, as the temporal phase is non-zero and time dependent ($\phi(t) \neq 0$). These are the cases where the Uncertainty Principle cannot be simplified (these are designated as a chirped pulse). When the phase becomes dependent on time or frequency, the variations between the FTL and chirped pulse can become increasingly complex and diverse. As a result, a 2D TFR can be useful in presenting phase information that is not communicated in a single 1D representation. Particularly, the STFT spectrogram allows for the temporal and spectral profiles of the pulse to be shown simultaneously in a single representation, which is key for visualizing phase.

An important clarification of terminology must be made when describing chirped pulses. In the technical definition, a chirp describes the application of a temporal phase. However, it is common in the research field (Born et al., 2019; Diels et al., 2006) to use chirp to also describe spectral phase modification (e.g., the use of a dispersive medium). The more general use of chirp will be used throughout this work to refer to the application of both temporal and spectral phase.

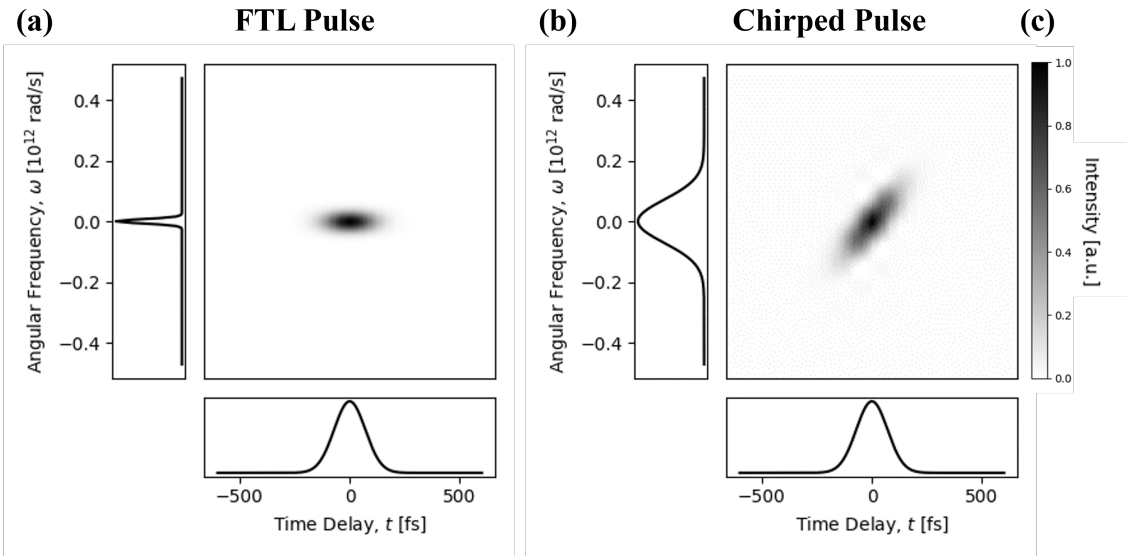

Figure 2.1 : STFT Spectrograms Demonstrating the Uncertainty Principle.

STFT Spectrograms of two pulses : the FTL in (a) and a Chirped in (b). A Gaussian pulse with a full-width half-maximum (FWHM) temporal duration of 100 fs is presented. In (a), the FTL pulse is shown, where the phase component, $\phi(t)$, is zero. This can be compared to a musical chord, where all notes are played at once for the duration of the pulse. Meanwhile, in (b), the same envelope is presented with a non-zero phase component, $\phi(t) \neq 0$. As a result, the spectral bandwidth undergoes a temporal evolution throughout the duration of the pulse. The chirped spectrogram is more similar to a scale, where notes are played consecutively. Both pulses share the same temporal profiles but distinct spectral bandwidths, demonstrating the analytic benefits of the STFT method. Their shared colormap is in (c), describing the intensity, $I_{\mathbf{STFT}}(\omega, \tau)$.

The STFT is theoretically linked to the experimental FROG and FROSt processes discussed in Section 1.2. The Equation 2.1 closely resembles Equations 1.4 and 1.5, showing the similarities between the techniques. These methods are differentiated by the windowing functions used, though they are all convolutions generated by a temporal delay. The STFT represents the optimal case, where the window function is much shorter than the ultrafast pulse and is able to accurately sample it. However, we know that this condition can rarely be achieved experimentally for an ultrafast pulse, as the pulse itself is already the shortest possible event to exist. In these cases, FROG can be used, wherein the pulse is self-gated ($W(t) = E(t - \tau)$) such that the windowing function is equivalently short as the pulse itself. If acquired with the correct conditions, the FROG trace can be used to recover phase information using simple numerical methods (DeLong et al., 1994). Since the recovered intensity in FROG (see Equation 1.4) is dependent solely on the electric field of the pulse, we need only identify a single variable from the 2D TFR, which can easily be achieved. As a result, FROG is a close second to the optimal STFT method. However, it was discussed in Section 1.2 that FROG has experimental limitations that make it inaccessible to certain pulses (e.g., a low-intensity pulse that cannot generate a second harmonic). In these cases, a technique such as FROSt is the closest experimental approximation to the STFT. FROSt differs more significantly from STFT than FROG, since the windowing function (in this case, the switch) is temporally very long compared to the pulse (since the bounds of $S(t)$ are non-zero, $\lim_{t\to-\infty} S(t) > 0$). It is necessary to ensure that

the inversion dynamics are fast enough for the switch intensity, $|S(t)|^2$, to be non-constant over the time window spanned by the pulse under study, Δt, such that $\frac{d(|S(t)|^2)}{dt}\big|_{\Delta t} \neq 0$. This condition is easily met using materials like those discussed in Section 1.3 (Leblanc et al., 2019, 2021). The most significant effect of using a very slow switch is that the resulting TFR will not resemble the one generated by a STFT (as only half the TFR is properly gated). More notably, the use of the switch does create additional complexities in the numerical recovery of the electric field. Unlike FROG, the measured intensity in FROSt is dependent on the pulse, as well as the switch (see Equation 1.5). As a result, the solution for a FROSt TFR requires two vectors (the pulse and switch) while those for FROG need only identify one (the pulse) (DeLong et al., 1994). This means that the solutions for a FROSt trace will exist as a subset of the electric field and switch spaces, which creates additional complexities when implementing the numerical recovery. Because of this, the numerical recovery method for FROSt is relatively complex and time consuming (see Section 2.2 for more).

In all experimental cases, regardless of the window properties, only the spectral or temporal intensities of an ultrafast pulse can be measured (due to the ultrafast oscillations of the electric field), which results in a real-valued TFR. However, when measuring the intensity, the real and imaginary components of the electric field are combined into a single vector. Additional information is then necessary to re-identify the two components from the infinite possible combinations. This is a simple Pythagorean problem, where the x and y components of a hypotenuse can only be recovered if both the magnitude and angle of that hypotenuse are known. If both parameters are unknown, there exists infinite possible combinations of real and imaginary components that result in the same hypotenuse. As such, for any real-valued 2D TFR, there requires some numerical recovery to re-identify the real and imaginary components.

In short, the existence of the Uncertainty Principle means that, to characterize a pulse fully, the amplitude and phase components must both be described. Without the knowledge of both, the properties of the pulse can vary greatly. Simultaneously, because of experimental limitations, only the intensity components can be measured. This creates what is known as the 1D PRP, which is a fundamental problem with infinite valid solutions. Because the 1D PRP has infinite equally valid solutions, it is impossible to identify which unique solution truly exists in that system. Simply, a pulse cannot be fully characterized with only the 1-dimensional description of either the temporal or spectral intensities. Using techniques like FROG and FROSt, a 2D TFR representation can be acquired. If it is collected with the necessary resolution (see Section 2.2 for more on the redundancy condition), the TFR carries additional information in the second dimension. This introduces the 2D variant of the PRP, which narrows the solution space such that it can generally be solved through numerical methods, like the ptychographic recovery method mentioned in Section 1 and discussed in Section 2.2 (Jaganathan et al., 2016; Boashash, 2016).

2.2 Ptychography

We have made extensive reference to ptychographic recovery methods throughout this work, as FROSt was developed by pairing pump-probe experiments with the well-known microscopy process (Rodenburg, 2008; Maiden et al., 2011; Ou et al., 2013; Thibault et al., 2009). Here, we describe the development of spatial and temporal ptychography techniques, as well as the specific numerical processes used to solve for the switch and pulse vectors in FROSt.

2.2.1 Spatial and Temporal Ptychography

Spatial ptychography is the basis upon which temporal ptychography methods were developed. In spatial ptychography, there are two unknowns (a pulse, $E(x, y)$, and an object, $S(x, y)$) that must be characterized in 2D space, (x, y). This can be described visually through Figure 2.2. The incident pulse, $E(x, y)$, demarcated in red, interactions with the object, $S(x, y)$, which sits on the object plane (note, the focal point of the pulse occurs at the object plane). The interference pattern, $I(x, y)$, which results from the interaction between pulse and object, is detected in the far-field. The incident pulse is then shifted spatially to the coordinate $(x - x_0, y - y_0)$ and the measurement is repeated to generate many interference patterns, $I(x, y, x_0, y_0)$.

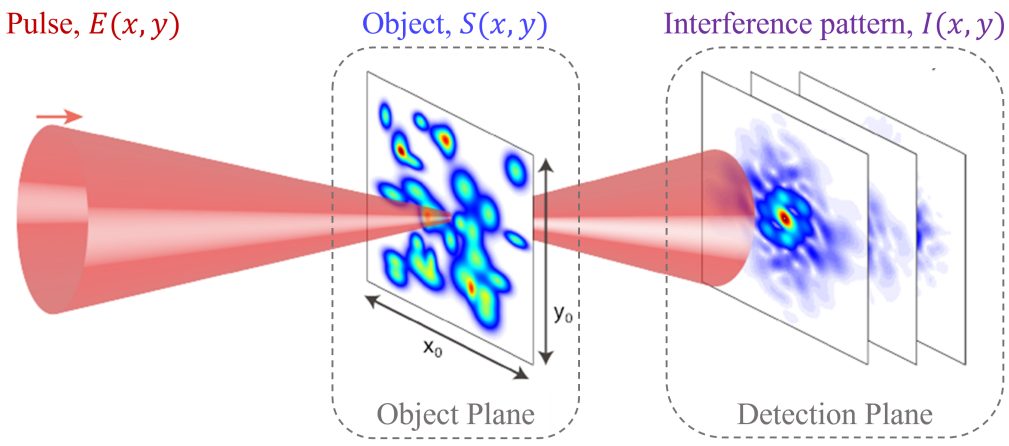

Figure 2.2 : Visualization of Spatial Ptychography.

A visualization of a generic spatial ptychography experiment. The incident pulse, $E(x, y)$, is shown as a beam in red. The object, $S(x, y)$, is an arbitrary 2D object shown in shades of blue. The measured intensity, $I(x, y)$, is the convolution between pulse and object at a given spatial offset, (x_0, y_0). The offset is varied to generate a number of interference patterns that can be used to reconstruct the spatial profiles of both beam and object. Partially reproduced from Leblanc (2016).

This procedure can be described through the mathematical relation in Equation 2.2, where ω is the angular frequency while $r = (x, y)$ and $r_0 = (x_0, y_0)$ for simplicity (Leblanc, 2016).

$$I(\omega, r_0) = \left| \int_r E(r)\, S(r - r_0)\, e^{j\omega r}\, dr \right|^2 \tag{2.2}$$

Equation 2.2 is notably similar to the governing FROSt theory in Equation 1.5, the Equation 1.4 describing SHG FROG, and the STFT method presented in Equation 2.1. The equations are related because these processes can all be considered as temporal ptychographies. Simply, the spatial offset is replaced by a temporal offset between beam and object to achieve the scanning effect. FROSt can be re-conceptualized using this context. Consider that the pulse being characterized is the incident beam and the object is the switch. The temporal offset between the switch function and the probe is achieved by offsetting the moment of photo-excitation (i.e., the overlapping of the high-intensity pump and the switch). This offset between the pump and probe allows the interference pattern to be measured at various points in time. This process can be described by a variant of Equation 2.2, where the spatial variable, r, is replaced by a temporal one, t. The resulting equation is nearly identical to the one describing FROSt in Equation 1.5. As with spatial ptychography, the collected measurement must undergo numerical recovery to extract the profiles of both the object and the probe (though the profiles are now resolved temporally, not spatially).

Object and Detection Planes. Ptychography, like many experimental setups, relies on the mathematical relation between the object and detection planes. As visualized in Figure 2.2, the object plane exists at the focal point of the probe's electric field, $E_P(t)$, where it interacts with the object, $S(t)$, at a various temporal delays, τ. The 2D electric field at this plane is denoted as $E(t, \tau)$ and can be calculated through Equation 2.3. The Equation 2.3 describes the 2D electric field on the object plane, which is simply the inverse FT of the complex TFR.

$$E(t, \tau) = \int_\tau E_P(t)\, S(t - \tau)\, d\tau \tag{2.3}$$

The electric field of the object plane described in Equation 2.3 cannot be directly measured experimentally. The measured quantity is collected instead at the detection plane where the far-field Fraunhofer effect takes place. Therefore, just as the object plane is where the probe shines on the switch, the detection plane is where the probe incidents on the detector. The property at this point is denoted as $\tilde{E}(\omega, \tau)$ and is related to the electric field at the object plane through the FT as presented in Equation 2.4. Therefore, the measured intensity, $I(\omega, \tau) = |\tilde{E}(\omega, \tau)|^2$, is the convolution between object and pulse profiles, where $\tilde{E}(\omega, \tau)$ is defined as the complex 2D spectrum in Equation 2.4.

$$\tilde{E}(\omega, \tau) = \int_t E(t, \tau)\, e^{-j\omega t}\, dt \tag{2.4}$$

The known relation between the object and detection planes (Equation 2.3) will be exploited in each iteration of the recovery process to reach convergence on a valid solution.

Redundancy Condition. Ptychography is a successful methodology only if an important condition is met while collecting the TFR. The ptychographic recovery process relies on repeated information from successive measurements that is stored in the interference patterns collected at different offsets. This is key as these redundancies generate constraints on the 1D PRP, causing the set of solutions to become finite. This redundancy is described mathematically as the information that overlaps between successive 1D scans, ensuring continuity in the characterized pulse. This is a common criteria required for unambiguous and accurate phase recovery from TFRs (Dorrer et al., 2001; Lucchini et al., 2018; Jaganathan et al., 2016; Boashash, 2016).

Consider the spatial experimental case presented in Figure 2.2, where a pulse is impinging on the object to characterize it spatially. When the pulse overlaps with the object at its focal point, there is a small area where the convolution occurs (we call this area, a_1, for simplicity). When the pulse is shifted by (x_0, y_0), there is a new area where the pulse and object overlap (we call this area a_2). This is repeated over the entire object, creating many small areas, a, where the object and pulse overlap. This is the essence of a raster scan, where each scanning area a becomes a pixel to form an image (Gonzalez et al., 2018). For ptychographic recovery to be unambiguously successful, it is very important that these areas also overlap partially with one another (i.e., a_1 overlaps partially with a_2, which is partially overlapped with a_3, and so on). This overlap is known as the redundancy condition and it ensures continuity between successive scans. If this condition is not met, the image will lack sufficient resolution to undergo ptychographic recovery.

Temporal ptychography has an equivalent redundancy condition, though the areas where the pulse and object interact are now temporal slices. The size of each "area" is decided instead by the temporal overlap between the object and the pulse, which is in turn governed by the delay step. As with spatial ptychography, it is necessary that the step size, τ, be small enough that the temporal slices are partially overlapped with one another. Experimentally, the step size must be selected carefully to ensure the adequate redundancies are contained in the TFR. We can simplify this concept by considering a puzzle where connected pieces have continuities in the larger image. The FROSt TFR is the image while the individual temporal slices are puzzle pieces that must be arranged in chronological order to show the overlaying image.

The redundancy condition can be conceptualized mathematically with the help of Nyquist's Theorem, which states that the sampling rate must be at least twice the maximum frequency being measured to avoid aliasing. To ensure the condition is met, the selected step size must meet the following criteria : $\tau < 1/2\,\omega_{min}$, where τ is the offset step size and ω_{min} is the minimum angular frequency that can be detected without aliasing (Oppenheim et al., 2007). For a pulse having an estimated minimum angular frequency of $1.5 \times 10^{12}\,\mathrm{rad\,s^{-1}}$ (i.e. a period of 4.18 ps), the maximum allowable τ is 2.10 ps. Experimentally, this temporal resolution will relate to the spatial step size of

the delay stage (e.g., the step of a stepper motor). Optimally, the step size would be much smaller than the maximum allowable value to acquire a higher resolution TFR. If the redundancy condition is not met, the numerical recovery procedure described below will not converge successfully.

2.2.2 Numerical Recovery Procedure

The goal of ptychographic recovery is to find the probe, $E_P(t)$, and object, $S(t-\tau)$, that generate the transmitted electric field, $E(t,\tau)$, with a spectrum, $\tilde{E}(\omega,\tau)$, that most closely matches the experimentally-measured intensity. The general procedure followed for ptychographic recovery is similar to both the Gerchberg-Saxton and Naganuma algorithms (Fienup, 1982; Naganuma et al., 1989) and is described at length in Leblanc (2016). The four step process followed in each iteration is shown in Figure 2.3 and described for iteration $g = 0$.

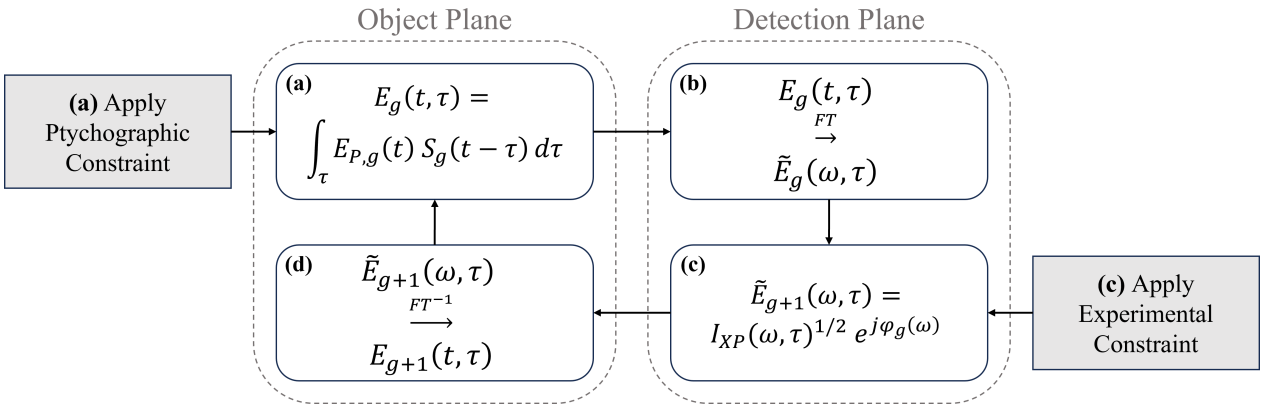

Figure 2.3 : Iterative Ptychographic Recovery Process.

The four distinct steps involved in ptychographic recovery are presented. For the initial iteration, $g = 0$, one of the two inputs, $E_{P,0}(t)$ and $S_0(t-\tau)$, is randomly selected, while the other is determined through the ptychographic constraint. In Step (a), the resulting electric field, $E_0(t,\tau)$, as it would occur on the object plane, is calculated from the initial inputs using Equation 2.3. In Step (b), the FT is taken to acquire the related spectrum on the detection plane, $\tilde{E}_0(\omega,\tau)$, using Equation 2.4. In Step (c), an experimental constraint is applied, where the phase of the previous guess is combined with the experimentally collected intensity to improve conversion. In Step (d), the inverse FT is acquired to calculate the electric field at the object plane for the next iteration, $E_1(t,\tau)$. As the process passes to the next iteration, $g = 1$, the ptychographic constraint is applied to determine the next iteration's input. These steps are repeated cyclically until convergence is achieved.

In Figure 2.3, a single cycle of the recovery process is presented. At Step (a), an estimate is made on either the electric field of the probe, $E_{P,g}(t)$, or the object, $S_g(t-\tau)$. For the first iteration, $g = 0$, this will be a random guess. For later iterations, it will be the estimate generated by the previous cycle. The ptychographic constraint (described momentarily) is used to determine the profile for the other input. The resulting theoretical electric field, $E_g(t,\tau)$, is calculated from Equation 2.3. In Step (b), the Equation 2.4 is applied to acquire the spectrum at the detection plane, $\tilde{E}_g(\omega,\tau)$.

Experimental Constraint. To assist in convergence, Step (c) uses the experimental intensity, $I_{XP}(\omega,\tau)$, to acquire the true spectral amplitude, $\sqrt{I_{XP}(\omega,\tau)}$. This amplitude is combined with the spectral phase component determined by that iteration to generate a modified spectrum, $\tilde{E}_{g+1}(\omega,\tau)$, described by Equation 2.5.

$$\tilde{E}_{g+1}(\omega,\tau) = \sqrt{I_{XP}(\omega,\tau)}\, e^{j\varphi_g(\omega)} \tag{2.5}$$

In Step (d), the inverse FT of Equation 2.5 is taken to recover the modified electric field for the next iteration, $E_{g+1}(t,\tau)$. However, a problem arises at this point, because the next guess, $E_{g+1}(t,\tau)$, is a function of the probe, $E_{P,\,g+1}(t)$, and the object, $S_{g+1}(t-\tau)$. Both of these quantities are unknown and solving for $E_{g+1}(t,\tau)$ returns us to a variation of the fundamental PRP. It is necessary to determine which probe and object generate the electric field, $E_{g+1}(t,\tau)$. This is achieved by applying what is known as the ptychographic constraint.

Ptychographic Constraint. The ptychographic constraint is applied to determine the new profiles of the electric field and switch vectors. This process is fundamental to both spatial and temporal ptychographies and so is well established in other works (Thibault et al., 2009; Leblanc, 2016; Bourassin-Bouchet et al., 2013; Lucchini et al., 2018). The process will be described in the context of the FROSt recovery. To determine the new profiles, $E_{P,\,g+1}(t)$ and $S_{g+1}(t-\tau)$, the sum of both profiles that most closely generates the last iteration's electric field is determined. For simplicity, we let $g = g+1$ such that : $E_{g-1}(t,\tau) \simeq \int_\tau E_{P,\,g}(t)\, S_g(t-\tau)\, d\tau$. The error generated on any given $E_{P,\,g}$ and S_g can then be restated as $\varepsilon_g(E_{P,\,g}, S_g)$ in Equation 2.6.

$$\varepsilon_g(E_{P,\,g}, S_g) = \int_t \int_\tau |E_{g-1}(t,\tau) - E_{P,\,g}(t)\, S_g(t-\tau)|^2\, d\tau\, dt \tag{2.6}$$

To apply the ptychographic constraint, the pulse and object parameters for iteration $g+1$ must be selected to minimize this error, ε_g. To resolve this mathematically, some assumptions must be made :

— The pulse, $E_{P,\,g}(t)$, and the object, $S_g(t-\tau)$, are independent and separable.

— The partial derivative of any derivable function A with respect to any complex variable z is :

$$\frac{\partial A}{\partial z} = \frac{1}{2}\left(\frac{\partial A}{\partial \mathrm{Re}(z)} - j\frac{\partial A}{\partial \mathrm{Im}(y)}\right)$$

— The partial derivative of any derivable complex function A in terms of a complex variable z satisfies the Cauchy-Riemann's Equations (Stein et al., 2007) :

$$\frac{\partial \mathrm{Re}(A)}{\partial \mathrm{Re}(z)} = \frac{\partial \mathrm{Im}(A)}{\partial \mathrm{Im}(z)} \quad \text{and} \quad \frac{\partial \mathrm{Re}(A)}{\partial \mathrm{Im}(z)} = -\frac{\partial \mathrm{Im}(A)}{\partial \mathrm{Re}(z)}$$

— Equation 2.6 can be developed as follows (where A^* is the complex-conjugate of A) :

$$\begin{aligned}|E_{g-1}(t,\tau) - E_{P,\,g}(t)\,S_g(t-\tau)|^2 = |E_{g-1}(t,\tau)|^2 + |E_{P,\,g}(t)\,S_g(t-\tau)|^2 \\ - E^*_{g-1}(t,\tau)\,E_{P,\,g}(t)\,S_g(t-\tau) \\ - E_{g-1}(t,\tau)\,E^*_{P,\,g}(t)\,S^*_g(t-\tau)\end{aligned}$$

Equation 2.6 shows that the error, ε_g, is dependent on the 2D electric field at the object plane, $E_{g-1}(t,\tau)$, as well as the product of the electric field, $E_{P,\,g}(t)$, and the switch, $S_g(t)$. To determine two vectors that produce the minimum error, the partial derivatives of Equation 2.6 are taken in terms of the electric field, $E_{P,\,g}(t)$, and the switch, $S_g(t)$. As shown in Equations 2.7, these partial derivatives can be set to zero to solve for the input vectors.

$$\begin{aligned}0 = \frac{\partial \varepsilon_g}{\partial E_{P,\,g}} &= 2\int_t\int_\tau |S_g(t-\tau)|^2\,E_{P,\,g}(t) - S^*_g(t-\tau)\,E_{g-1}(t,t-\tau)\,d\tau\,dt \\ 0 = \frac{\partial \varepsilon_g}{\partial S_g} &= 2\int_t\int_\tau |E_{P,\,g}(t+\tau)|^2\,S_g(t+\tau) - E^*_{P,\,g}(t)\,E_{g-1}(t+\tau,\tau)\,d\tau\,dt\end{aligned} \tag{2.7}$$

We can rearrange partial derivatives in Equations 2.7 to isolate the pulse, $E_{P,\,g}(t)$, and switch, $S_g(t)$. The process is developed for the former :

$$\begin{aligned}\int_t\int_\tau |S_g(t-\tau)|^2\,E_{P,\,g}(t)\,d\tau\,dt &= \int_t\int_\tau S^*_g(t-\tau)\,E_{g-1}(t,t-\tau)\,d\tau\,dt \\ \int_t E_{P,\,g}(t)\,\left(\int_\tau |S_g(t-\tau)|^2\,d\tau\right)\,dt &= \int_t\left(\int_\tau S^*_g(t-\tau)\,E_{g-1}(t,t-\tau)\,d\tau\right)dt \\ E_{P,\,g}(t)\,\left(\int_\tau |S_g(t-\tau)|^2\,d\tau\right) &= \int_\tau S^*_g(t-\tau)\,E_{g-1}(t,t-\tau)\,d\tau\end{aligned}$$

From the steps presented above, the Equations 2.8 and 2.9 are generated to apply the ptychographic constraint and determine either the next pulse, $E_{P,\,g}(t)$, or the next switch, $S_g(t)$, from the last electric field, $E_{g-1}(t,\tau)$.

$$E_{P,\,g}(t) = \frac{\int_\tau S^*_g(t-\tau)\,E_{g-1}(t,t-\tau)\,d\tau}{\int_\tau |S_g(t-\tau)|^2\,d\tau} \tag{2.8}$$

$$S_g(t) = \frac{\int_\tau E^*_{P,\,g}(t+\tau)\,E_{g-1}(t+\tau,\tau)\,d\tau}{\int_\tau |E_{P,\,g}(t+\tau)|^2\,d\tau} \tag{2.9}$$

Equations 2.8 and 2.9 are then used interchangeably to apply the ptychographic constraint. This ensures that both predicted vectors can combine to form the closest approximation of the 2D electric field, $E_{g-1}(t,\tau)$. Using the ptychographic constraint means only a single approximation is made on either the probe or the object profiles at each iteration. Equation 2.8 or 2.9 is then used in conjunction with the electric field determined from Step (d) to produce the approximated input

for the following iteration. This ensures that the combined result of the probe and object used in a single iteration always produce the known trace.

Simply put, at iteration $g = 0$, the probe and object, $E_{P,0}(t)$ and $S_0(t-\tau)$, are both randomly selected to reach a first estimate of the spectrum, $\tilde{E}_0(\omega,\tau)$. The probe, $E_{P,0}(t)$, is then combined with the output of Step (d), $E_1(t,\tau)$, to determine the approximated input for $g = 1$ using Equation 2.9. As a result, for $g = 1$, the inputs are actually $E_{P,0}(t)$ and $S_1(t-\tau)$ for the probe and object, respectively. Equivalently, once $E_2(t,\tau)$ is determined from $S_1(t-\tau)$ in Step (d) of $g = 1$, Equation 2.8 is applied to produce $E_{P,2}(t)$. This is repeated until convergence, where each iteration requires only one of the two inputs to be approximated, while the other is determined through the ptychographic constraint. While this does add complexity to the implementation, it also ensures that the recovery converges successfully by further isolating the subset of valid solutions.

Calculating Loss. At each iteration, the loss is calculated after applying the experimental constraint at Step (c). The modulated complex TFR, $\tilde{E}_g(\omega,\tau)$ calculated from Equation 2.5, can be used to determine the modulated intensity : $I_g(\omega,\tau) = |\tilde{E}_g(\omega,\tau)|^2$. The difference between the predicted TFR intensity, $I_g(\omega,\tau)$, and the experimental TFR intensity, $I_{XP}(\omega,\tau)$, is determined using Equation 2.10 at each iteration.

$$Err_g = \left(1 - \frac{\left(\sum\limits_{\tau,\omega} I_{XP} \cdot I_g \right)^2}{\sum\limits_{\tau,\omega} I_{XP}^2 \sum\limits_{\tau,\omega} I_g^2} \right)^{1/2} \tag{2.10}$$

This loss is used to determine the optimization gradient used in the convergence. At the earliest iterations, the error, Err_g, will be large, resulting in a steep optimization map. As the predicted intensity, $I_g(\omega,\tau)$, approaches the true intensity, $I_{XP}(\omega,\tau)$, the loss approaches zero, $Err_g \xrightarrow{g \to G} 0$. The use of both experimental and ptychographic constraints exploits the theoretical understanding of the problem to ensure the loss function converges in a minimum of iterations. The final output, after $g = G$ iterations, will be a very close approximation to the probe and object profiles. The final approximation is evaluated by determining the total error between the approximated trace, $I_G(\omega,\tau)$, and the experimental trace, $I_{XP}(\omega,\tau)$, using Equation 2.10. As with all numerical processes, if this error does not approach zero over the course of many iterations, the algorithm is considered to be diverging and a valid solution will not be isolated. However, if done successfully, the output profiles, $E_{P,G}(t)$ and $S_G(t-\tau)$, can be taken as the most correct solution.

2.3 Artificial Intelligence

The goal of this work is to replicate the ptychographic process described in Section 2.2 using machine learning tools. Before proceeding to the methodology in Section 3, it is necessary to understand the fundamental mathematical functions of machine learning (Brunton et al., 2017; Goodfellow et al., 2016; Bishop, 2006). NNs can be conceptualized as layers of matrices that each perform mathematical operations on an input. These layers can be organized in series or in parallel and can be combined to replicate mathematical equations using linear algebra operations. When describing the architecture of a NN, it is the physical organization and the operations performed by these layers being defined.

2.3.1 Building a Neural Network

Layers are defined by the mathematical operation they perform on the input. Each neuron in a layer (i.e., a cell in the matrix) has a weight that describes a parameter of the operation. In a fully-connected NN, like the ones used in this work, each neuron in a layer will operate on every cell of the input matrix. As a result, the operation of the entire layer can be described as a function of the weights of each neuron. In the following Section, the layers used in this work are described.

Convolutional Layers. Convolutions are a fundamental operation used in NNs. In 1D, a convolution is simply a sliding window operation as described by Equation 2.11, where the output, $y_o(t)$, is the convolution of the input, $y_i(t)$, and the filter function, $W(t)$, displaced by τ at each summation (Mallat, 2016; Goodfellow et al., 2016).

$$y_o(t) = \sum_{\tau} y_i(t)\, W(t-\tau) \tag{2.11}$$

Convolutions can be extended into the 2D, where the vectors are instead matrices. As a result, the filter function becomes a matrix that slides across pixels in an input image to generate feature maps. In fact, the filters applied in the Figure 1.7 would be considered single convolutional layers, where the filter functions were the 3-by-3 kernel matrices used for each filter. Changing the weights of each neuron (i.e., the coefficients) in the kernel altered the output feature map (as was shown by the different maps generated in Figure 1.7).

The process to generate the feature maps in Figure 1.7 can be mathematically described by Equation 2.12. An input image has a shape of $(M; N_{i,x}, N_{i,y})$, where $N_{i,x}$ and $N_{i,y}$ are the pixel dimensions in (x, y) of each channel M. For clarity, N indicates the dimension, while n represents the matrix index (such that $n \in [0, N-1]$ for (x, y)). A black-and-white image will have a single channel, $M = 1$, while a colored image would have $M = 3$, where red, green, and blue are each

their own channel (Gonzalez et al., 2018). A convolutional layer must then have P filters of shape $(N_{k,x}, N_{k,y})$, where $(N_{k,x}, N_{k,y})$ is the kernel size in (x, y), respectively. Each filter P operates as a sliding window on each input channel M to produce a total of $P\,M$ output channels. The output from the convolutional layer will then be in the form $y_o(P\,M; N_{o,x}, N_{o,y})$, where each channel $P\,M$ contains a feature map with a pixel size of $N_{o,x}$ and $N_{o,y}$ in (x, y), respectively. A convolutional layer will produce several feature maps, $y_o(P\,M)$, which are related to the input, $y_i(M)$, through the windowing function, $W(P, M)$. Equation 2.12 demonstrates this relation, where the windowing function, $W(P, M)$, is shifted pixel-wise across the input map, $y_i(M)$, to produce a set of output maps, $y_o(P\,M)$ (Mallat, 2016; Goodfellow et al., 2016).

$$y_o(P\,M; n_{o,x}, n_{o,y}) = \sum_{\substack{n_{k,x} \\ n_{k,y} \\ P,\,M}} y_i\left(M; n_{i,x} + n_{k,x}, n_{i,y} + n_{k,y}\right)\, W(P, M; n_{k,x}, n_{k,y}) \tag{2.12}$$

The filters, $W(P, M)$, are defined by their weights (coefficients), which can be set programmatically (an aparametric NN) or determined through a training procedure (a parametric NN). For the aparametric filters used in the Figure 1.7, a kernel size of $N_k = 3$ is used with the matrix weights presented in the inlays. The NN used to generate the feature maps in Figure 1.7 had a single convolutional layer with $P = 3$ filters, which produced 3 output feature maps from the single channel input. As a result, the input image is propagated through these filters independently of one another (i.e., parallelized). In many cases, it can be advantageous to further disassemble the input image into more detailed feature maps. This can be achieved by organizing convolutional layers with multiple filters in series, such that the feature maps output from the first layer is passed to the next for further processing. Each successive convolution extracts additional details from the input image, causing it to be decomposed into its smallest features.

The impact of parallelized and serialized convolutions is demonstrated in Figure 2.4. A simple NN, with three convolutional layers in series, is used. Each layer has three filters ($P = 3$), which is the equivalent of having three parallelized convolutions at each layer. Overall, the NN architecture has nine convolutions, three in each layer. The filter kernels are 3-by-3 matrices ($N_k = 3$) and the activation function is linear (more on activation functions in the following paragraphs). The input is a black-and-white image of a FROSt trace, shown in Figure 2.4a. When the input is propagated through the first layer of the NN, it is convolved with the three filters to produce three feature maps, which are presented in Figure 2.4b. These three filters were applied to the same input, making them independent of each other (effectively parallelized). The feature maps in Figure 2.4b are further decomposed by the second layer of the NN to produce Figure 2.4c. These feature maps are now the convolution of a convolved input, allowing them to extract more complex features than the first layer. This is repeated for a third layer to generate the images in Figure 2.4d. These maps have now passed

through three consecutive convolutions, which results in an increasingly complex set of features to be seen. Figure 2.4 shows that physical architectures of layers will impact the information extracted from the input image. An effective NN will combine aspects of both parallelized and consecutive layers to extract fine patterns from the input images. However, this must be done carefully to avoid arbitrarily extracting noise after many convolutions. This must be carefully considered when designing architecture for an AI application.

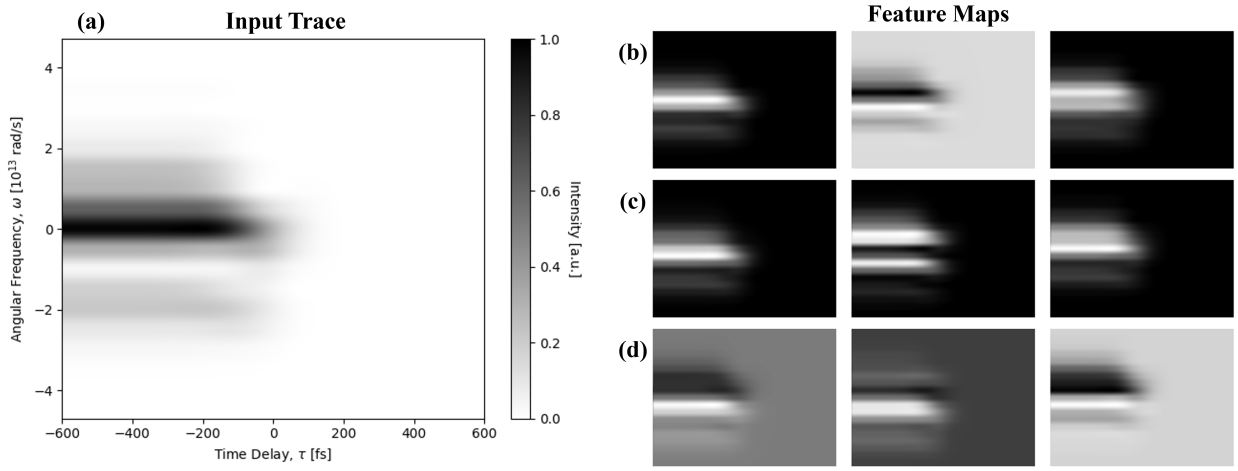

Figure 2.4 : Serialized Convolutions of a FROSt TFR.

Feature maps generated from a NN with 3 layers of convolutions are presented, where each layer has three filters ($P = 3$) and a kernel size of $N_k = 3$. A black-and-white FROSt trace ($M = 1$), (a), is propagated through the NN described. In (b), the feature maps output from the first layer of filters are presented (where each map is generated from one of the three filters in the layer). The maps in (b) are next propagated through a second, identical layer to produce the feature maps in (c). This is repeated for a third layer to generate the feature maps presented in (d). The images in (d) have therefore passed through three consecutive layers (for a total of nine convolutions) while those in (b) have passed through only one layer of three convolutions. A larger number of convolutions results in the further decomposition of the input image, which leads to finer details being extracted.

For simplicity, Equation 2.12 assumed that some parameters were set to 0 or 1, which will now be explored. When performing a convolution, the stride, s, describes the size of the offset between the filter and the image at each step. Therefore, s indicates the number of pixels by which the filter is shifted for each step. Additionally, the pixel dimensions of the output changes proportionally to the stride and size of these filters. As such, it is convenient to use a padding parameter, p, which forces the desired output shape. This is often necessary to ensure successive matrix operations are conformable (i.e., the matrices are of the correct shape to perform the desired mathematical operation together). The size of the output channels are calculated through Equation 2.13, where N_o is the output size, N_i is the input size, p is the padding parameter, s is the stride, and N_k is the kernel size (Goodfellow et al., 2016). This can be used equivalently for (x, y) dimensions.

$$N_o = \left\lfloor \frac{N_i + 2p + N_k}{s} \right\rfloor + 1 \tag{2.13}$$

In Equation 2.12, the shift term was simplified to $+n_k$ in each dimension. This was due to the selected parameters for the stride, $s = 1$, and padding, $p = 0$. When considering other possible values of s and p, the shift term becomes instead $+s\,n_k - p$ for all dimensions. It is important to note that these kernel parameters are often set programmatically, while the neural weights are generally determined through a training procedure. In this work, we rely on architectures with pre-established stride and padding values, as the study of these parameters is not within the scope of this work (Zahavy et al., 2018).

Flatten and Dropout Layers. The flatten operation is commonly used after a number of convolutions. After a few layers, the output will be a very large matrix with high dimensionality, which will carry a significant number of feature maps. The use of a flattening layer brings all the features from each map into a single channel, effectively equalizing all the pixels. Once the features are extracted from the maps, this large single channel is passed to dense layers to form connections. This is done to bridge the difference in matrix operations between convolutional and dense layers. Often, a dropout layer is included after the flattening layer and before the dense layers to remove a percent of features in the matrix (Brunton et al., 2017; Goodfellow et al., 2016). The percentage of features to remove, as well as which features to remove, are considered hyperparameters that alter the information of the output vector. This is another trainable parameter that tends to have a significant effect on over-fitting, where including a small dropout rate can make the AI more robust to data outside the training set.

Dense Layers. The large output vector containing all the features is then passed to a dense layer, which simply performs a linear transform relative to the neural weights, W, of the layer, as described by Equation 2.14. The input, y_i, is a flattened vector of N_i pixels, and y_o is the output of N_o pixels. The neural weights, $W(n_o, n_i)$, is a matrix of (N_o, N_i), defined by the relation : $W(n_o, n_i) = A(n_o, n_i) + b(n_o)$, where A is the magnitude and b is the bias for each neuron, n (Mallat, 2016; Goodfellow et al., 2016).

$$y_o(n_o) = \sum_{n_i} y_i(n_i)\, W(n_o, n_i) \tag{2.14}$$

Dense layers take the individual features from the flattened vector and make linear connections between them. The neural weights are trained to adjust the contribution from each pixel to achieve the desired output. These relations are fundamentally linear, which limits the complexity of the functions being mapped, but nonlinearities can be introduced through activation layers.

Activation Layers. Mathematically, an activation layer executes the operation, $y_o = f\,(y_i)$, where y_i is the input, y_o is the output, and f is the activation function (Brunton et al., 2017). These layers can be used to introduce nonlinearities to the relations generated by the convolutio-

nal and dense layers. A linear activation function can be applied, though this continues to limit the model to the linear form : $A(n) + b$. This is insufficient to model more complex relations, which require the application of nonlinear activation functions, such as rectified linear unit (ReLU) ($\text{ReLU}(n) = \max(0, n)$), sigmoid ($\text{sigmoid}(n) = 1/(1 + e^{-n})$), and tanh $\left(\tanh(n) = \frac{e^{2n}-1}{e^{2n}+1}\right)$ (Brunton et al., 2017). These allow the training process to extract nonlinear relations between the input and output. This is an important quality, as very few real world cases are simple enough to model using only linear models. Activation functions can be used between any type of layer, including convolutional and dense.

Architecture. With the function of each individual layer understood, they are now considered within the context of a larger model. It was established that placing convolutional layers in series and in parallel produces different outputs (as in Figure 2.4). Assembling these layers in a particular order allows an image to be dissected into many parts, which can then be related to each other by the dense layers. This must be done with consideration to matrix conformity, computational cost, and the minimum feature size.

Consider how convolutional layers produce an output for each filter in the layer (such that a convolutional layer of P filters operating on M channels produces $P\,M$ output channels). Within a larger model, the next layer must then be the correct shape to operate on the given number of inputs. Putting many convolutional layers in series will quickly create an exponentially large matrix. Not only can this distort the original input beyond any relevant patterns, but computationally becomes difficult to manage. Conversely, a single convolutional layer with few filters will not provide significantly more information than the original image. Similarly, the organization of connections, dropouts, and neurons on dense layers will alter the relationships extracted by the overall model. The impacts of this organization of layers has been widely studied (Alzubaidi et al., 2021; Mallat, 2016; LeCun et al., 2015; Szegedy et al., 2014) and this work will rely heavily on established, successful architectures for the given application (Zahavy et al., 2018).

2.3.2 Training Parameters

Just as the architecture has high-level parameters that are optimized through training, the training process itself has similar parameters that can alter its behavior (Goodfellow et al., 2016; LeCun et al., 2015). Poorly selected parameters can lead to ineffective convergence. Some important parameters are discussed below, as some will be considered after the variables of interest are optimized.

Loss Function. During SL, when the architecture makes a prediction, it is compared to the label using a loss function. Many loss functions exist (primarily, these are mean absolute, square mean, L1 and L2 (Goodfellow et al., 2016; Li et al., 2025)), which can used for many applications. The loss

can be considered as the distance from the prediction to the true label, which is used during training to generate a gradient and determine the optimization path (see Optimizers for more) (Li et al., 2025; Kingma et al., 2015). Error is used to describe the deviation between prediction and label, while the loss describes the impact this error has on the gradient. As a result, the loss function and error are very closely related and will often be used interchangeably. After calculating the prediction error on each data point, the global loss is back-propagated through the model to update weights for the next iteration. Theoretically, as training continues, the loss function alters the weights and the prediction error is decreased. The reduction in error in turn relates to a smaller difference between the current and optimal weights (i.e., a smaller loss). This is how the loss function governs the training process to achieve convergence.

Metrics. Metrics are used to evaluate the quality of a trained NN. Metrics have no impact on the training process itself but are convenient for comparing the quality of their behaviors. Oftentimes, the metrics are similar to the loss function used to calculate the back-propagation error, though it is not related. Metrics are instead used after training is completed to compare NN and their predictions on a common dataset. These will generally be the mean absolute error (MAE) or the mean squared error, but can be others (Goodfellow et al., 2016).

Optimizer. The optimizer is responsible for determining the next step along the gradient generated by the loss function. Like in numerical methods, the step must carry towards the gradient's global minimum in order to achieve convergence on the optimal weights (Brunton et al., 2017). The step determines the magnitude and direction of change applied to the weights of each neuron at every iteration. Optimizers are available from established libraries such as `keras`. Adam, a derivative of the stochastic gradient descent method, is a robust optimizer that is often used in machine learning since it requires little computational time or memory (Kingma et al., 2015).

Learning Rate. The learning rate is a parameter of the optimizer and it determines the step size taken by the optimizer (which identifies the direction of the step). The optimization process is a study of its own, the results of which are made available through optimizers such as Adam. A dynamic step-size is used to select the best next step depending on not only the gradient but its momentum as well (Kingma et al., 2015). This leads Adam to a more efficient optimization when compared to other algorithms.

Batches. The iterations between back-propagations are called batches. Batching is used in computing processes to reduce latency. Training datasets are extremely large and holding the entire dataset in memory at once is not only costly for the processor but impacts the training optimization. Breaking the dataset into smaller batches of data reduces this issue and also means that

the loss is calculated on smaller groups of data at a time, which can improve resistance to small variations in data. Batch sizes are often selected as powers of 2 for computational efficiency.

Epochs. After the entire dataset has been batched, it is often beneficial to repeat the process again. A single epoch has processed the entire dataset once, regardless of the batch size used. Often, a validation step is included between epochs, where the quality of the training is evaluated on untested data. Repeatedly passing the training data improves the optimization process significantly but can also lead to over-fitting if over-exhausted.

Some of the above parameters can be selected depending on the application of the design. Optimizers and loss functions are generally determined in such a way. However, hyperparameters are difficult to sweep simultaneously and are often optimized through the use of objective libraries such as `Optuna` (Goodfellow et al., 2016). The parameters selected, as well as the optimization process used for this work, will be detailed in Section 3.

3 METHODOLOGY

The methodology followed in this work is heavily influenced by that of Zahavy et al. (2018) and the ptychography process described in Section 2.2. The method used to develop the dataset, the architecture, and training procedures is established in this Section. Much of the code written in this methodology can be found in Section 7.1.

3.1 Dataset Generation

Generating a dataset of experimental FROSt traces is not possible for two reasons. First, it would be impractical from a resource standpoint to collect a dataset of the necessary size. Second, the label in this case is the electric field of the pulse and this is fundamentally not known (see Section 2.1 on the PRP). Instead, a dataset of simulated electric fields and their FROSt traces are used to train the NN initially. We first describe the overall process, before proceeding to a more detailed procedure of the random phase generation in Section 3.1.1.

Electric Field : $E_P(t)$. The simulated pulses must describe a wide variety of possible spectral phases. An FTL Gaussian pulse is generated, with an amplitude of $E_{0,\,\mathrm{FTL}}$, and a full-width half-maximum (FWHM) temporal spread of $\sigma_{\mathrm{FTL}} = \mathrm{FWHM}/(2 \times \sqrt{2 \times \ln 2}$. The FTL pulse, $E_{\mathrm{FTL}}(t)$, is described by Equation 3.1.

$$E_{\mathrm{FTL}}(t) = E_{0,\,\mathrm{FTL}}\, e^{-0.5\left(\frac{t}{\sigma_{\mathrm{FTL}}}\right)^2} \tag{3.1}$$

Variants of the FTL pulse are generated by inducing random, continuous spectral phases, $\varphi(\omega)$ (described in Section 3.1.1). The phase is added to the spectrum of the FTL pulse in Equation 3.1 to generate the newly chirped pulse, $\tilde{E}_{\mathrm{C}}(\omega)$, as described by Equation 3.2.

$$\tilde{E}_{\mathrm{C}}(\omega) = \int_t E_{\mathrm{FTL}}(t)\, e^{-j\omega t}\, dt\, e^{j\varphi(\omega)} \tag{3.2}$$

By inversing the FT and returning to the time-domain, the complex amplitude of the newly chirped pulse, $E_C(t)$, is described by Equation 3.3, where $\phi(t)$ is the temporal phase function related to the spectral phase function, $\varphi(\omega)$. The pulse has an amplitude of $E_{0,\,\mathrm{C}}(t)$, and a modified spread of σ_{C}.

$$E_{\mathrm{C}}(t) = E_{0,\,\mathrm{C}}(t)\, e^{-0.5\left(\frac{t}{\sigma_{\mathrm{C}}}\right)^2}\, e^{j\phi(t)} \tag{3.3}$$

The above procedure is repeated on a large scale by generating N random phase functions using the process described in Section 3.1.1 to produce N chirped electric fields, $E_C(t)$. These make up the simulated chirped pulses used in the dataset and will be referred to interchangeably with the probe pulses, $E_P(t)$, for the rest of this work. The code used to generate the FTL then apply the phase can be found in Figures 7.1 and 7.2, respectively.

Switch : $S(t)$. It is also necessary to simulate a switch functions similar to the ones generated by photo-excitation in various materials. A Gaussian curve, as seen in Equation 3.4, is used to approximate the inversion. The switch, $S(t)$, is generated temporally, t, with a maximum amplitude of S_0 and a spread of σ_{S}. This quantity is then integrated over the temporal window, Dt, to achieve a one-sided increasing curve.

$$S(t) = \left| \int_{-Dt/2}^{Dt/2} S_0 \, e^{-0.5\left(\frac{t}{\sigma_{\mathrm{S}}}\right)^2} \, dt \right|^2 \tag{3.4}$$

A range of temporal durations can be generated by varying the FWHM spread, $\sigma_s = \frac{\mathrm{FWHM}}{2\sqrt{2\ln 2}}$. Two training sets will be compared based on this parameter. In the first case, a single switch function with a FWHM of 125 fs is used to generate all the traces in the dataset. In the second case, a range of temporal durations are used by varying the FWHM spread from 100 fs to 400 fs with a step size of 1 fs to generate a total of 300 unique switches. In both cases, the generated Gaussian curves are normalized, with a minimum of 0 and a maximum of 1 This normalization is applied in the dataset to ensure the output of the NN will be constrained by physical principals. The code used to generate this curve can be found in Figure 7.3.

This procedure generates integrated Gaussian curves that loosely approximate the switching behavior of population inversion in materials. Modeling the specifics of the population inversion dynamic is studied elsewhere (Amstrup et al., 1993; Satou et al., 2011; Gornushkin et al., 2017) and is not the focus of this work. It is seen in past works (Leblanc et al., 2019) that FROSt is independent of the experimental switch profile, because the ptychographic recovery process solves for both the switch and electric field profiles. The durations are included to generate some additional complexity, but it must be noted that the NN will not be trained on a diverse array of switch profiles.

FROSt Trace : $I(\omega, \tau)$. The simulated electric fields and switches were used in combination to generate FROSt traces. An aparametric NN, referred to as FROStNET, is used to generate a trace, $I(\omega, \tau)$, from the electric field, $E_P(t)$, and the switch, $S(t)$, by mapping Equation 1.5. This was done using a NN, instead of traditional computing methods, to ensure integration into the training procedure. Programmatically, generating the FROSt trace is equivalent to taking the STFT using the switch function as the window.

This is implemented with minimal computational costs by using `tensorflow` matrix operations and an aparametric NN. This architecture is presented in Figure 7.4. The layers of this model each perform an operation seen in Equation 1.5 and are structured to apply them in the necessary order. For example, a layer can represent an FT if its weights are set to be the Fourier matrix. As a result, the output of this layer is related to its input through a linear transform governed by the Fourier relation (i.e., an FT). Equivalently, the switch is applied through matrix multiplication by storing a copy of each temporally delayed vector in a single 2D matrix (achieved using the `numpy.sliding_window_view` command). This process is visualized in Figure 3.1 where the FROStNET architecture is presented. The probe, $E_P(t)$, and switch, $S(t)$, are input. The probe is multiplied by the switch, which was shifted by τ. The FT is taken to result in the FROSt trace, $I(\omega, \tau)$, shown.

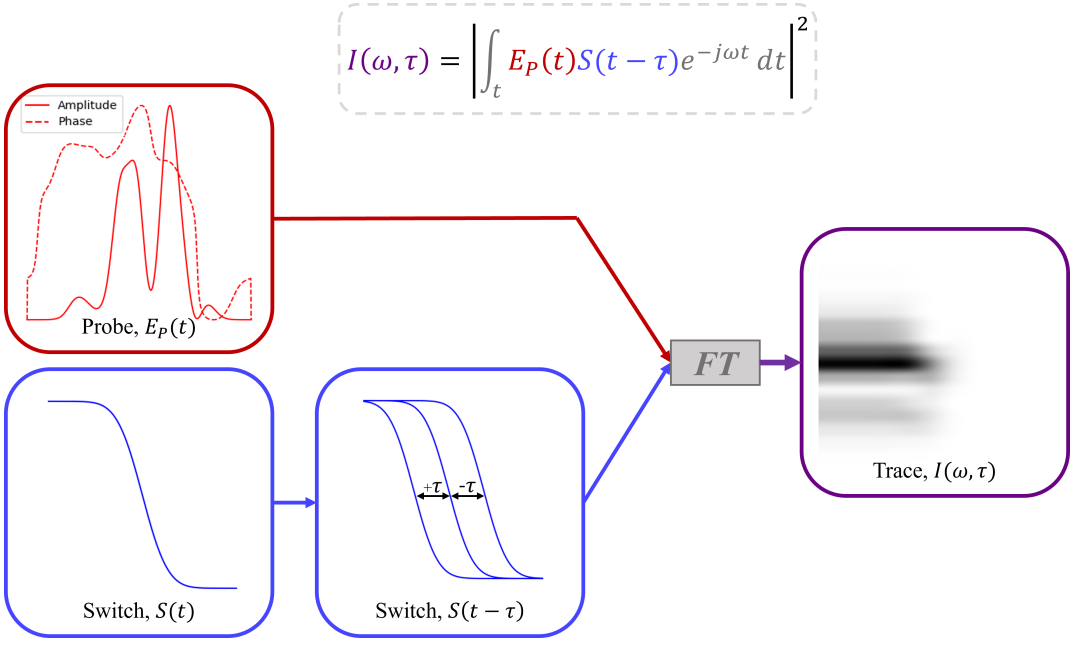

Figure 3.1 : FROStNET Architecture.

The NN architecture used to generate the FROSt traces is shown. The inputs are a complex pulse, $E_P(t)$ (in red), and the switch, $S(t)$ (in blue). The theoretical FROSt Equation (Equation 1.5) is presented with colors relating each operation to a step of FROStNET. First, in blue, the switch is shifted by varying τ. The probe and the shifted switch functions are then multiplied and an FT is applied (in gray). The output is a FROSt trace, $I(\omega, \tau)$ (in purple).

The results of the described process is demonstrated in Figure 3.2 where the FTL and one random chirped pulse is presented. The temporal and spectral profiles of both pulses are shown. Additionally, the switch is shown, as well as the traces generated for each pulse using FROStNET. It is noted that the pulses have very different temporal profiles and traces, though they share the spectral bandwidth. All pulses generated in a single dataset will share the same fundamental bandwidth.

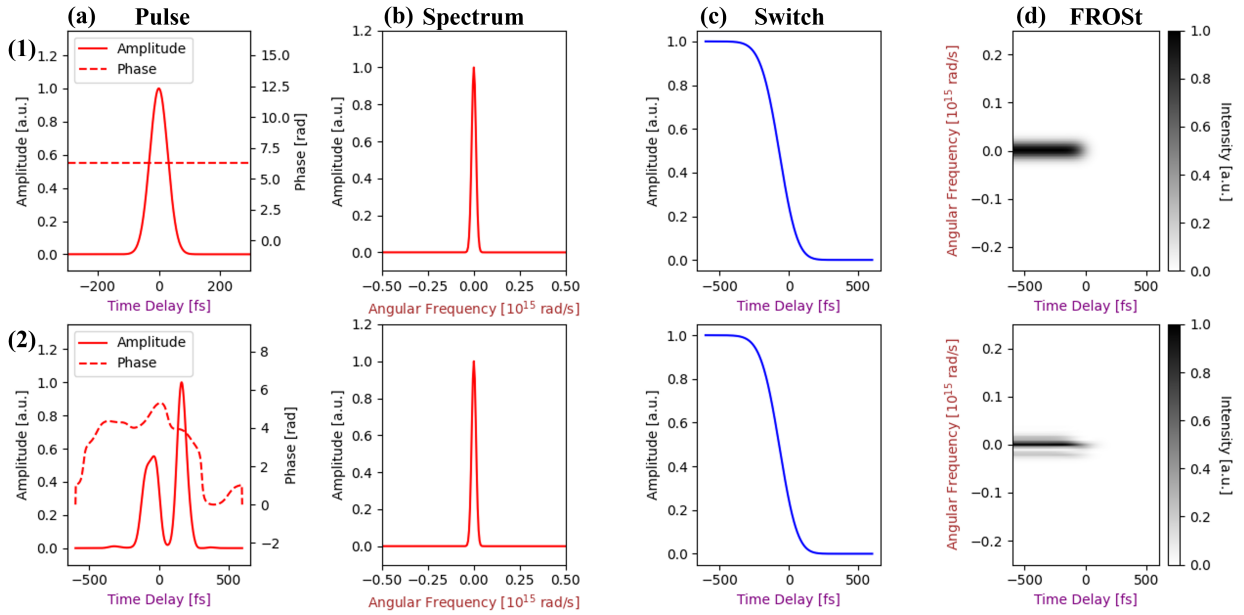

Figure 3.2 : Temporal and Spectral Profiles of an FTL and a Chirped Pulse.

In (a), the temporal profile of the FTL (1) and a chirped (2) pulse are shown. The FTL pulse has a FWHM of 100 fs**. The spectrum of both pulses is shown in (b). The switch profile, shared by both cases, is presented in (c). Finally, in (d), the FROSt traces are shown. The chirped case in row (2) exhibits spectral beating in the temporal profile.**

3.1.1 Random Phase Generation

The random spectral phase functions were generated using a polynomial described by Equation 3.5, where R_o is a randomly generated coefficient for each order o. Three parameters were varied to generate a variety of phases : the order, o, the number of monomial terms, N_m, and the magnitude of each coefficient, R_o.

$$\varphi(\omega) = \sum_{o=O-N_m+1}^{O} R_o\,\omega^o \tag{3.5}$$

The spectral functions were generated with orders, o, from 1st through 4th degree polynomials (where O is the maximum order in any given polynomial such that $o \in [O-N_m+1, O]$). Figure 3.3 shows the temporal profile of four pulses, each demonstrating the behavior generated by increasing the order of the phase polynomial. Each pulse shown in Figure 3.3 was generated from the same FTL pulse described in Section 3.1. First-order polynomials produced a constant temporal phase, which simply shifts the temporal profile. Quadratic functions cause pulse broadening (and eventual splitting). Cubic functions produce spectral beating, while fourth order functions result in complex behaviors known as tesseractic dispersion. Higher orders are considered negligible in the Taylor series expansion and therefore were not included (Diels et al., 2006). The described methodology

allows the generation of near infinite pulses with a broad variation in phase behaviors, as shown in Figure 3.3.

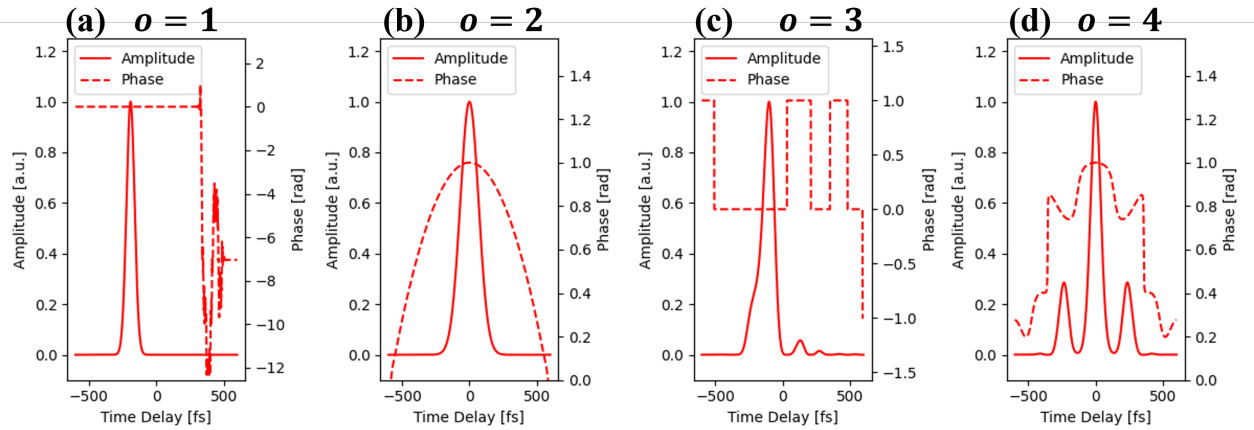

Figure 3.3 : Electric Fields Exhibiting Monomial Phases.

The temporal profiles of four pulses (amplitude in continuous and phase in dashed) generated using the process described. Each has undergone dispersion of single order ($N_m = 1$) with various polynomial coefficients, R_o, resulting in behaviors unique to each order, o. In (a), group delay is induced by a spectral linear phase ($o = 1$ with $R_o = 1$). In (b), the pulse broadens and splits due to the spectral quadratic phase ($o = 2$ with $R_o = 0.1$). In (c), spectral beating is generated by a cubic phase ($o = 3$ with $R_o = 1$). In (d), a spectral quartic phase ($o = 4$ with $R_o = 0.1$) induces a combination of spectral beating and broadening simultaneously.

The pulses above exhibit a single order of dispersion at a time because they were generated with a single monomial term, $N_m = 1$. However, under experimental conditions, it is rare that pulses will experience only a single order of dispersion at a time (e.g., propagation through a hollow core fiber would induce both first and second order dispersion). By including a parameter that varies the number of terms in the polynomial, N_m, Equation 3.5 can generate multi-term polynomials (when $N_m > 1$) that produce composite behaviors in the temporal profile. For multi-term polynomials, the magnitude of each order's coefficient, R_o, is varied to adjust the contribution from each degree of dispersion. The random generation of R_o is constrained, where higher orders are more heavily limited. As a result, Equation 3.5 can produce a variety of unique spectral phase functions which reproduce experimental conditions with high fidelity.

3.1.2 Dataset Quality Assurance

Once a large dataset has been generated using the above procedure, it is important to ensure it is of high quality. The quality of the dataset contributes to the effectiveness of the convergence and is widely considered among the most important elements when training an AI (Alzubaidi et al., 2021; LeCun et al., 2015). Two main considerations are used to ensure this is achieved in this work : experimental constraints and ambiguity removal.

Experimental Constraints. Experimental constraints are applied to ensure that the pulses generated describe physically realistic conditions while remaining programmatically valid. It is known that certain properties of the electric field must be true for a pulse to physically exist. For example, it is well understood that the intensity, $I(t)$, must be a non-negative entity ($I(t) = |E(t)|^2 \geq 0$). This is experimentally a trivial constraint, since the sign of a pulse can be inverted mathematically without affecting other properties of the pulse, but must be considered programmatically. By introducing these experimental constraints, the phase functions in the dataset are ensured to be accurate reflections of the pulses generated through experimental conditions. The three parameters constrained for this purpose are group delay, energy dispersion, and instantaneous frequency.

From Figure 3.3, it can be seen that the chirped pulses have spread and shifted within the window, comparatively to the FTL pulse. In the case of Figure 3.3a, group delay has shifted the temporal profile to the edge of the scanning window. Experimentally, when phases induce a linear shift and this occurs, the scanning window is simply adjusted to recenter the pulse. Programmatically, however, this cannot be achieved and the pulse will shift partially or entirely out of the window. It is important to avoid the inclusion of pulses that have shifted from the window (i.e., an empty vector). The phase functions are therefore constrained by evaluating the relative shift they induce in the electric field. This behavior is simple to isolate programmatically when inducing primarily linear dispersions, where the pulse still exhibits a single peak. However, when multi-term phases with many orders of dispersions are induced, the temporal profile changes, such that a single, clear peak may no longer exists. In these cases, instead of simply evaluating the shift of the main peak, the energy dispersion caused by the phase is constrained to ensure the pulse is conserved within the window.

A third important parameter, instantaneous frequency, is constrained to further improve the quality of the dataset. Instantaneous frequency, $f_{\text{inst}}(t)$, is described as the temporal rate of change of a pulse, such that $f_{\text{inst}}(t) = \frac{1}{2\pi}\frac{d\phi}{dt}$, where $\frac{d\phi}{dt}$ is the time derivative of the temporal phase. Physically, there is a maximum instantaneous frequency that can exist in an electric field, due to the fundamental relative nature of phase. To respect this property in the dataset, the phase functions are scaled to ensure the instantaneous frequency is limited to a range of $-\pi \leq \Delta\phi(t) \leq \pi$. This ensured that the pulses generated in the dataset remain within the scanning window and finite.

Ambiguity Removal. Ambiguities are cases where two different traces produce trivially similar pulses. Inclusion of these in the dataset leads to poor convergence during the training procedure. It is necessary that each trace in the dataset has a single correct electric field to approach (i.e., the system has a single minimum to converge towards). However, it was established in Section 2.1 that, due to the PRP, a FROSt trace will have many possible solutions. Some solutions are related through time-space symmetries and can be considered trivial, while others will be entirely distinct. A dataset that includes trivially similar traces, each with distinctly different labels, will cause instability and divergence in the training. An example of an ambiguous pulse is shown in

Figure 3.4 where a single FROSt trace is related to four possible time profiles, each defined by an important time-space ambiguity that is described further below.

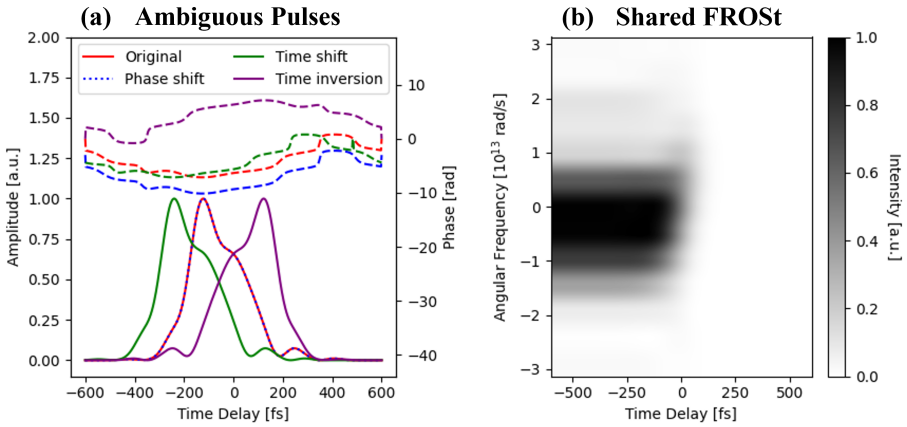

Figure 3.4 : Examples of Time-Space Ambiguities.

In (a), four complex electric fields are presented (amplitude in continuous, phase in dashed). The original pulse is shown in red, whereas the three others are related to it through an ambiguity. A relatively constant phase shift is shown in dotted blue, a time shift in green, and a time inversion in purple. In (b), the FROSt trace that relates to all four pulses is shown. These are trivially ambiguous pulses that generate a single FROSt trace, which should be avoided in the dataset.

Because of the relative nature of phase, a constant phase shift is mathematically insignificant. This is demonstrated by the dotted blue electric field, which overlaps identically with the original red pulse, presented in Figure 3.4a. Consider two pulses with phase functions offset by a factor of π. Their FROSt traces are identical and, experimentally, this difference is meaningless because it is a relative parameter. The same can be said for phases shifted by multiples of $N\pi$. Phases related by constant phase shifts are therefore removed from the dataset.

Similarly, a temporal shift in the phase is simply a linear dispersion that induces group delay (where the temporal profile of the pulse is unchanged but shifted). If the window is not re-centered around the delayed pulse, there is an equivalent shift in the trace. However, because experimentally collected pulses are centered in their windows, this is considered a trivial ambiguity. This is demonstrated in Figure 3.4a, where the green electric field is shown to be identical to the original red. Therefore, the experimental traces acquired from a pulse, regardless of group delay, will be identical. This must be accounted for by removing data points related by a constant time shift.

Lastly, time inversion is considered an important symmetry in pump-probe experiments. These experiments, FROSt included, use the relative offset between pump and probe to measure the intensity. Since the direction of movement is arbitrary, the pulse can be recorded either forward or backwards in time. These two pulses will be mirrors of each other, in amplitude and phase, and both are considered equally valid, as is shown by the original red and time inverted purple pulses of Figure 3.4a. The inversion and conjugation of each phase component is evaluated to ensure the removal of this ambiguity.

As it is seen in Figure 3.4, these ambiguities result in related temporal profiles and identical traces. Training an algorithm to overcome a single ambiguity would result in global instabilities, meaning they should be removed from the dataset. Ultimately, a single case is used to represent a class of related pulses.

3.2 Multi-Resolution DNNs

The architecture used to extract the pulse and switch from the traces was was directly inspired by the architecture used in Zahavy et al. (2018). This architecture uses 2D convolutional layers followed by dense layers, all with ReLU ($\text{ReLU}(n) = \max(0, n)$) activations (Zahavy et al., 2018; Szegedy et al., 2014). It is referred to as a multi-resolution (MR) architecture due to its use of parallelized convolutions with varying kernel sizes. This is a benefit, as features of different sizes will be extracted into the maps by each kernel. The entire model is made up of repeating MR blocks, which will be presented before proceeding with the description of the entire model.

Figure 3.5 presents the architecture of a single MR block, assuming the input to be a FROSt trace. The trace is passed into the parallelized MR convolutions to generate multiple feature maps, which are then concatenated into a single matrix. The matrix then propagates through a single convolution that doubles the number of channels and halves the pixels in both dimensions. This is achieved by selecting the stride, s, and padding parameters, p, to produce the desired shape using Equation 2.13. Simply, a single black-and-white input image ($M_i = 1$) will produce six feature maps ($M_o = 6$) at half the size of the input (meaning one MR block has $P_{MR} = 3$ MR filters and $P_c = 2$ regular filters). This MR block can be scaled to produce the same output relative to the size of the input matrix. For example, an input image with three channels ($M_i = 3$) would produce an output of 18 features maps ($M_o = 18$) at half the size (i.e., $P_{MR} = 9$ and $P_c = 2$). Since the behavior of the serialized convolution is unchanged, P_c is always two. As such, the scaled MR blocks are described by the number of MR filters only.

This MR block is scaled and repeated to produce the global architecture that is presented in Figure 3.6. For the first MR block, the process follows exactly the one described in Figure 3.5. The trace, of shape $(M = 1; N_t, N_\omega)$, is propagated into an MR block with $P = 3$ parallelized filters to produce 6 feature maps. This is input to a second MR block, having $P = 18$ filters, to generate 36 channels. A third MR block is used, with $P = 108$ filters, to produce the final output of 216 feature maps. At each MR block, the size of the feature maps is halved with regards to the input maps. After these three MR blocks, the input of a single channel has been decomposed into 216 feature maps, where each contains unique portions of the initial image. These features are then flattened into a single vector of shape $(1; 216 \times \frac{N_t}{8} \times \frac{N_\omega}{8})$. A dropout layer is used, then flattened matrix is passed to a common dense layer with N_D neurons, which is fully connected to the output layers. Each of the two output layers has 128 neurons to extract the approximations for both the electric field and the switch. The exact value of N_D, as well as the number of dense layers at Figure 3.6f,

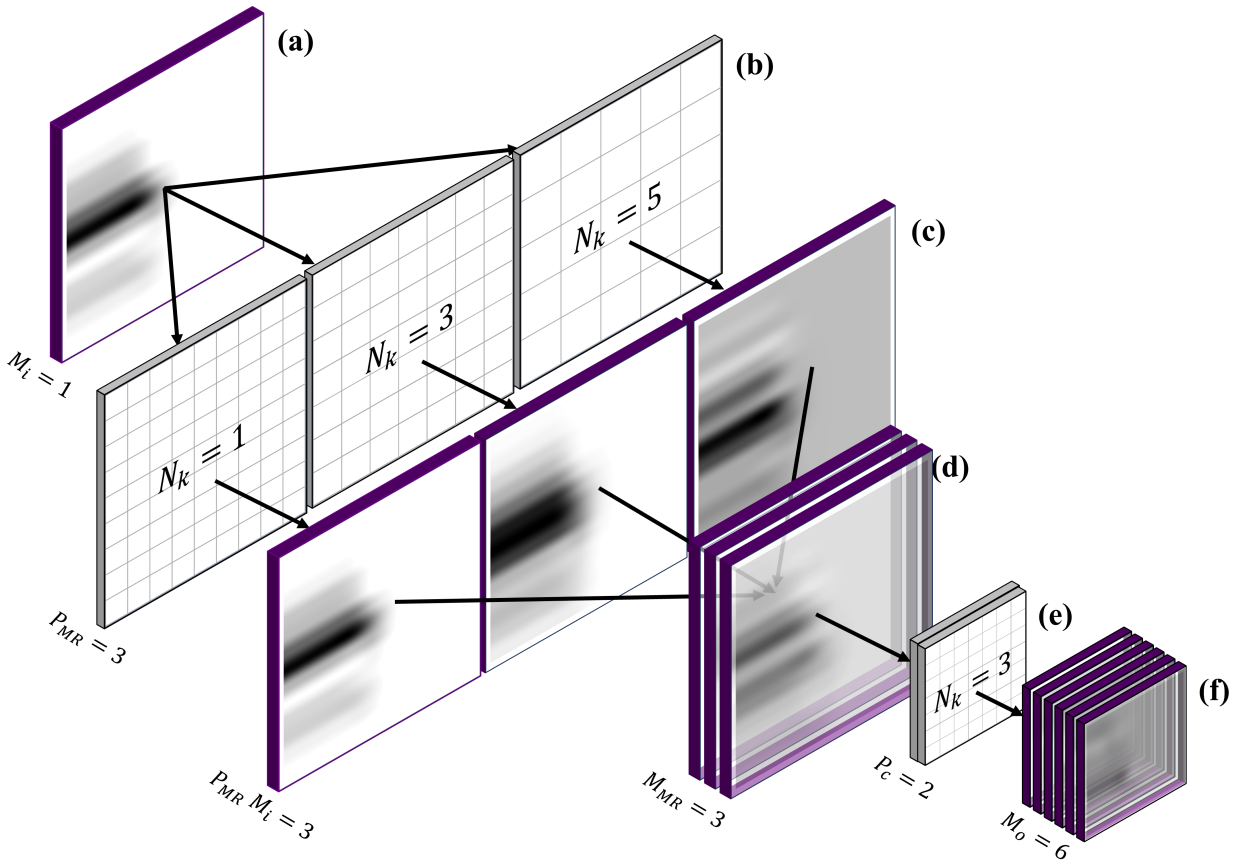

Figure 3.5 : Multi-Resolution Block Architecture.

The architecture of a single MR block is presented. Panels bordered in purple with images of FROSt traces represent inputs and outputs to layers. The number of pixels in each map is represented by the size of the panels. Convolutional layers are boarded in gray and are noted by the kernel sizes, N_k. At the input, (a), a black-and-white trace of $(N_t = 128, N_\omega = 128)$ pixels is represented as a purple panel with a single channel, $M_i = 1$. At (b), this input is passed into three parallelized MR convolutions, with $P_{MR} = 3$ filters and kernel sizes of $N_k = 1, 3,$ and 5, respectively. The output, shown at (c), is $P_{MR}\, M_i = 3$ feature maps of $(N_t = 128, N_\omega = 128)$ pixels. These are then concatenated into a single matrix of (N_t, N_ω) pixels and $M_{MR} = 3$ channels shown at (d). At (e), this matrix is passed into another convolutional layer with a kernel size $N_k = 3$ and $P_c = 2$ filters. The stride and padding were selected using Equation 2.13 to generate an output matrix with half the pixels, $(N_t/2 = 64, N_\omega/2 = 64)$. The MR output is shown at (f), where the $M_o = P_c\, M_{MR} = 6$ feature maps are half the size of the initial input. This model was generated using `tensorflow` and the code is in Figure 7.5.

are both used as parameters in the optimization process (described in Section 3.3). The high-level parameters of this architecture are optimized in the training procedure described in Section 3.3.

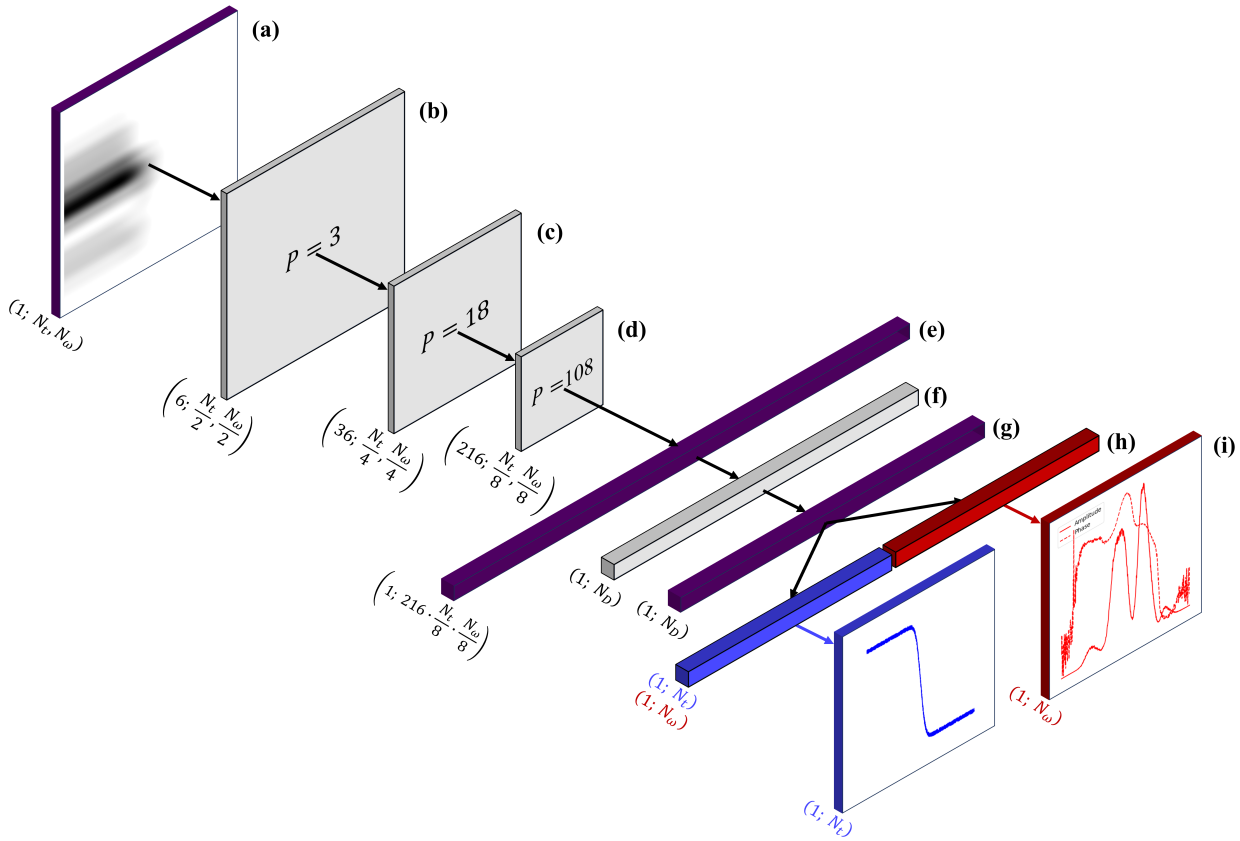

Figure 3.6 : Global Multi-Resolution Architecture.

The global architecture, which uses three serialized MR blocks, is presented. Panels bordered in purple represent inputs and outputs to layers. MR blocks are boarded in gray and are denoted by the number of MR filters, P, in each. The output shape of each MR block is denoted below each layer. At the input, (a), a black-and-white trace of $(M_{(a)} = 1; N_t = 128, N_\omega = 128)$ is represented as a purple panel. The first MR block at (b) is exactly as presented in Figure 3.5 and produces six feature maps at half the size of the input : $\big(M_{(b)} = 6; N_t/2 = 64, N_\omega/2 = 64\big)$. At (c), this is passed to a scaled MR block, with $P = 18$ filters. The output from (c) is a matrix of 36 feature maps, each a quarter of the size of the input : $\big(M_{(c)} = 36; N_t/4 = 32, N_\omega/4 = 32\big)$. This is followed by a third MR block at (d), scaled for a total of $P = 108$ filters. The output from (d) is 216 feature maps, each 1/8th of the size of the input image : $\big(M_{(d)} = 216; N_t/8 = 16, N_\omega/8 = 16\big)$. This is flattened to produce the purple vector of shape $\big(M_{(e)} = 1; 216 \times N_t/8 \times N_\omega/8\big)$ at (e). At (f), this is passed to the first dense layer with $\big(M_{(f)} = 1; N_D\big)$ neurons, which produces the purple vector of $\big(M_{(g)} = 1; N_D\big)$ at (g). This is decomposed by the final, parallel dense layers, at (h), which outputs two vectors of $\big(M_{(h)} = 1; N_t = 128\big)$ and $\big(M_{(h)} = 1; N_\omega = 128\big)$ points each. These vectors are approximations for both the switch (in blue) and the pulse (in red), as shown in the output panels (i). This model was generated using `tensorflow` and is included in Figure 7.6.

3.3 Training Procedure

The training procedure used for this work was custom developed using primarily low-level `tensorflow` and `keras` functions. It was closely inspired by Zahavy et al. (2018) but was lightly modified for this application. The DNN builds directly on the methodology established in Sections 3.1 and 3.2. The code for the training processes described here are included in Figure 7.7.

The training dataset contains $N = 2^{16}$ paired labels and images. Each label contains two vectors, where the complex electric field, $E_{P,\,L}(t)$, has $N_\omega = 128$ points and the switch, $S_L(t)$, has $N_t = 128$ points. These are propagated into FROStNET (which was described in Section 3.1) to generate the paired image (i.e., the simulated trace, $I_L(\omega, \tau)$). As such, a single data point in the set will have an electric field, a switch, and a trace.

3.3.1 Supervised Learning Procedure

The SL procedure was based on Zahavy et al. (2018) and is presented in Figure 3.7. In this training procedure, the simulation-generated labels, $E_{P,\,L}(t)$ and $S_L(t)$, are input to FROStNET to generate the trace, $I_L(\omega, \tau)$. The trace is input to the MR architecture described in Section 3.2. This architecture outputs a prediction for the pulse, $E_{P,\,AI}(t)$, and the switch, $S_{AI}(t)$, depending on the current layer weights.

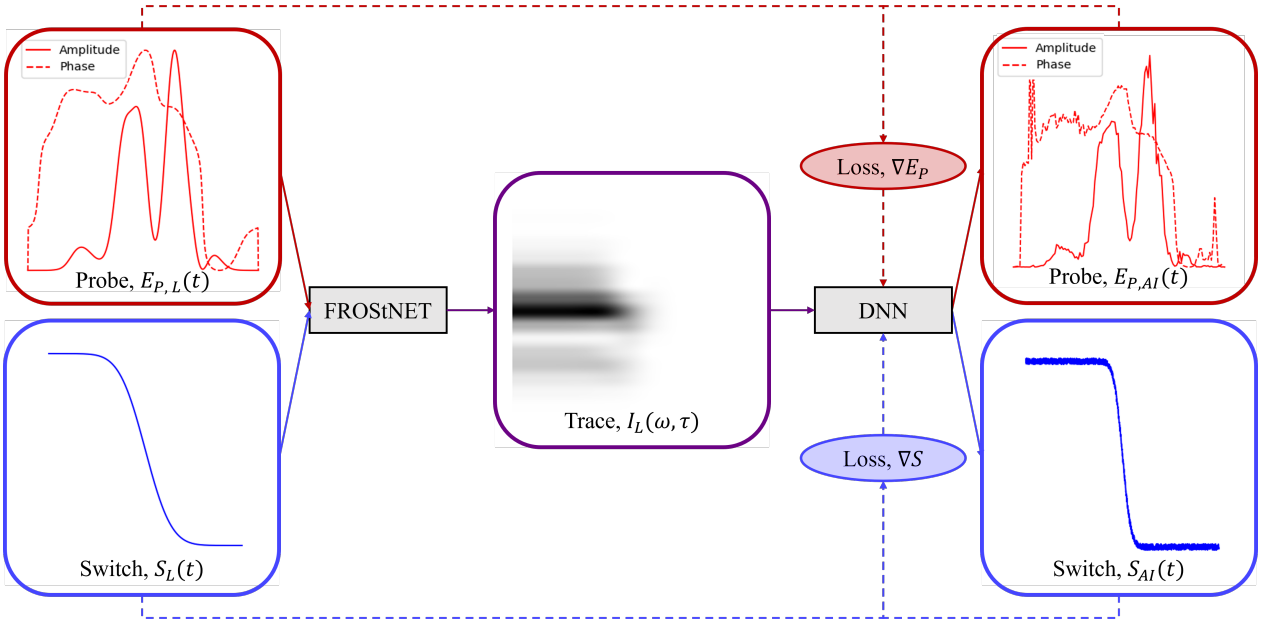

Figure 3.7 : Supervised Learning Training Procedure.

The iterative process used in the SL procedure is presented. The label probe, $E_{P,\,L}(t)$, and switch, $S_L(t)$, are propagated through FROStNET to generate the labeled trace, $I_L(\omega, \tau)$, which is then passed to the DNN. The DNN generates a prediction on both the probe and switch, $E_{P,\,AI}(t)$ and $S_{AI}(t)$. These predictions are dependent on the current state of the neural weights in each layer of the DNN. The errors between the predictions and the labels for both pulse and switch, ∇E_P and ∇S, respectively, is calculated through the loss function in Equation 3.6, which is back-propagated to the DNN to modify the neural weights of each layer. This is repeated iteratively to find the optimal weights.

During training, the loss for a single vector is evaluated using the MAE : $\nabla y = |y_L - y_{AI}|$, where y is any variable (for a complex variable y, the MAE is calculated on the magnitude). The loss function, $\mathrm{Err}_{\mathrm{SL}}$, is described in Equation 3.6 as the sum of the probe and switch losses, ∇E_P and ∇S, respectively, for all N data points. This function generates a gradient of the loss as a

function of the neural weights which is used by the optimizer, Adam, to assess the path determining in direction the weights are modulated.

$$\mathrm{Err}_{\mathrm{SL}} = \nabla E_P + \nabla S = \frac{1}{N}\sum_N |E_{P,\,L} - E_{P,\,AI}| + |S_L - S_{AI}| \tag{3.6}$$

Additionally, the effect of added noise on the training dataset is evaluated. This is achieved by adding white Gaussian noise (WGN) to the traces prior to the training process. The addition of WGN will be tested alone and in combination with other factors included in the test cases.

Test Cases. The SL procedure is evaluated using a series of test cases (presented in Table 3.1) that compare two important training parameters : the DNN output and the loss variable. The most successful test cases will also be used to evaluate the two datasets described in Section 3.1.

First, an experimental constraint (XPC) can be applied, similarly to the constraint used in ptychographic recovery (see Step (c) of the process in Section 2.2 for more details). This constraint advantageously uses the spectral intensity that is measured experimentally to identify the most valid solution. In the traditional cases, the DNN makes a prediction on the entire electric field, both the amplitude and phase. When applying the XPC, the DNN outputs a prediction on the spectral phase only, $\varphi_{AI}(\omega)$. This is combined with the known spectral amplitude from the trace to generate the predicted spectrum, $\tilde{E}_{AI}(\omega)$, using Equation 2.5. The effect of applying the XPC is evaluated by comparing test cases where the DNN output is only the spectral phase, $\varphi(\omega)$, to those where the entire spectral field, $\tilde{E}(\omega)$, is recovered.

Secondly, the loss at batch can be calculated on any variable. Because there are many variables used to describe the properties of a pulse, there are many parameters that can be matched by the training process, though not all will necessarily be equivalently successful. In these test cases, four variables are compared : **(L1)** the unbound spectral phase, $\varphi(\omega) \in (-\infty, \infty)$, **(L1-W)** the bound spectral phase, $\varphi(\omega) \in [0, 2\pi]$, **(L2)** the complex spectrum, $\tilde{E}(\omega)$, and **(L3)** the electric field, $E(t)$. For each case, the loss function will be evaluated on the variable of interest (note, the entire electric field is always determined). Simply put, each case uses a different optimization curve upon which the training procedure occurs, though all cases provide the necessary information to characterize the temporal amplitude and phase of the electric field.

These two parameters are studied through the six test cases described in Table 3.1. Each test case is identified by an ID code which identifies the loss function used (L#-) and whether the XPC is applied (-XP). Each ID describes the DNN output and the loss variable used in Equation 3.6.

Table 3.1 : Training Test Cases.

Parameters used in each test case, as well as their identifiers. The DNN Output describes the output parameter from the DNN while the Loss Variable describes the variable used to calculate the error for back-propagation. The IDs are used to identify the test case results.

Test Case ID	DNN Output	Loss Variable
L1-XP	$\varphi(\omega)$	$\varphi(\omega) \in (-\infty, \infty)$
L1-XP-W	$\varphi(\omega)$	$\varphi(\omega) \in [0, 2\pi]$
L2	$\tilde{E}(\omega)$	$\tilde{E}(\omega)$
L2-XP	$\varphi(\omega)$	$\tilde{E}(\omega)$
L3	$\tilde{E}(\omega)$	$E(t)$
L3-XP	$\varphi(\omega)$	$E(t)$

3.3.2 Unsupervised Learning Procedure

The SL process is modified to produce a more complex UL training procedure, as presented in Figure 3.8. As before, the simulated trace, $I_L(\omega, \tau)$, is input to the DNN to generate predictions, $E_{P,\,AI}(t)$ and $S_{AI}(t)$, which are now propagated through FROStNET to generate the predicted trace, $I_{AI}(\omega, \tau)$.

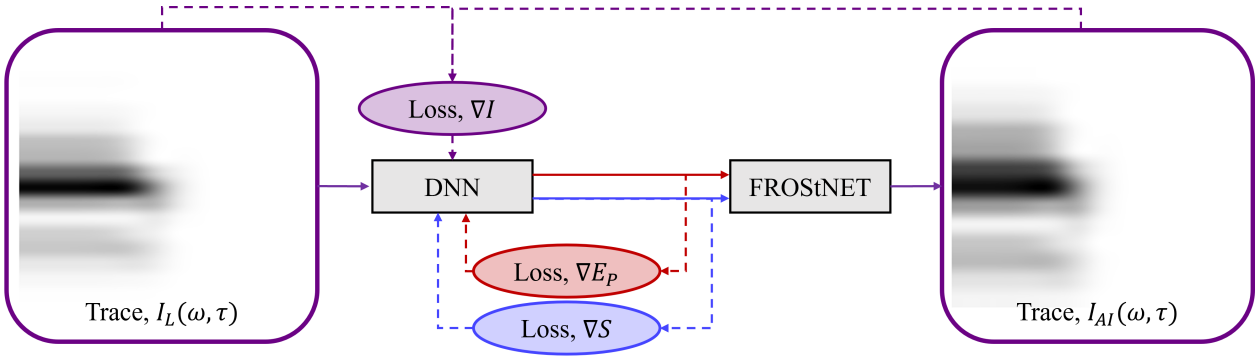

Figure 3.8 : Unsupervised Learning Training Procedure.

Modifications to the SL training are made to generate the UL process as presented. The initial generation of the simulated traces, $I_L(\omega, \tau)$, proceeds as it would for SL (see Figure 3.7 for reference). The traces are propagated into the DNN as before to produce the predictions on the pulse and switch. The SL losses, ∇E_P and ∇S, are calculated as before using Equation 3.6. In the UL process, the DNN's predictions are input to FROStNET to generate the predicted trace, $I_{AI}(\omega, \tau)$. A new loss value, ∇I, is calculated from Equation 3.8 and it is used to modulate the global error, as in Equation 3.7. This global error is back-propagated to the DNN throughout the training process.

Unlike SL training, in the UL procedure, the predicted trace is considered in the loss function, $\mathrm{Err}_{\mathrm{UL}}$, defined by Equation 3.7. Here, the supervised error from Equation 3.6 is modulated by the intensity error, ∇I, whose contribution is governed by a hybridization parameter, Λ.

$$\mathrm{Err}_{\mathrm{UL}} = \nabla E_P + \nabla S + \Lambda \nabla I \tag{3.7}$$

The intensity error, ∇I, is not the MAE, as it is for the switch and pulse errors, ∇S and ∇E_P. Instead, the intensity error is related to the fundamental mathematical link between the electric field, switch, and FROSt trace. This relation is described by Equation 3.8, which utilizes the weights of FROStNET to generate an estimate loss on the temporal intensity.

$$\nabla I = |\nabla E_P \, \nabla S|^2 \tag{3.8}$$

The inclusion of this term in the back-propagated error, $\mathrm{Err}_{\mathrm{UL}}$, is key to the UL procedure. This quantity relates directly to the underlying mathematical relation between the labels and data in the SL procedure. Importantly, the UL procedure requires a stabilized SL process, but may produce a DNN that is more robust to the sim-to-real gap and experimental noise.

Optimization. After the global training procedure is stabilized, a selection of machine learning hyperparameters, presented in Table 3.2, are optimized using an `Optuna` objective. Batch sizes are varied from 8 to 64 in powers of 2. The number of neurons, as well as the number of layers, of the dense NN are varied as shown. The rate on the dropout layer after flattening and Adam's learning rate are also varied. The epochs are set to 50 with early stopping to opt out of further training iterations if signs of over-fitting occur (Goodfellow et al., 2016). The objective was trialed for $N = 50$ to determine the combined parameters that result in the lowest experimental error. For the test cases described in Section 4, these values are set to those shown in Default. The code used to execute this optimization is shown in Figure 7.9.

Table 3.2 : Hyperparameters for `Optuna` Study.

The hyperparameters optimized with the `Optuna` objective are presented, as well as the range over which they were varied. The default values used in the test cases is also included. The objective was allowed to run for $N = 50$ trials with the goal of minimizing the electric field validation error on the experimental dataset.

Parameter	Min	Max	Default
Batches	**8**	**64**	**32**
Dense Neurons, N_D	**256**	**2048**	**512**
Dense Layers	**1**	**5**	**1**
Dropout Rate	**0%**	**50%**	**0%**
Learning Rate	$1e-5$	$1e-1$	$1e-3$

3.4 Experimental Validation

The test cases described are compared using the MAE of the electric field and the switch as the metrics, calculated using Equation 3.6. For simulated validation sets, this is easily achieved using the labels generated. However, these algorithms must be evaluated on experimental traces to truly

examine their real world behaviors. To achieve this, the trained DNNs are evaluated in comparison with the standard recovery method described in Section 2.2. For this work, the ptychographic implementation is identical to Leblanc (2016) and has been integrated into a user interface (UI) known as PtyChoPy (referring to the two languages used, C and Python). Python handles the UI components, while the iterative program is written in C and wrapped using `Cython` functions to reduce computation time.

As stated in Section 2.2, ptychographic recovery requires preliminary pre-processing steps which we integrated directly into PtyChoPy. First, the traces are sliced temporally. Background removal is performed using `SciPy`, followed by removal of outliers ($I > 4\,\overline{I}$, where $\overline{I}$ is the average intensity in both ω and τ). Spectral filtering is performed using a hyper-Gaussian filter of order 80. Similarly, temporal smoothing is done using a Gaussian filter of order 6. Finally, cubic interpolation is done in both time and frequency domains to ensure the resolution meets Nyquist's criterion. The code that implements these steps is included in Figure 7.10.

The recovery algorithm implemented after pre-processing is identical to that described in Leblanc (2016); Bourassin-Bouchet et al. (2013); Fienup (1982). The recovery algorithm implements a combination of two variants of ptychography, error reduction (ER) and hybrid input-output (HIO). These two algorithms differ only in how they approach the approximation achieved after each iteration. The ER algorithm is exactly as described in Section 2.2, while HIO uses a combination of the proceeding step and the approximation it generated to calculate the successive steps. This alters the optimization path generated by the algorithm, allowing for more effective identification of a minimum. The algorithm used in this work applies first the HIO algorithm for some iterations then uses the ER algorithm until convergence is reached (see Chapter 12 of Leblanc (2016) for more details).

An example case of PtyChoPy is presented in Figure 3.9. The original experimental trace is presented in Figure 3.9a, with the processed version in 3.9b. The processed trace is passed through the ptychographic recovery method to retrieve the electric field and switch profiles presented in Figure 3.9d and 3.9f. These are propagated through FROStNET to generate the retrieved trace shown in Figure 3.9c. This reconstruction was highly successful with an error of only 2.58% when compared to the processed trace in 3.9b. Additionally, the spectrum of the electric field, shown in Figure 3.9e, is acquired computationally using the FT. These profiles are used as labels when evaluating the experimental traces but, as metrics, have no impact on the training process.

An experimental dataset of $N = 77$ traces are collected upon which the model is evaluated. These traces were generated from a variety of experimental conditions. The collected experimental traces have a minimum, maximum, and mean central angular frequencies of 0.18×10^{13} , 2.51×10^{13} , and $1.15 \times 10^{13}\,\mathrm{rad\,s^{-1}}$, respectively, and a mean SNR of 28.93 dB. These are processed using PtyChoPy to generate the ptychographic recovery which is used treated as the true label to evaluate the model. The error on the ptychographic recovery is calculated through Equation 2.10 and is an average of

3.55% for the described dataset. This error will be used as a threshold to determine at what point the AI method is comparable to ptychography.

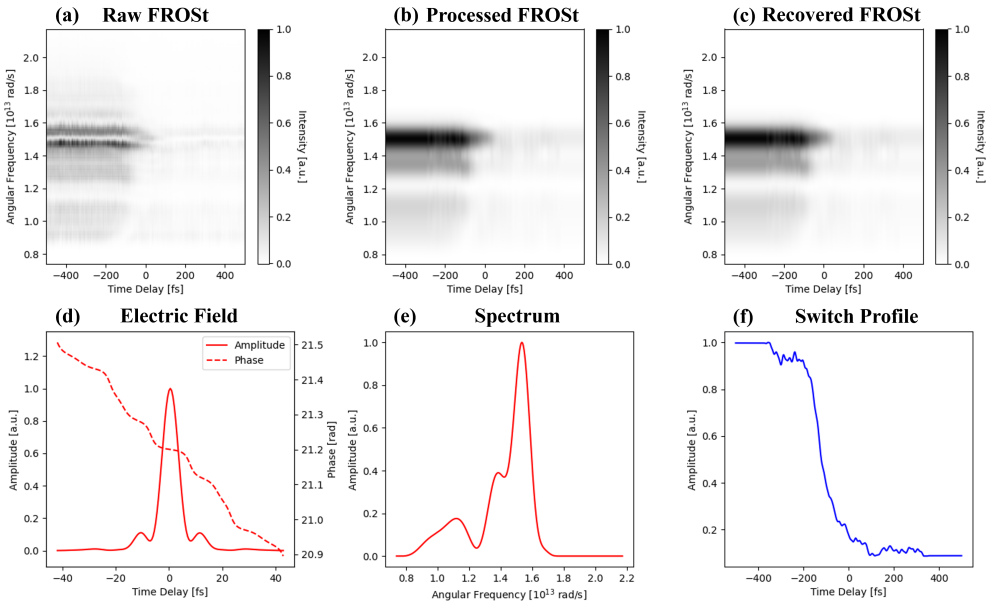

Figure 3.9 : PtyChoPy's Inputs and Outputs.

The process performed by PtyChoPy is visualized. In (a), the original experimental FROSt trace is shown. The FROSt trace was acquired experimentally from a pulse with a central angular frequency, ω_0, of $1.30 \times 10^{13}\,\mathrm{rad\,s^{-1}}$ and an SNR of $31.21\,\mathrm{dB}$, after having been propagated through $6\,\mathrm{mm}$ of fused silica. The same trace is shown after pre-processing in (b). This processed trace is used as the input for the ptychographic recovery. The recovery process outputs the solutions in (c) through (f). In (c), the trace generated from the recovered electric field and switch is shown. The electric field (amplitude in continuous and phase in dashed) is shown in (d) and its spectrum in (e). The switch amplitude is shown in (f). This reconstruction has an error of 2.58% calculated from Equation 2.10, which is considered very low for ptychography.

4 RESULTS

We now present the results generated from executing the training procedure of Section 3.3. In each Section, we will comment briefly on the results presented, which will be followed by a more detailed analysis in Section 5. Before proceeding with a more detailed analysis in Sections 4.2 and 4.3, two successful predictions are presented. This is followed by Section 4.2 where the test cases and datasets are analyzed, which we revisit in Section 5.1. Section 4.3 presents results from adding WGN (in Section 4.3.1) and the UL procedure (in Section 4.3.2). Finally, the optimized DNN, including hyperparameters, will be detailed in Section 4.3.3 (with further discussion in Section 5.2).

The evaluation process was achieved by generating predictions on a dataset of $N = 2^{14} = 16384$ data points to make up the validation set, generated as described in Section 3.1. The NNs have never been exposed to these data points before. Metrics are taken on these predictions using Equation 2.10 for the traces and Equation 3.6 for the electric field and switch predictions. This process was then repeated using the experimental validation set collected in Section 3.4.

4.1 Example Predictions

Before proceeding to a statistical analysis of the predictions, we demonstrate the general success of the methodology by showing predictions generated on data points from the simulated and experimental validation sets in Figures 4.1 and 4.2, respectively. They are shown here with the predictions generated from the final DNN.

Figure 4.1 shows the prediction generated from a simulated data point in the validation set. The labeled trace is shown in Figure 4.1a, while the reconstructed trace generated from the predicted vectors is in Figure 4.1b. The temporal envelope and phase for the labeled and predicted electric field is presented in Figure 4.1c and d, respectively. Figure 4.1e shows the spectra of these electric fields. The labeled and recovered switch profiles are shown in Figure 4.1f. This prediction had an error of 1.45%, 11.03%, and 9.99% on the switch, electric field, and trace, respectively. The predicted temporal envelope is not identical to the label, though it shares many of the general features (the spectral beating seen in both label and prediction is a result of the third-order phase). We note that there is remarkable agreement between the temporal phases, given that the offset seen in Figure 4.1d is a multiple of 2π. Despite being generated from two unique electric fields, the labeled and predicted traces are visually very similar. This reinforces the predicament described by the 2D PRP ; two different electric fields can generate very similar FROSt traces.

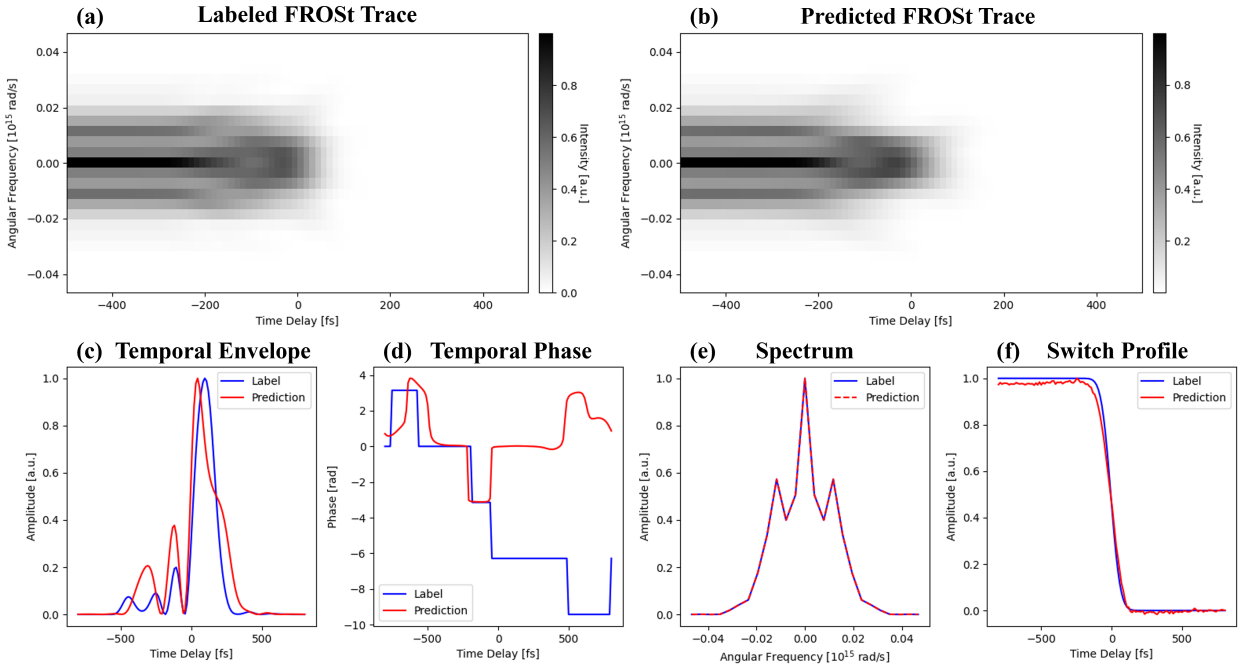

Figure 4.1 : Prediction Generated on Simulated Data.

The labeled trace is shown in (a), as it was input to the NN. The reconstructed trace, generated from the predictions, is shown in (b). The labeled (in blue) and predicted (in red) vectors for the temporal pulse envelope and phase are shown in (c) and (d), respectively. The spectrum of the labeled and predicted electric field is presented in (e). The temporal amplitude of the labeled and predicted switch is shown in (f). For this case, the reconstruction errors on the switch, pulse and trace are 1.45%, 11.03%, and 9.99%, respectively.

Similarly, a prediction generated from an experimental trace is compared to its ptychographic label in Figure 4.2. It is important to note that the predictions are generated from the original, unprocessed experimental trace, while the ptychographic recovery does require the pre-processing steps described in Section 3.4 to acquire an equivalent prediction. This prediction has an error of 7.88%, 9.74%, and 46.45% for the switch, electric field, and trace, respectively. Despite being provided with the unprocessed trace, it can be seen that the DNN recovered almost exactly the temporal envelope. Importantly, the temporal phase is constantly offset from the label, indicating that the prediction is related to the label through one of the time-space ambiguities discussed in Section 3.1. This demonstrates that a DNN trained using the methodology described in Section 3 is successfully able to reconstruct electric fields from both simulated and experimental FROSt traces. Five other cases of experimental traces are shown in Appendix B. These exhibit similar patterns to the one shown here, where the electric field of the pulses have a strong agreement though the switch profile does not.

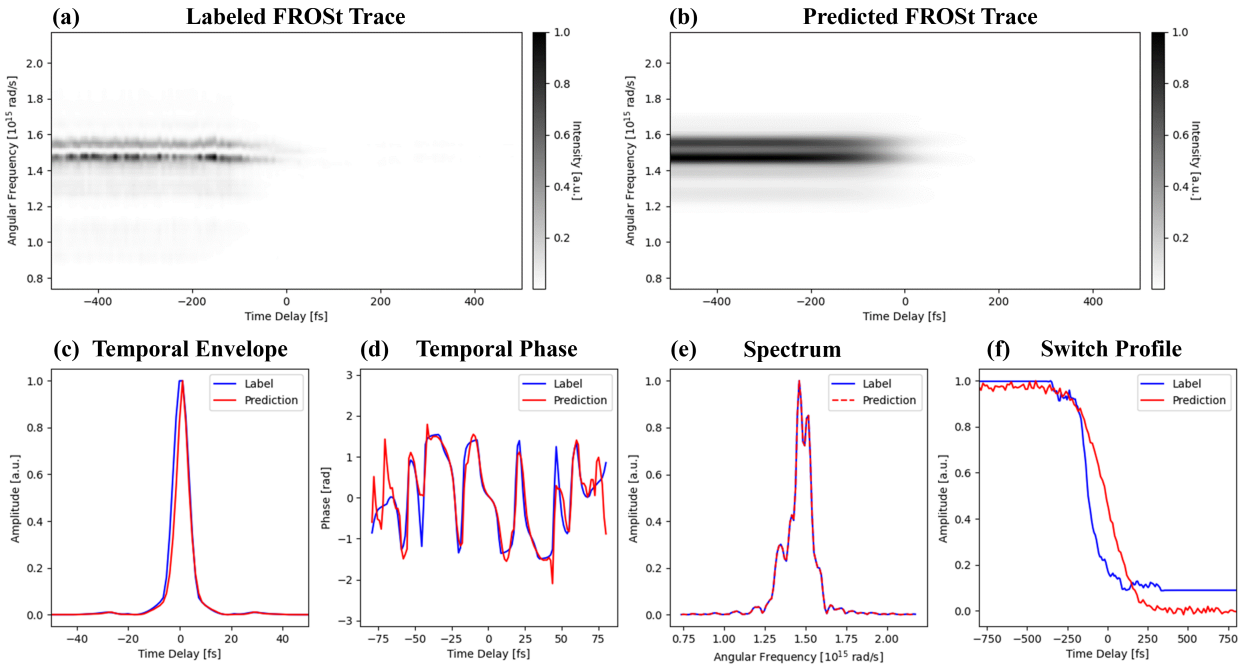

Figure 4.2 : Prediction Generated on Experimental Trace.

The raw experimental FROSt trace is shown in (a). This trace did not undergo any filtering or pre-processing prior to generating a prediction. The reconstructed trace, generated from the DNN predictions, is shown in (b). The labeled (in blue) and predicted (in red) vectors for the pulse envelope and phase are shown in (c) and (d), respectively. The spectrum of the labeled and predicted electric field is presented in (e). The temporal amplitude of the labeled and predicted switch is shown in (f). For this case, the reconstruction errors on the switch, pulse and trace are 7.88%, 9.74%, and 46.45%, respectively. Note, that the experimental labels are recovered from the ptychographic process described in Section 3.4.

4.2 Test Cases

Not all test cases generated successful predictions like those shown in Section 4.1. We now present the results from each of the test cases described in Table 3.1 on one of three datasets. The three datasets are the single-switch (SS) and multi-switch (MS) datasets of Section 3.1, while the third is an augmented (AUG) dataset generated from predictions. The predictions generated from each test case are evaluated in Figure 4.3 where the switch, electric field, and trace errors are shown for both simulated and experimental validation sets.

First, we consider the SS dataset, where 65536 simulated electric fields are paired with one switch to produce 65536 unique traces. Each of the six test cases were trained on this dataset and their predictions were evaluated to generated the errors presented in row 1 of Figure 4.3. Four of the test cases were repeated using the same 65536 electric fields paired instead with one of 300 switch profiles to produce 65536 unique traces. The predictions from these test cases are compared in row 2 of Figure 4.3.

Finally, to reduce the reliance on the simulated dataset, the predictions generated from the SS test cases are used to augment the training dataset, for a total size of $N = 2 \times 2^{16} = 131072$ data points. The AUG dataset is made up of the original 65536 simulated electric fields, as well as the predictions generated on these inputs (such as the one presented in Figure 4.1). These electric fields are paired with predicted switches to produce 131072 unique traces. The predicted electric fields and switches provide more variations than the original simulated data, which is desirable to overcome the sim-to-real gap (see Section 1.4 for more). The best two test cases were then trained on the augmented dataset and are compared as before in row 3 of Figure 4.3. In each case, the reconstructed experimental traces are compared to the mean ptychographic recovery error acquired from PtyChoPy for the same dataset.

Test Case Training Times. Each test case required a different set of calculations to generate the loss functions. In Table 4.1, the time taken to train a single epoch of 65536 data points in each of the test cases is shown. Training was performed on a standard PC using an NVIDIA GeForce RTX 3090 graphical processing unit (GPU) and homemade `tensorflow-gpu` functions in Python 3.10 (script included in Figure 7.7). This includes the time required to generate the FROSt traces from the label electric fields and switches using FROStNET.

Table 4.1 : Training Time Per Epoch.

The training time for each test case is presented for a single epoch of 65536 points.

Test Case	Epoch Time
L1-XP	9.45 min
L1-XP-W	9.55 min
L2	3.89 min
L2-XP	9.76 min
L3	7.71 min
L3-XP	9.80 min

It is interesting to note that the quickest test case (L2, which requires just under 4 minutes per epoch) requires the fewest computations (as no FTs are necessary when calculating the loss function) but is also by far the least accurate. By comparing the time for L2 and L3, we can see over 3 min of computation is added by the FTs required for the loss function. This is taken into consideration when selecting the best test cases to study further.

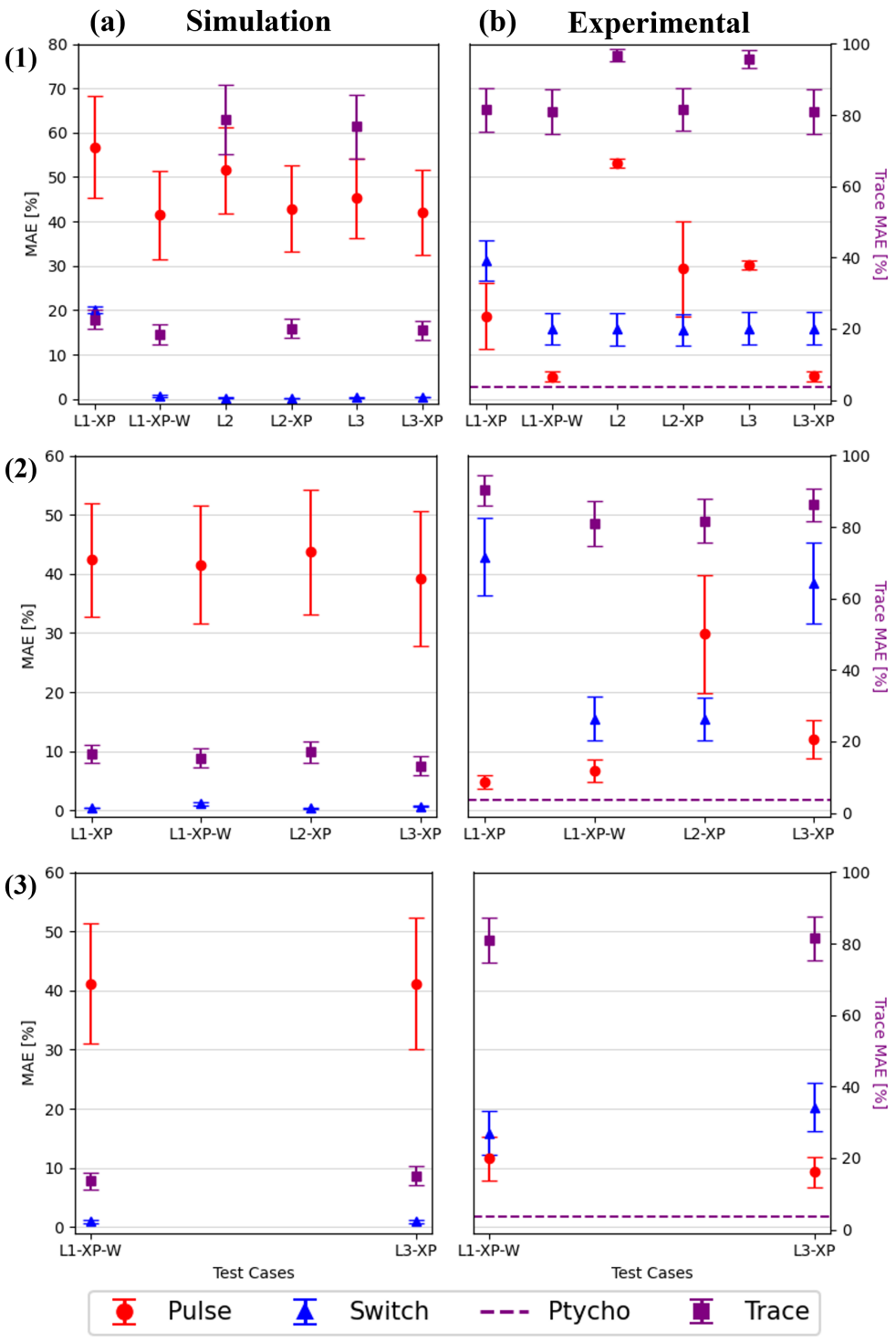

Figure 4.3 : Evaluation of Test Cases on Validation Sets.

Cases were tested on Simulated (a) and Experimental (b) Validation Sets. The validation error (calculated from Equation 3.6) for the pulse (in red, circles), the switch (in blue, triangles), and the trace (in purple, squares) is shown for each of the test cases. Error bars represent the MAE across the validation sets for each metric. On the experimental errors, (b), the dashed purple line shows the average trace error for the ptychographic recovery on the same dataset. These errors are shown for test cases trained on three datasets : row (1) SS, row (2) MS, and (3) AUG.

4.3 Parameter Optimization

Two important variables were swept in order to measure the training as a function of both parameters. First, we considered the addition of WGN to the training dataset in an attempt to improve experimental predictions. Next, the UL hybridization parameter, Λ, was considered. These variables will be combined to optimize the DNN in Section 4.3.3.

4.3.1 White Gaussian Noise

Test cases L1-XP-W and L3-XP are repeated using the MS dataset. This time, WGN was added to the traces before training and the validation process was repeated on the noise-trained DNNs. The SNR was swept from 0 to 30 dB, using 10 steps, and the resulting errors are presented in Figure 4.4.

For the test case L1-XP-W in row 1 of Figure 4.4, the trace error was constant with the added noise for both simulated and experimental validation sets. For simulated validation sets in Figure 4.4a, the error on the electric field and switch was also constant. For experimental sets in Figure 4.4b, the reconstruction on the switch increased slightly for SNRs between 15 and 25 dB before decreasing again. Meanwhile, the recovery for the electric field was at a minimum of 4.73% for 3.5 dB and a maximum of 10.46% for 20 dB. Similarly to the switch, the error seemed to rise in the region from 15 to 25 dB before decreasing. The error did not return to the minimum seen with no training noise but remained comparable.

For test case L3-XP in row 2 of Figure 4.4, very little fluctuations were noted for the trace error in either simulated and experimental validation sets. The simulation error on the switch was also constantly low. For experimental validation set, the electric field error reached a minimum of 13.39% at 7.04 dB, which was lower than the 12.08% error at a 0 dB. The switch error increased slightly between 25 and 30 dB but remained relatively constant. In both simulated and experimental validations, the training cases with 30 dB showed very poor recovery of the electric field.

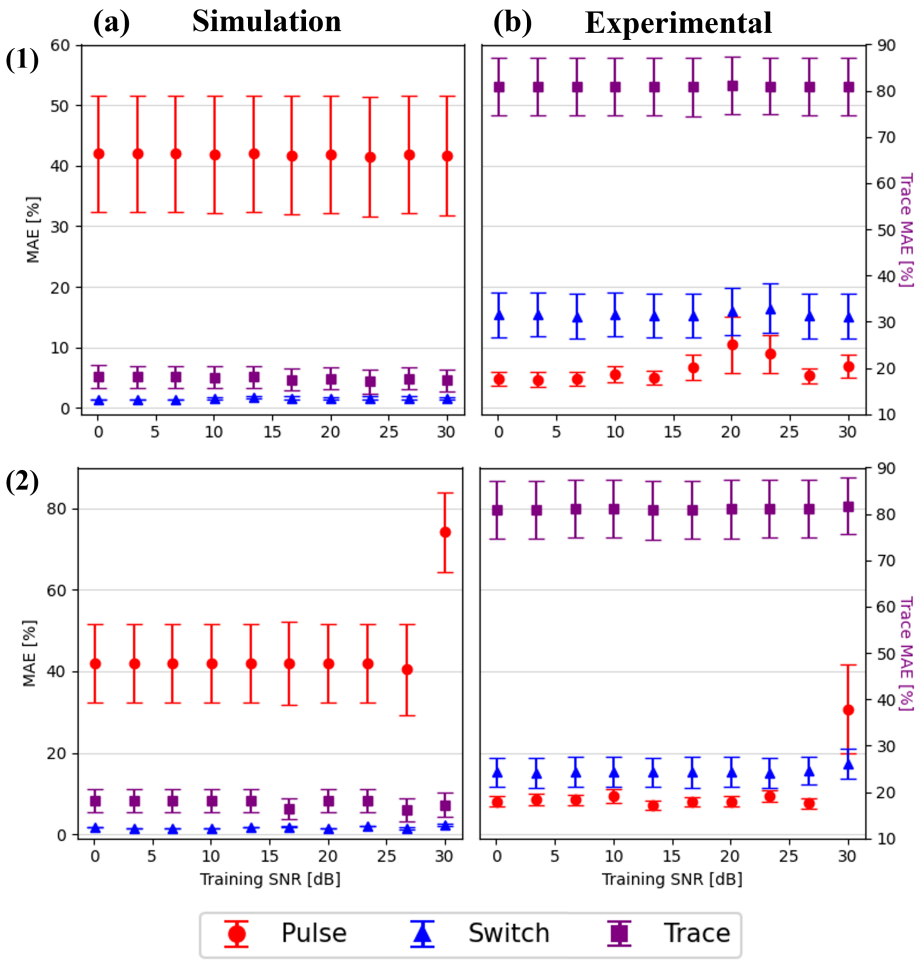

Figure 4.4 : Evaluation of WGN on Training Validation Sets.

Evaluation of WGN on L1-XP-W, row (1), and L3-XP, row (2). Training Procedures for Simulated (a) and Experimental (b) Validation Sets. The validation error (calculated from Equation 3.6) for the pulse (in red, circles), the switch (in blue, triangles), and the trace (in purple, squares) is shown for each SNR added to the training dataset. Error bars represent the MAE across the validation sets for each metric.

4.3.2 Unsupervised Learning Hybridization

Only test case L1-W-XP was repeated using the modified UL loss function from Equation 3.7. The original MS dataset with 65536 electric fields and 300 switch profiles to produce 65536 unique traces for the training dataset. The hybridization parameter, Λ, was swept from 0.1 to 0.9 with 10 steps. This parameter indicates the percent of UL loss that modulates the back-propagation (where $\Lambda = 0$ is identical to the SL loss in Equation 3.6).

From Figure 4.5a, it was seen that the recovery of the electric field and the switch from simulated data was largely unaffected by the inclusion of the UL procedure. Some variation was seen in the trace error, which increased insignificantly with the hybridization parameter. For experimental data,

in Figure 4.5b, it can be seen that the UL procedure had very little impact on the trace or switch recovery error, which remained at 80% and 15%, respectively. However, the recovery of the electric field was significantly influenced by the UL procedure with a minimum error of 4.72% for a value of $\Lambda = 0.45$ and a maximum error of 12.42% reached at $\Lambda = 0.18$ (which seemed to be an outlier). Importantly, the UL seemed to outperform the SL when recovering the electric field (a low of 4.72% for the UL compared to 5.02% for the equivalent SL).

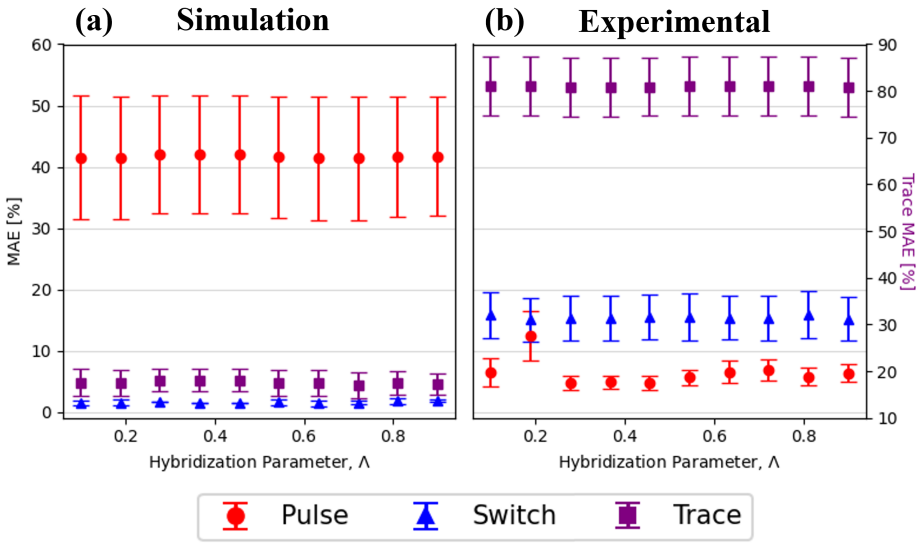

Figure 4.5 : Evaluation of UL on Training Validation Sets.

Evaluation of UL Test Case L1-XP-W for Simulated (a) and Experimental (b) Validation Sets. The validation error (calculated from Equation 3.6) for the pulse (in red, circles), the switch (in blue, triangles), and the trace (in purple, squares) is shown for the hybridization parameter, Λ. Error bars represent the MAE across the validation sets for each metric.

4.3.3 Hyperparameters

Test case L1-XP-W was determined to be the most successful training procedure. This was used as the basis for the `Optuna` study developed. The WGN rate and the hybridization parameter were added to the hyperparameters planned for the `Optuna` study in Table 3.2. The hybridization and noise rate parameters were allowed to vary around the minima regions previously noted to find a global minimum. After 50 trials, the following parameters were determined to produce the best trained DNN.

We compared the optimized DNN to the other variations of test case L1-XP-W, as shown in Figure 4.6. The validation error for variants of L1-XP-W training procedures were presented : SS, MS, AUG, WGN of 3.5 dB, and UL with $\Lambda = 0.45$. Additionally, a case where WGN of 3.5 dB and a hybridization parameter of 0.45 were applied simultaneously was trained on the MS and AUG datasets (designated C-MS, and C-AUG, respectively). The optimized case, determined from the

`Optuna` study, is named DFROStNET and has an experimental error of 15.11%, 4.39%, and 80.92% for the switch, electric field, and trace, respectively.

Table 4.2 : Optimized Parameters from `Optuna` Study.

The ranges allowed for each parameter in the `Optuna` study (script included in Figure 7.9). This was performed only on the most successful test case of Table 3.1. DFROStNET shows the training parameters used for the optimized training procedure.

Parameter	Min	Max	DFROStNET
Dropout Rate	**10%**	**50%**	**21%**
Learning Rate	$1e-5$	$1e-1$	$1.78e-3$
Batches	**16**	**128**	**32**
Dense Neurons, N_D	**256**	**2048**	**575**
Dense Layers	**1**	**5**	**5**
WGN	5 dB	15 dB	7 dB
Hybridization	**0.1**	**0.5**	**0.17**

While it can be seen that training with either the WGN or UL procedures had a slight improvement on the SS predictions, there was a notably lower error on MS and AUG predictions. Overall, there seemed to be very little improvement in the switch predictions when trained with the more variable switch datasets. The addition of WGN and the UL procedure clearly improved the pulse predictions on the more complex datasets. However, there seemed to be a general plateau at around 5% error on the electric field predictions. The optimization of the hyperparameters was able to improve the error rate from 4.73% (C-MS) to 4.39% (DFROStNET).

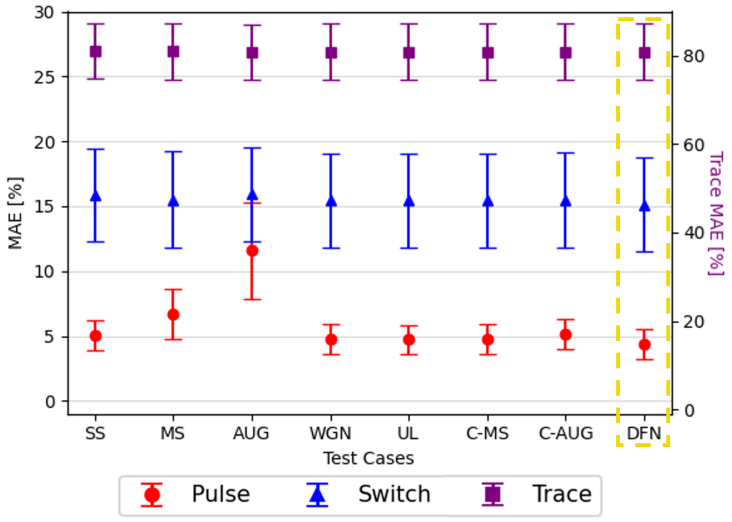

Figure 4.6 : Comparing Variants of L1-XP-W on Experimental Validation Set.

The validation error (calculated from Equation 3.6) for the pulse (in red, circles), the switch (in blue, triangles), and the trace (in purple, squares) is shown for the following cases : SS, MS, AUG, WGN of 3.5 dB, UL with $\Lambda = 0.45$, combined WGN and UL with MS (C-MS) and with AUG datasets (C-AUG). The optimized training, DFROStNET (DFN), is highlighted by the dashed golden box. Error bars represent the MAE across the validation set for each metric.

5 DISCUSSION

From the results presented in Section 4, we can state that the methodology presented in this work is effective in training a DNN to extract the temporal amplitude and phase of an electric field from its FROSt trace. In this Section, we elaborate on the results presented in each part of Section 4. The behaviors of various test cases and datasets are discussed in Section 5.1. This is followed by a discussion on the parameters that improved the experimental predictions in Section 5.2. Section 5.3 compares the proposed method to the traditional ptychographic recovery method.

5.1 Evaluating DFROStNET

In the following section, we discuss the various test cases and the effect of different switch training sets. Figure 4.3a is the focus of Section 5.1.1, while Section 5.1.2 discusses the impact of training on the MS and AUG datasets (covering Figures 4.3b and 4.3c).

5.1.1 Test Cases

The results from the test cases trained on the SS dataset are discussed first. We focus on comparing the effects of three individual parameters, as they were presented in Section 3.3. First, we consider the bounds of the spectral phase function used in L1-XP. This is followed by a discussion on the XPC. Finally, each of the three loss variables is discussed — **L1** : spectral phase, $\varphi(\omega)$, **L2** : complex spectrum, $\tilde{E}(\omega)$, and **L3** : complex electric field, $E(t)$.

Bounds of L1 : $\varphi(\omega)$. We compare the results from L1-XP and L1-XP-W, where there is a notable difference in behavior. The former used the unwrapped phase ($\varphi(\omega) \in (-\infty, \infty)$), while the latter used the wrapped spectral phase ($\varphi(\omega) \in [0, 2\pi]$). The DNN trained using the wrapped spectral phase (test case L1-XP-W) shows much better recovery errors on the switch and pulse profiles for both simulated and experimental validation sets. This is due to the ambiguities discussed in Section 3.1. It is known that, physically, a phase offset of any multiple of π is meaningless. However, this is an ambiguity to which DNNs are sensitive. If two phases are identical but offset by a large multiple of π, the loss function using the unwrapped phase will report a high error associated with the offset. Meanwhile, the same two phases, but wrapped, show a very small error when compared. Interestingly, the test case L1-XP also had a very high error on switch reconstruction compared to other test cases for both simulated and experimental datasets. Meanwhile, the switch recovery from L1-XP-W was comparable to all other test cases. As a result, the trace error for L1-XP was generally slightly larger than that of L1-XP-W. Clearly, using the wrapped spectral phase in the loss

function is important to improve accuracy of the training procedure. This demonstrates a fallibility of DNNs when it comes to simple ambiguities such as a relative offset.

Experimental Constraint. The effect of the XPC is examined by comparing test cases L2 to L2-XP and L3 to L3-XP. In Figure 4.3a, it can clearly be seen that test cases where the XPC is applied show much lower error on the reconstructed traces (around 80% compared to around 20%, with and without XPC, respectively). This pattern is also noted in the XP dataset, where the error on reconstructed traces remains near 100% for test cases L2 and L3 but drops to 80% with the XPC. It can be seen that the application of the XPC has a significant impact on the quality of training. This is sensible, as the XPC is designed to improve the convergence during optimization, and indicates that this was successful. Without the application of the XPC, the proposed method is generally ineffective.

Loss Variable. Now, the loss variables are compared. It can be noted that certain loss variables produce more effective optimization maps for training. When examining experimental errors in Figure 4.3b, it is clear that L1-XP-W and L3-XP both outperform L2. L1-XP-W and L3-XP both show remarkable recovery of the electric fields, as well as equivalent switch and trace errors. Moreover, we note the increased unreliability of the solutions generated by test case L2-XP. The experimental error on the pulse shown in Figure 4.3b is much higher for this test case than for any other. We hypothesize that this is because the spectral phase has a limited bandwidth of relevant information (as the phase values are irrelevant where no amplitude exists). Since the pulses in the training dataset shared a constant, relatively narrow bandwidth, much of the label was made up of redundant information. Meanwhile, the electric fields were all broader than the FTL and had very few indices that were zero, meaning more phase information was provided by each data point. As a result, the increased error on the complex spectrum may not be replicated if trained on a dataset with a different underlying methodology. However, for this training dataset, it can be determined that, of the three loss variables used, calculating loss with L1, the spectral phase, and L3, the electric field, showed the best results. Since it was seen that L1 takes less time than L3 (Table 4.1), L1 is favoured as the most resource effective loss variable.

5.1.2 Training Datasets

With an improved understanding of each of the test cases, we now examine the effect of diversifying the switch profiles used in the training dataset. Each of the experimentally constrained test cases were trained again on more diverse switches (MS) and a larger prediction-generated dataset (AUG), which is compared to the simplest dataset (SS). The effect of training on each dataset is discussed below.

Single-Switch Dataset. For the test cases trained on a single switch dataset in row 1 of Figure 4.3, the reconstruction of the switch is consistently of relatively poor quality. A plateau is reached at around 15% error for the experimental validation set. This is due to the limited number of switches used in the dataset. Creating a more diverse, realistic dataset of switch profiles would improve the reconstruction, though that is not the focus of this work. Moreover, it is interesting to note that the pulse reconstruction does not suffer because of the poor switch recovery. While the switch error plateaus, the pulse recovery remains very small for test cases L1-XP-W and L3-XP. However, this does have a significant effect on the trace reconstruction error. Even a very well reconstructed pulse will have a high trace error if paired with the incorrect switch. This is an important limitation to the process described in this work, as the trace reconstruction error is a significant validation to ensure the generated solution is accurate.

Multi-Switch Dataset. When training on a dataset made of up 300 unique switch profiles, the training behaviors were altered. From row 2 of Table 4.3, it can be seen that the simulation errors were relatively unaffected by the additional switches. The experimental validation shows that L1-XP-W and L2-XP both performed equivalently when trained on either single or multiple switches. However, L1-XP and L3-XP seem to be very sensitive to varying switch profiles. In the case of L1-XP, the recovery went from 30% with SS to almost 45% using MS. Similarly, for L3-XP, the error goes from 15% to 40%. Notably, case L1-XP showed signs of over-fitting despite the early stop implemented, as the electric field predictions were identical for all inputs. Meanwhile, the electric field recovery for L1-XP improves significantly while the opposite is true for L3-XP. Additionally, both L1-XP and L3-XP showed slightly higher errors on the reconstructed trace. Interestingly, L2-XP seems to have very little modulation related to the switch.

In short, the added complexity of multiple switch profiles had an impact on the final predictions, though not significantly or positively. The parameters used for the switches (i.e., varying the FWHM) is not adequate to exemplify the variety of behaviors seen experimentally. Moreover, using 300 switches is not nearly large enough to generate a diverse dataset to train on. As a result, the model exhibited signs of a phenomenon known as mode collapse (Yao et al., 2023; Zhang, 2021; Shumailov et al., 2024), as demonstrated in Figure 5.1. As was mentioned, modeling the material dynamics that generates the population inversion is studied elsewhere (Satou et al., 2011; Amstrup et al., 1993; Gornushkin et al., 2017) and is not within the scope of this work. Further implementation of a switch dataset is discussed in Section 6.

Augmented Dataset. This leads to the third dataset type that was tested. The augmented dataset was generated by combining the simulated dataset with the predictions generated from preliminary DNNs. This was done carefully to avoid the risk of mode collapse, a phenomenon where models trained on AI-generated data experience compounding degradation in quality over successive generations (Yao et al., 2023; Zhang, 2021; Shumailov et al., 2024). As a result, there is notable degradation in the switch predictions from the AUG-trained model, as shown in Figure 5.1. The

perfectly integrated Gaussian switch used in the first-generation SS training dataset is shown next to some of the predictions generated after being trained on the AUG dataset.

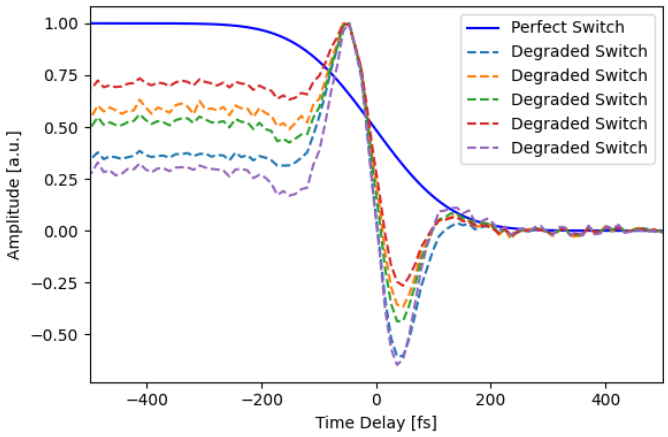

Figure 5.1 : Degenerating Switch Predictions.

Switch predictions generated by L1-XP-W when trained on the augmented dataset. The perfect switch is shown in blue, while the degraded switches are presented in a variety of dashed colors. The significant degradation between the original switch profile and the second-generation predictions indicates that the DNN undergoes mode collapse.

There was no significant improvements when the test cases were trained on this augmented dataset. From row 3 of Figure 4.3a, the metrics on simulated data remain very similar to those trained on purely simulated datasets. However, for experimental validation, the electric field error increases for both cases to around 10%. The switch recovery was unchanged at 15.92% for L1-XP-W but increased to 20.22% for L3-XP, while the trace error remained unchanged. From the predictions generated by these DNNs, it can be seen that degeneracy is already beginning on the switch profiles. While the predicted electric fields are not perfect matches to their labels, they are still simple pulses that are mathematically valid. However, the switch profiles took on increasingly diverse behaviors, as shown in Figure 5.1. The predicted switch profiles from the AUG-trained networks exhibit shapes that appear qualitatively similar to the derivative of the original switching function, which is not reflective of naturally occurring switch behaviors. This suggests that the DNN may be partially responding to the slope of the switching transition rather than its absolute amplitude.

One possible explanation is the combined effects of the Adam optimizer (Kingma et al., 2015) and the loss function used during training (described in Section 3.3). Gradient-based optimization methods, such as Adam (Kingma et al., 2015), adaptively scale kernel parameter updates according to the magnitude of local gradients (Bishop, 2006; Goodfellow et al., 2016). In the case of the switch, which exhibits sharp temporal transitions, these high-gradient regions dominate the loss, effectively biasing the network to fit the slope of the derivative rather than its absolute intensity. This effect is amplified by pointwise loss functions, such as MAE or mean squared error (Li et al., 2025; Goodfellow et al., 2016; Bishop, 2006)„ which penalize rapid local deviations more strongly than smooth global offsets. Consequently, the trained model can produce predictions that resemble the

temporal derivative of the switch function, an artifact that arises from Adam's inherent sensitivity to steep variations in the loss landscape (Goodfellow et al., 2016; Li et al., 2025; Kingma et al., 2015; Bishop, 2006).

When the model was trained using first-generation data, this deviation was minimal, as demonstrated with the successful predictions acquired from the SS-trained model in Figures 4.1 and 4.2. However, when trained on those successful predictions, the AUG model clearly tends towards the behavior demonstrated in Figure 5.1. While the AUG model's predictions are still considered stable, this shows how low-quality training datasets can easily impact the effectiveness of the training process (Goodfellow et al., 2016; Li et al., 2025). This indicates that simply using the predictions from previously trained models introduces a clear risk of degeneration and should be avoided with the current training methodology. As a result, further exploration into AI-generated training datasets was not included within the scope of this work.

5.2 Overcoming the Sim-to-Real Gap

Examining the results from the SS trained L1-XP-W procedure in Figure 4.6, it is shown that the proposed methodology was able to recover the temporal amplitude and phase of the pulse with a minimum of 5.02% error. However, further improvements were achieved with the application of techniques used to reduce the sim-to-real gap. In order to optimize predictions on experimental data while still training on simulated datasets, two techniques were considered : the addition of WGN and the UL procedure discussed in Section 3.3. The effect of both are discussed below.

White Gaussian Noise. Adding WGN to the traces had an interesting effect on the training procedure. In this, test case L1-XP-W outperformed L3-XP significantly. As a result, L1-XP-W was made the focus of the optimization process. As seen in Figure 4.4a, there is very little impact on the simulation validation set. The switch and electric field recovery remains relatively constant, while there is an insignificant decrease in trace error as the noise increases. Meanwhile, Figure 4.4b shows that there is a significant impact on the experimental validation. The trace error remains relatively constant, despite the variations in the electric field and the switch recoveries. The electric field error increases quickly from a minimum of 4.73% to a maximum of 10.46% over a range of 3.42 to 20 dB. Similarly, the switch recovery error increases from 15.44% to 16.47% over nearly the same range. Interestingly, both recovery errors decreased again after this region.

From this, it can be seen that adding WGN to the SS training dataset does not yield significant improvements in the final predictions. It is, however, useful to consider that the DNN was already so effective at recovering the electric field that the additional complexity of the WGN did not significantly alter the convergence path taken during training. This indicates a relatively decisive convergence that was not greatly altered by the perturbations introduced by the WGN.

Conversely, from Figure 4.6, it can be seen that the addition of WGN has a strong positive effect on the training procedures with the MS and AUG datasets. When the noise was combined with the UL procedure, the DNNs trained on the more complex datasets performed equivalently to those trained on the SS dataset. This was accompanied with an insignificant decrease in the switch error, resulting from the slightly larger dataset of switches. Overall, the addition of WGN did improve the MS and AUG trainings but did still not significantly improve the switch predictions.

Unsupervised Learning Procedure. We now consider the UL procedure that was presented in Figure 4.5. The expectation is that the underlying mathematical relation (Equation 3.8) will assist in convergence. We examine first the impact of the hybridization parameter on test case L1-XP-W in Figure 4.5.

From Figure 4.5a, it can be seen that the UL has very little impact on simulated validation sets. The electric field and switch errors remain relatively constant, while the trace error increases very slightly. This was anticipated, knowing that simulated data was well-approximated with the SL procedure. From Figure 4.5b, it can be seen that the hybridization of the SL and UL procedures had a relatively significant impact on the electric field recovery. As with the simulated dataset, the trace and switch errors remain relatively constant. Meanwhile, the electric field error followed a general trend where it first decreased slightly before increasing again. Interestingly, the electric field error approached a minimum of 4.73%, which outperforms the SL procedure. In fact, almost all the UL cases, independent of the hybridization parameter used, outperformed the SL. From this, we can see that the hybridization of the SL and UL procedures produces an improved DNN. This was further validated in Figure 4.6, where the DNN trained on MS and AUG datasets with the combined WGN and UL procedures performed equivalently with the SS dataset.

As stated previously, all test cases seem to have plateau'd at a minimum electric field error of 4.73% error (seen by the C-MS case). A final optimization step was taken using the `Optuna` objective to produce the best case presented in Table 4.2. Minimum errors of 15.11%, 4.39%, and 80.92% for the switch, electric field, and FROSt traces are achieved with the DFROStNET parameters in Table 4.2.

5.3 Switch Recovery : Another Phase Retrieval Problem

We've shown that the error on the electric field recovery is acceptably low for our applications, indicating that DFROStNET was successfully trained under this methodology. However, we've noted that the switch recovery remains above 15%, meaning DFROStNET is not able to accurately retrieve the switch. This, in turn, results in a large trace error. This is expected, as the dataset was based on the simulated electric fields and had very little variation in the switches. The focus of this

work was on the recovery of the electric field and not the switch, because, in our applications, the material dynamics are simply a tool used to measure the pulse.

While DFROStNET achieves the necessary goal, it fails to perform equivalently to ptychography since it cannot recover the switch. Moreover, in ptychography, the recovery of the switch is an important validation step. Because ptychography recovers both vectors, a new reconstructed FROSt trace can be generated. This allows us to validate that the switch and electric field profiles do indeed generate a trace that resembles the experimental one. In fact, the process converges based on the error on the recovered FROSt trace. This cannot be achieved with DFROStNET, leaving no method to validate the correctness of the electric field. Additionally, for those interested in studying the switch profile, DFROStNET alone is useless, given its relatively high error on the switch recovery. Because of this limitation, DFROStNET falls short of fully replicating the behavior of ptychography.

The above limitation can be partially overcome by combining DFROStNET with the numerical processed used in ptychography. In Section 2.2, we saw that an important step of ptychography is the application of the ptychographic constraint (see Step (a), Figure 2.3). This constraint is applied when the complex 2D electric field of the object plane, $E(t,\tau)$, is defined and one of the two input vectors (either pulse or switch) is known. In our case, by solving the Equation 2.9, the switch profile can be recovered. However, this introduces an important consideration that was determined to be outside the scope of this work. For our applications, we've modeled the switch through its intensity, as that is of importance to recover the electric field of the pulse. In reality, the switch would have a real and imaginary component (i.e., an amplitude and phase), just as the pulse does. This introduces another PRP to recover the switch.

It is important to note that in order to recover the complex switch, the ptychographic constraint can accurately be applied only if we have the complex object plane field, $E(t,\tau) \in \mathbb{C}$. In ptychography, this value can be calculated at each iteration from the estimations. However, as usual, experimentally, we are limited to measuring the intensity of any quantity. This means our closest approximation to the experimental object plane, $E_{XP}(t,\tau)$, is by performing an inverse FT on the real-valued FROSt trace : $E_{XP}(t,\tau) = \frac{1}{2\pi}\int_\omega \sqrt{I_{XP}(\omega,\tau)}\, e^{j\omega t}\, d\omega$ (essentially, reversing Equation 2.4). From this, Equation 2.9 can be rewritten as is shown in Equation 5.1, where the switch is a function of the shifted AI-predicted electric field, $E_{P,\,AI}(t+\tau)$, and its complex conjugate, $E^*_{P,\,AI}(t+\tau)$.

$$S_{AI}(t) = \frac{\int_\tau E^*_{P,\,AI}(t+\tau)\, E_{XP}(t+\tau,\tau)\, d\tau}{\int_\tau |E_{P,\,AI}(t+\tau)|^2\, d\tau} \tag{5.1}$$

From Equation 5.1, a complex switch profile can be recovered. However, it is important to understand that this is but one valid solution from the subset of valid solutions and not necessarily the one that minimizes the error. We lack additional information in order to identify the most valid solution. In fact, the recovered switch profile is an unlikely solution, as it assumes the perfect imaginary component to cancel out the phase of the pulse. This assumption is necessary to be made,

as only the real-valued FROSt trace is known, but is experimentally unlikely. While not a perfect solution, the application of the ptychographic constraint allows us to get a first estimate of the switch behavior. As a result, by applying the ptychographic constraint, we are able to also recover the intensity of the switch profile. This shows that DFROStNET is a versatile and useful tool for experimental characterization of ultrafast laser pulses.

With the ptychographic constraint applied, DFROStNET is more closely comparable to ptychography. However, ptychography still holds the benefit of the validation step, provided by having an approximation on both the electric field and switch. With DFROStNET, we rely on the accuracy of the AI-predicted electric field to recover the switch (and we know that the recovered switch is but one of a class of solutions). However, this work shows that DFROStNET can reliably reproduce the pulse recovered from the ptychographic method. The framework of these two processes are compared in the Figure 5.2. An experimental FROSt trace is shown passing through the traditional ptychographic methodology and DFROStNET. For the ptychographic method, the raw trace is first pre-processed which requires user input and takes time. These inputs include estimates of the central frequency, as well as spectral and temporal filter properties (see Section 3.4 for details on pre-processing methods). This pre-processing will produce the Figure 5.2Pa, which can then be input to the iterative ptychographic recovery. Once the computational time has passed, the electric field and switch are recovered. These are input to FROStNET to produce the reconstructed FROSt trace. Meanwhile, for the AI procedure, the raw FROSt trace is input to DFROStNET. This outputs an accurate prediction on the electric field and a relatively nonsense prediction on the switch, which is discarded. The electric field and the experimental FROSt trace are used to solve for a switch profile using the ptychographic constraint. As a result, these two processes extract very similar information from the trace with minimal error. Ptychography still maintains the benefits of the validation step provided by the dual output. However, DFROStNET provides a reliably accurate electric field in real time. In many cases, the proposed methodology provides unique benefits when compared to the traditional ptychography.

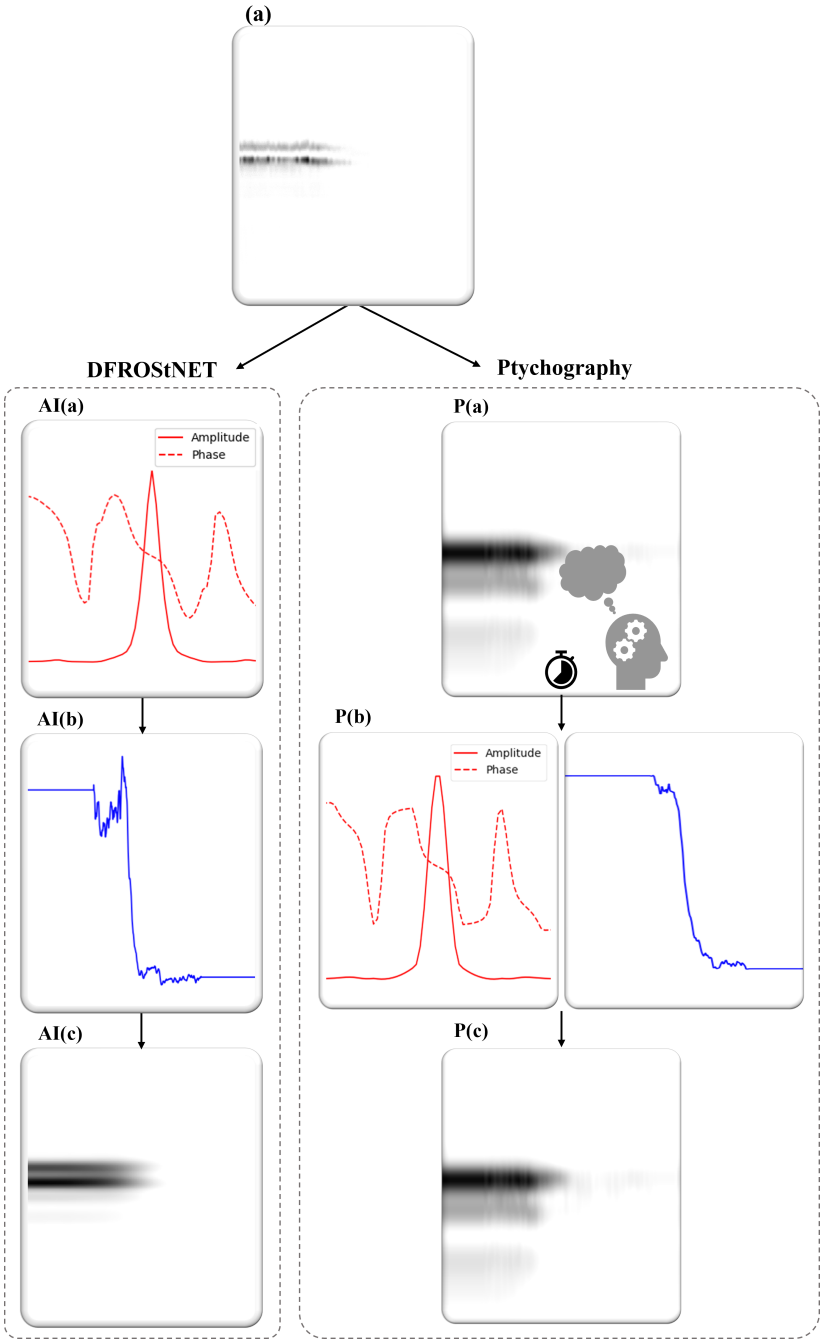

Figure 5.2 : Comparison of DFROStNET and Ptychographic Processes.

The procedures used for both the AI (AI(a) - (c)) and the ptychographic (P(a) - (c)) processes are shown. In (a), the raw experimental FROSt trace is presented. For the AI procedure, the trace is input directly to the DNN which produces a reliable prediction on the electric field, as shown in AI(a). By applying the ptychographic constraint in Equation 5.1, the switch in AI(b) is calculated. From the electric field and switch, the recovered FROSt trace in AI(c) is determined using FROStNET. Meanwhile, for the ptychographic process, the experimental trace undergoes pre-processing to recover the processed trace in P(a). Human intervention and time is necessary to achieve this. This is input to the ptychographic recovery which, after some time, generates the electric field and the switch, in P(b). These can be recombined using FROStNET to generate the recovered FROSt trace in P(c). The two processes are compared to show the additional steps and inputs required for the ptychography.

6 CONCLUSIONS AND NEXT STEPS

6.1 In This Work : DFROStNET

The goal of this work, as stated in Section 1.5, was to generate a methodology by which a DNN could be trained to recover the amplitude and phase of a pulse from a FROSt TFR with an accuracy comparable to the standard ptychographic recovery. As shown in Section 4, DFROStNET is able to recover both the amplitude and phase of the pulse within an error of 4.39%. This demonstrates that the goal was successfully achieved. This indicates that the methodology developed to create the training datasets (described in Section 3.1) was successful in imitating real electric fields. Moreover, the architecture that was inspired by Zahavy et al. (2018) was efficient in de-convolving the FROSt traces. From the test cases, it is shown that the L1-XP-W procedure was able to recover the electric field from experimental traces with a very low error when compared to ptychography. The inclusion of WGN and the UL procedure were able to further improve this to less than 5% error on more complex datasets. Overall, DFROStNET meets the goals set for this work. While it is true that the DNN methodology is not able to accurately recover the switch profile, the application of the ptychographic constraint and its assumptions (as described in Section 5.3) partially remedies this short-coming. As a result, we have achieved more than was stated in the goal. Together, this shows that DFROStNET is able to rival ptychography in accuracy.

Moreover, the proposed methodology provides some unique benefits when compared to ptychography. Notably, DFROStNET can generate a prediction in real-time, while the ptychographic method requires a significant amount of computational time to achieve the same recovery. The AI method does require some computations but this is considered trivial in comparison to the iterative computations required for ptychography. Most importantly, DFROStNET can be achieved without the use of pre-processing steps. This is incredibly important, as it removes the effects of researcher bias. When performing the pre-processing, inputs such as the spectral bandwidth, central frequency, temporal window, and filter properties are required. The act of selecting these variables, while seemingly innocent, introduces a bias from the researcher that alters the optimization path taken by the recovery. When the same experimental trace is pre-processed with different inputs, the ptychographic output changes. DFROStNET removes this bias to generate a more robust prediction, meaning that the recovery will always produce the same output from an input. We can also consider this beneficial as it removes the necessity of human intervention before producing the prediction. Using the traditional ptychographic method, it is impractical to generate a prediction while actively collecting data. Generally, the recovery is achieved is post-processing after data collection has been completed. The ability to acquire data while simultaneously recovering its entire electric field proves to be a unique benefit of the AI methodology. These unique benefits collectively make DFROStNET a competitive alternative to ptychographic recovery.

6.2 Developing Training Datasets

While DFROStNET already provides a unique set of benefits compared to the standard alternatives, further improvements and future developments can be made. One significant weakness that has been discussed is the switch recovery. It has been clearly established that the electric field dataset alone is inadequate to train DFROStNET to recover the switch. Moreover, other methods of data generation exist by using architectures that were not explored in this work such as the generative adversarial network (GAN). We discuss possible improvements for the training dataset by, first, including switch profiles and, second, different generation methods.

Switch Dataset. As has been discussed in Section 5.3, DFROStNET is not trained to accurately predict the switch. While a workaround can be achieved with the ptychographic constraint described previously, we are again limited to the intensity of the switch and it does not provide a safety net of a validation step. To amend this issue, DFROStNET must be trained on a more diverse dataset of switches than the simple 300 perfect Gaussian curves presented. This requires generating a dataset on the scale of $N = 2^{14} = 65536$ unique switches that accurately and realistically model the material dynamics of the switch. Just as our data generation for the electric fields required validation, it is necessary that the dataset accurately model the population inversion of the switch, which can be complex and poorly understood (Satou et al., 2011; Amstrup et al., 1993; Gornushkin et al., 2017). Importantly, in experimental conditions, these dynamics will depend on the pump intensity and frequency, and switch variables such as semiconducting material, doping ratios and thickness. This author would suggest generating a dataset using analytical solutions for the Optical Bloch Equations with input parameters varied to describe a number of possible cases (Noh et al., 2010). This would allow the physical properties of switch dynamics to be modeled at a large scale, but requires a more thorough study of laser-matter interaction than the scope of this work allows. Developing a high-quality dataset of switches to marry with the dataset of electric fields from this work would allow an enormous number of unique FROSt traces to be generated.

Generative Adversarial Network. While generating a dataset of switches would be useful to recover the switch profiles accurately, the entire AI procedure would likely benefit from relying less on simulated datasets. This was attempted in this work by using the predictions generated from a preliminarily trained DNN to create the AUG dataset. However, this suffered from significant degradation after just one generation of predictions (see Figure 5.1). Instead, data generation for both the pulse and switch could be achieved using an architecture such as a GAN (Goodfellow et al., 2016; Zhang, 2021; Srivastava et al., 2017). Briefly, GAN is an AI architecture that uses two competing NNs to generate realistic data. The two NNs, one being the generator and the other the discriminator, work in competition with each other. A much smallest dataset can be fed into the GAN, where the generative NN acts very similarly to the architecture used in this work. From the simulated dataset, the generative NN will produce a predicted data point that is

evaluated by the discriminator. In early stages of training, the predictions will be sufficiently poor that the discriminator will easily identify as dissimilar from the training dataset. However, as the generator is trained, the predictions will begin to resemble more closely the training data provided to the discriminator. Eventually, the predictions will closely resemble the training data and the discriminator will begin to consistently classify the predictions as true, at which point training is halted. This process simultaneously trains a DNN while also creating new predictions that are sufficiently similar to the training data to qualify as "real" data points. Essentially, this process trains two DNNs in parallel, where one is trained to make predictions (as the DNN does in this work) while the other is trained to differentiate between those predictions and true data. This requires, of course, the use of two loss functions that back-propagate to update the DNNs individually. GANs are a relatively complex class of NNs and are presented in detail elsewhere (Brunton et al., 2017; LeCun et al., 2015; Alzubaidi et al., 2021; Goodfellow et al., 2016). We can note, however, that GANs are a more systematic and robust version of the AUG dataset generation used in this work. GANs generally prove to be more stable than what was seen here, though they are still relatively vulnerable to mode collapse (Srivastava et al., 2017; Zhang, 2021).

Training Times. As we discuss training more complex architectures on exponentially larger datasets, it is necessary to acknowledge that training DNNs takes significant computational time (as was highlighted in Table 4.1). Already for the proposed work, the optimal test case takes between 30 min to 1 h to train, depending on parameters (e.g., training on WGN and UL processes take longer). This is the time taken to batch the electric fields, pass the input vectors to FROStNET, and train DFROStNET. Now consider the same architecture being trained on the combined training dataset of 65536 switches and 65536 electric fields, which would produce $65536 \times 65536 = 4.3 \times 10^9$ FROSt traces. The computational times in Table 4.1 would increase exponentially to process and train on a dataset of this size. The computation time and required available memory becomes increasingly difficult to manage, especially because studying the effect of training parameters requires training many different models. It can be expected that trying to train a DNN to identify both the switch and pulse would require consideration to this resource. It may be beneficial to consider the use of a newer GPU or even renting time on a supercomputer that runs on tensor processing units (TPUs) which are designed to handle the matrix-heavy operations used in AI training (Nikolić et al., 2022). This would be a fundamental necessity to handle the complex computations required to achieve the goal of training on both the pulses and switch profiles.

6.3 Experimental Implementations

While further improvements can be made to the switch recovery, DFROStNET is capable of accurately identifying the electric field of the pulse from the unprocessed FROSt trace, which is already very experimentally useful. Below, we discuss some next steps required to implement DFROStNET

experimentally. Additionally, we propose some possible future experimental techniques that may be of interest.

User Interface. Since we've achieved real-time recovery, the first step to implementation is simply integration into the established experimental data collection. This would require implementation into a UI, which would be most effective if integrated directly into the experimental FROSt data collection set-up (shown in Figure 1.5). By integrating the DNN directly into the data collection procedure, the predictions can be generated in real-time from the pre-established infrastructure. When a FROSt trace is finished being acquired experimentally, the data acquisition program would proceed directly to recovery using DFROStNET. As shown in Figure 5.2, the pulse is recovered, then the switch approximation is made, and both can be plotted in real-time with no intervention from the researcher required. Not only is the recovery time reduced, but the pre-processing is no longer necessary to acquire an accurate pulse profile. This means real-time predictions can be generated near-instantaneously for researchers. When using ptychography, an estimate must be made in the lab, with the actual recovery acquired later. With the AI method, the response time is immediate, allowing the researcher to make modifications. This would facilitate optimization and alignment, as the system feedback is constantly provided to the researcher. A similar software is available for FROG recovery (DeLong et al., 1994; DeLong, 2009). Developing a similar UI for data acquisition and recovery would significantly streamline experimental procedures.

One-shot Techniques. With DFROStNET integrated into a real-time UI, experimental data acquisition can be made more complex. We can use this tool and pre-existing single-shot techniques to further improve our experimental characterization methods. Many TFR data acquisition methodologies have been simplified to single-shot versions, where the variable delay line is no longer necessary (Brizard et al., 2020; Louisy et al., 2017; Kane et al., 1993). These techniques allow for entire TFRs to be acquired more quickly and with a stationary set-up. Just as GRENOUILLE is a stationary, simplified version of FROG (O'Shea et al., 2004; Trebino et al., 2021), FROSt can be simplified to a single-shot acquisition (Brizard et al., 2020). Integrating this single-shot FROSt acquisition with the real-time DFROStNET recovery would result in the full, near-instantaneous characterization of an optical system. With the development of the DFROStNET UI, FROSt becomes equivalently competitive to FROG. The ptychographic recovery will no longer limit the applications of FROSt, making it even more versatile and accessible than previously. In some cases, FROSt would strongly outperform FROG (e.g., characterizing low-intensity and heavily chirped pulses).

Experimental Switch Recovery. Throughout this work, the focus has been primarily on the recovery of the pulse electric field. However, as we've discussed, FROSt can be used equivalently to characterize the switch. While DFROStNET and the ptychographic constraint are not able to recover the exact complex switch profile, this could be achieved experimentally.

Consider the experimental case where a probe and switch are both unknown. A series of steps can be taken experimentally to fully characterize the system. First, a FROSt trace is acquired using the traditional switch and DFROStNET is used to recover the complex pulse electric field. From this first trace, we solved for the pulse, $E_P(t)$. A second measurement is performed where the traditional switch is combined with a simple dispersive medium (e.g., glass). At this point, it is necessary to assume that there is no absorption in the dispersive medium. The specifics of this dispersive quantity, $\tilde{D}(\omega)$, are dependent on the materials' refractive index and its thickness, but independent of the temporal delay, τ. From these two measurements, we acquire the trace, $I_1(\omega,\tau)$, and its modulated partner, $I_2(\omega,\tau)$.

$$I_1(\omega,\tau) = \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt \right|^2$$
$$I_2(\omega,\tau) = \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt\, \tilde{D}(\omega) \right|^2$$

From these acquisitions, we create a system of equations that can be solved to identify the complex switch profile. First, we take the ratio between the two traces to recover the dispersion, $\tilde{D}(\omega)$. Because we assume there is no absorption, $\tilde{D}(\omega) \in \mathbb{R}$ such that, in this case, we can neglect the PRP.

$$\tilde{D}(\omega) = \left(\frac{I_2(\omega)}{I_1(\omega)} \right)^{1/2}$$

Now, the entire system has been described with the exception of the complex switch, $S(t-\tau)$. By taking the difference between the two traces, we generate an equation with a single unknown. We can set this equation to zero, as these two traces create a closed system.

$$I_2(\omega,\tau) - I_1(\omega,\tau) = \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt\, \tilde{D}(\omega) \right|^2 - \left| \int_t E_P(t)\, S(t-\tau)\, e^{j\omega t}\, dt \right|^2$$

Because the equation cannot be solved analytically, we let the above equation be equal to some error, ε, that we wish to minimize. A simple iterative program can be used to identify the switch profile that produces the smallest error ε. While this returns to a numerical solution, this is a much simpler alternative to ptychography. In fact, this single step is the computational equivalent of applying the ptychographic constraint, as was suggested in Section 5.3. However, because we've added an additional redundancy (i.e., the addition of the dispersion function, $\tilde{D}(\omega)$, in one of the two traces), we are able to recover the complex switch profile, just as is achieved in ptychography.

We hypothesize that this proposed experimental methodology would adequately recover the complex switch profile. In fact, it may be simpler to implement this experimental procedure than it would be to train DFROStNET to recovery the switch profile. The most significant limitation of the proposed experimental methodology is the time required to perform two FROSt scans, which would be minimized with the integration of the single-shot FROSt technique mentioned above

(Brizard et al., 2020). Single-shot detection would streamline the collection of the FROSt traces, while the combination of DFROStNET and this simple numerical recovery allows for the accurate and real-time recovery of both the pulse and switch electric fields.

Study of Material Dynamics. By integrating DFROStNET and the switch recovery procedure proposed above, FROSt becomes an incredibly versatile and powerful tool for characterizing optical systems and material dynamics. Historically, the focus of FROSt has been the characterization of the ultrafast pulse. However, it is known that the switch profile used in FROSt is a direct result of the dynamics of population inversion caused by the photo-excitation from the high-intensity pump. Population inversion is non-trivial to model, due to the complexity of the many-body interactions, the timescale in question, and resulting secondary nonlinear behaviors (such as thermal effects, relaxation, and recombination) (Amstrup et al., 1993; Satou et al., 2011; Gornushkin et al., 2017). Moreover, we know that the dynamic depends on the atomic and molecular structure of the material, as well as the properties of the high-intensity beam inducing the population inversion (Noh et al., 2010). As a result, these dynamics are extremely complex and condition-dependent, meaning it is difficult to model. Overall, it would be greatly beneficial to develop a tool that can characterize the dynamics of various materials under photo-excitation conditions.

With a well-characterized ultrafast pulse, FROSt provides an experimental tool to examine materials, as the pump induces the necessary inversion and the probe resolves the switch dynamics. Once the probe pulse is characterized as necessary, both the amplitude and phase of the switch can be recovered through a simple numerical minimization. From this switch profile, population inversion models can be confirmed, denied, or altered to better represent the true dynamics. The switch can then easily be exchanged in order to study and compare the behaviors of different materials at various thicknesses under increasing pump intensities. The study of these dynamics can be achieved by implementing the experiment described in this work, as it minimizes variability between samples, improves the reproducibility, and reduces the reliance on complex numerical projections. Importantly, it cannot be achieved when relying on ptychography. Consider that for two FROSt traces, regardless of experimental conditions, ptychography will identify two different electric fields for the pulse, though we know this cannot be true. As such, it becomes impossible to know which pulse is truly probing the samples. Moreover, the two ptychographically-recovered switch profiles could not truly be compared, as they were theoretically produced by two different electric fields. As such, the implementation of DFROStNET and the experimental methodology described here would create a unique set-up for the study of material dynamics. Overall, the methodology of this work has presented opportunities to further strengthen and develop many fields, including ultrafast laser characterization, material studies, and machine learning.

BIBLIOGRAPHY

Ahmed M, Yamada M, Saito M (2001) Numerical modeling of intensity and phase noise in semiconductor lasers. *IEEE Journal of Quantum Electronics*, 37(12):1600–1610. DOI :10.1109/3.970907.

Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L (2021) Review of deep learning : Concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(53):1–73. DOI :10.1186/s40537-021-00444-8.

American S (2025) *Domestic Cat*. https ://www.scientificamerican.com/article/cats-recognize-their-own-names-even-if-they-choose-to-ignore-them/. Accessed : 2025-01-28.

Amstrup B, Lorincz A, Rice SA (1993) Population inversion in a multilevel system : A model study. *The Journal of Physical Chemistry*, 97(23). DOI :10.1021/j100125a015.

Anderson MH, Ensher JR, Matthews MR, Wieman CE, Cornell EA (1995) Observation of bose-einstein condensation in a dilute atomic vapor. *Science*, 269(5221):198–201. DOI :10.1126/science.269.5221.198.

Bishop C (2006) *Pattern recognition and machine learning*. volume 4. Springer New York.

Boashash B (2016) *Time-Frequency Signal Analysis and Processing : A Comprehensive Reference*. Academic Press.

Born M, Wolf E (2019) *Principles of Optics : Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, chapitre 7. Cambridge University Press.

Bourassin-Bouchet C, Mang MM, Gianani I, Walmsley IA (2013) Mutual interferometric characterization of a pair of independent electric fields. *Optics Letters*, 38(24):5299–5302. DOI :10.1364/OL.38.005408.

Bowlan P, Gabolde P, Trebino R (2007) Directly measuring the spatio-temporal electric field of focusing ultrashort pulses. *Optics Express*, 15(16):10219–10229. DOI :10.1364/OE.15.010219.

Brizard B, Leblanc A, Petit S, Delagnes JC, Éric Cormier, Ibrahim H, Légaré F, Lassonde P (2020) Single-shot phase-matching free ultrashort pulse characterization based on transient absorption in solids. *Optics Express*, 28(24):35807–35815. DOI :10.1364/OE.409342.

Brunton SL, Kutz JN (2017) *Data Driven Science & Engineering : Machine Learning, Dynamical Systems, and Control*, chapitre Part II : Machine Learning and Data Analysis. Cambridge University Press.

Carnio BN, Moutanabbir O, Elezzabi AY (2024) Advanced modeling of electro-optic sampling : Nonlinear vectoral-field solutions to maxwell's equations. *Advanced Physics Research*, 3(9):2400034. DOI :10.1002/apxr.202400034.

Chen J, Li P, Wang Y, Ku PC, Qu Q (2024) Sim2real in reconstructive spectroscopy : Deep learning with augmented device-informed data simulation. *APL Machine Learning*, 2(3):036106. DOI :10.1063/5.0209339.

Cireasa R, Boguslavskiy AE, Pons B, Wong MC, Descamps D, Petit S, Ruf H, Thiré N, Ferré A, Suarez J, Higuet J, Schmidt BE, Alharbi AF, Légaré F, Blanchet V, Fabre B, Patchkovskii S, Smirnova O, Mairesse Y, Bhardwaj VR (2015) Probing molecular chirality on a sub-femtosecond timescale. *Nature Physics Letters*, 11. DOI :10.1038/NPHYS3369.

DeLong KW (2009) *Frequency-Resolved Optical Gating Software.* https ://github.com/kenwdelong/frog. Accessed : 2025-01-29.

DeLong KW, Fittinghoff DN, Trebino R, Kohler B, Wilson K (1994) Pulse retrieval in frequency-resolved optical gating based on the method of generalized projections. *Optics Letters*, 19(24): 2152–2154. DOI :10.1364/OL.19.002152.

Diddams SA, Jones DJ, Ye J, Cundiff ST, Hall JL, Ranka JK, Windeler RS, Holzwarth R, Udem T, Hänsch TW (2000) Direct link between microwave and optical frequencies with a 300 thz femtosecond laser comb. *American Physical Society Physical Review Letters*, 84:5102–5105. DOI :10.1103/PhysRevLett.84.5102.

Diels JC, Rudolph W (2006) *Ultrashort Laser Pulse Phenomena Fundamentals, Techniques, and Applications on a Femtosecond Time Scale.* Academic Press.

Dorrer C, Walmsley IA (2001) Accuracy criterion for ultrashort pulse characterization techniques : application to spectral phase interferometry for direct electric field reconstruction. *Journal of the Optical Society of America B*, 19(5):1019–1029. DOI :10.1364/JOSAB.19.001019.

Errico C, Pierre J, Pezet S, Desailly Y, Lenkei Z, Couture O, Tanter M (2015) Ultrafast ultrasound localization microscopy for deep super-resolution vascular imaging. *Nature*, 527. DOI :10.1038/nature16066.

Fermann ME, Galvanauskas A, Sucha G, Harter D (1997) Fiber-lasers for ultrafast optics. *Applied Physics B : Lasers & Optics*, 65(2).

Fienup JR (1982) Phase retrieval algorithms : a comparison. *Applied Optics*, 21(15):2758–2769. DOI :10.1364/AO.21.002758.

Foster MA, Salem R, Geraghty DF, Turner-Foster AC, Lipson M, Gaeta AL (2008) Silicon-chip-based ultrafast optical oscilloscope. *Nature Letters*, 456:81–84. DOI :10.1038/nature07430.

Geographic N (2025) *Domestic Dog.* https ://www.nationalgeographic.com/animals/mammals/facts/domestic-dog. Accessed : 2025-01-28.

Gianani I, Walmsley IA, Barbieri M (2024) Spiderweb : A neural network approach to spectral phase interferometry. *Optics Letters*, 49(19):5415–5418. DOI :10.1364/OL.534767.

Gonzalez RC, Woods RE (2018) *Digital Image Processing*, chapitre 3, 5, 6, 11, 12. Pearson.

Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning.* MIT Press.

Gornushkin I, Glaus R, Nagli L (2017) Stimulated emission in aluminum laser-induced plasma : kinetic model of population inversion. *Applied Optics*, 56(3):695–701. DOI :10.1364/AO.56.000695.

Gravel P, Beaudoin G, Guise JAD (2004) A method for modeling noise in medical images. *IEEE Transactions on Medical Imaging*, 23(10):1221–1232. DOI :10.1109/TMI.2004.832656.

Hebden JC, Arridge SR, Delpy DT (1995) Optical imaging in medicine : I. experimental techniques. *Physics in Medicine & Biology*, 42(5):825. DOI :10.1088/0031-9155/42/5/007.

Hecht E (2017) *Optics*, chapitre 8, 9. Pearson.

Hospequip (2025) *Drinking Cup*. https ://www.hospequip.com.au/products/drinking-cup-crystal-clear-1208. Accessed : 2025-01-28.

Hoy CL, Ferhanoglu O, Yildirim M, Kim KH, Karajanagi SS, Chan KMC, Kobler JB, Zeitels SM, Ben-Yakar A (2013) Clinical ultrafast laser surgery : Recent advances and future directions. *IEEE Journal of Selected Topics in Quantum Electronics*, 20(2):242–255. DOI :10.1109/JSTQE.2013.2287098.

Huang J, Xia S, Chen Y, Li X, Wang K, Rui Y (2023) Experimental study of thermal damage to in vitro skin tissue welding by femtosecond laser. *Infrared Physics & Technology*, 129:104536. DOI :10.1016/j.infrared.2022.104536.

Jaganathan K, Eldar YC, Hassibi B (2016) Stft phase retrieval : Uniqueness guarantees and recovery algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):770–781. DOI :10.1109/JSTSP.2016.2549507.

Juhasz T, Loesel FH, Kurtz RM, Horvath C, Bille JF, Mourou G (1999) Corneal refractive surgery with femtosecond lasers. *IEEE Journal of Selected Topics in Quantum Electronics*, 5(4):902–910. DOI :10.1109/2944.796309.

Kane DJ, Trebino R (1993) Single-shot measurement of the intensity and phase of an arbitrary ultrashort pulse by using frequency-resolved optical gating. *Optics Letters*, 18(10):823–825. DOI :10.1364/OL.18.000823.

Kingma DP, Ba JL (2015) Adam : A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

Knox WH (2000) Ultrafast technology in telecommunications. *IEEE Journal of Selected Topics in Quantum Electronics*, 6(6):1273–1278. DOI :10.1109/2944.902178.

Leblanc A (2016) *Miroirs et réseaux plasmas en champs lasers ultra-intenses : génération d'harmoniques d'ordre élevé et de faisceaux d'électrons relativistes*. Phd thesis, Université Paris Saclay.

Leblanc A, Lassonde P, Petit S, Delagnes JC, Haddad E, Ernotte G, Bionta MR, Gruson V, Schmidt BE, Ibrahim H, Cormier E, Légaré F (2019) Phase-matching-free pulse retrieval based on transient absorption in solids. *Optics Express*, 27(28):28998–29009. DOI :10.1364/OE.27.028998.

Leblanc A, Longa A, Kumar M, Laramée A, Dansereau C, Ibrahim H, Lassonde P, Légaré F (2021) Temporal characterization of two-octave infrared pulses by frequency resolved optical switching. *Journal of Physics : Photonics*, 3:045002. DOI :10.1364/OE.27.010269.

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature*, 521. DOI :10.1038/nature14539.

Li C, Liu K, Liu S (2025) A survey of loss functions in deep learning. *Mathematics*, 13(15). DOI :10.3390/math13152417.

Liang H, Xie J, Huang B, Li Y, Sun B, Yang C (2025) A novel sim2real reinforcement learning algorithm for process control. *Reliability Engineering & System Safety*, 254:110639. DOI :10.1016/j.ress.2024.110639.

Liu JM (2016) *Principles of Photonics*, chapitre 11. Cambridge University Press.

Louisy M, Guo C, Neoričić L, Zhong S, L'Huillier A, Arnold CL, Miranda M (2017) Compact single-shot d-scan setup for the characterization of few-cycle laser pulses. *Applied Optics*, 56(32):9084–9089. DOI :10.1364/AO.56.009084.

Lucchini M, Nisoli M (2018) Refined ptychographic reconstruction of attosecond pulses. *Applied Sciences*, 8(12):2563. DOI :10.3390/app8122563.

Maiden AM, Humphry MJ, Zhang F, Rodenburg JM (2011) Superresolution imaging via ptychography. *Journal of the Optical Society of America A*, 28(4):604–612. DOI :10.1364/JOSAA.28.000604.

Mairesse Y, Quéré F (2005) Frequency-resolved optical gating for complete reconstruction of attosecond bursts. *American Physical Society Physical Review A*, 71(1):011401. DOI :10.1103/PhysRevA.71.011401.

Mallat S (2016) Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 374:20150203. DOI :10.1098/rsta.2015.0203.

Martínez A, Blasco J, Sanchis P, Galán JV, García-Rupérez J, Jordana E, Gautier P, Lebour Y, Hernández S, Spano R, Guider R, Daldosso N, Garrido B, Fedeli JM, Pavesi L, Martí J (2010) Ultrafast all-optical switching in a silicon-nanocrystal-based silicon slot waveguide at telecom wavelengths. *Nano Letters*, 10(4):1506–1511. DOI :10.1021/nl9041017.

Middleton D (1999) Non-gaussian noise models in signal processing for telecommunications : New methods and results for class a and class b noise models. *IEEE Transactions on Information Theory*, 45(4):1129–1149. DOI :10.1109/18.761256.

Miranda M, Arnold CL, Fordell T, Silva F, Alonso B, Weigand R, L'Huillier A, Crespo H (2012) Characterization of broadband few-cycle laser pulses with the d-scan technique. *OSA Optics Express*, 20(17):18732–18743. DOI :10.1364/OE.20.018732.

Muybridge E (1899) *Animals in Motion : An Electro-Photographic Investigation of Consecutive Phases of Animal Progressive Movements*. William Clowes and Sons.

Naganuma K, Mogi K, Yamada H (1989) General method for ultrashort light pulse chirp measurement. *IEEE Journal of Quantum Electronics*, 25(6):1225–1233. DOI :10.1109/3.29252.

Niikura H, Légaré F, Hasbani R, Bandrauk AD, Ivanov MY, Villeneuve DM, Corkum PB (2002) Sub-laser-cycle electron pulses for probing molecular dynamics. *Nature*, 417. DOI :10.1038/nature00787.

Nikolić GS, Dimitrijević BR, Nikolić TR, Stojcev MK (2022) A survey of three types of processing units : Cpu, gpu and tpu. *57th International Scientific Conference on Information, Communication and Energy Systems and Technologies*.

Noh HR, Jhe W (2010) Analytic solutions of the optical bloch equations. *Optics Communications*, 283(11):2353–2355. DOI :https ://doi.org/10.1016/j.optcom.2010.01.069.

Oppenheim AV, Willsky AS (2007) *Signals and Systems*, chapitre 7. Prentice Hall.

Ou X, Horstmeyer R, Yang C, Zheng G (2013) Quantitative phase imaging via fourier ptychographic microscopy. *Optics Letters*, 38(22):4845–4848. DOI :10.1364/OL.38.004845.

O'Shea P, Akturk S, Kimmel M, Trebino R (2004) Practical issues in ultra-short-pulse measurements with 'grenouille'. *Applied Physics B*, 79(1):683–691. DOI :10.1007/s00340-004-1644-x.

Pogna EAA, Celebrano M, Mazzanti A, Ghirardini L, Carletti L, Marino G, Schirato A, Viola D, Laporta P, Angelis CD, Leo G, Cerullo G, Finazzi M, Valle GD (2021) Ultrafast, all optically reconfigurable, nonlinear nanoantenna. *ACS Nano*, 15(7). DOI :10.1021/acsnano.1c03386.

Rodenburg JM (2008) Ptychography and related diffractive imaging methods. *Advances in Imaging and Electron Physics*, 150:87–184. DOI :10.1016/S1076-5670(07)00003-1.

Rosker MJ, Dantus M, Zewail AH (1987) Real-time femtosecond probing of chemical reactions. *The Journal of Chemical Physics*, 87(4):2395–2397. DOI :10.1063/1.453122.

Saleh B, Teich M (2019) *Fundamentals of Photonics*. Wiley Series in Pure and Applied Optics. Wiley.

Satou A, Otsuji T, Ryzhii V (2011) Theoretical study of population inversion in graphene under pulse excitation. *Japanese Journal of Applied Physics*, 50(7R). DOI :10.1143/JJAP.50.070116.

Scholten AJ, Tiemeijer LF, van Langevelde R, Havens RJ, van Duijnhoven ATAZ, Venezia VC (2003) Noise modeling for rf cmos circuit simulation. *IEEE Transactions on Electron Devices*, 50(3):618–632. DOI :10.1109/TED.2003.810480.

Shashua SDC, Mannor S, Castro DD (2021) Sim and real : Better together. *35th Conference on Neural Information Processing Systems*.

Shumailov I, Shumaylov Z, Zhao Y, Papernot N, Anderson R, Gal Y (2024) Ai models collapse when trained on recursively generated data. *Nature*, 631. DOI :10.1038/s41586-024-07566-y.

Sinelnik A, Lam SH, Coviello F, Klimmer S, Valle GD, Choi DY, Pertsch T, Soavi G, Staude I (2024) Ultrafast all-optical second harmonic wavefront shaping. *Nature Communications*, 15. DOI :10.1038/s41467-024-46642-9.

Sprangle P, Esarey E, Ting A, Joyce G (1988) Laser wakefield acceleration and relativistic optical guiding. *Applied Physics Letters*, 53(22):2146–2148.

Srivastava A, Valkov L, Russell C, Gutmann MU, Sutton C (2017) Veegan : Reducing mode collapse in gans using implicit variational learning. *31st Conference on Neural Information Processing Systems*, volume 30.

Stein EM, Shakarchi R (2007) *Complex Analysis*, chapitre 1. Princeton University Press.

Stibenz G, Steinmeyer G (2006) Optimizing spectral phase interferometry for direct electric-field reconstruction. *Review of Scientific Instruments*, 77(7):073105. DOI :10.1063/1.2221511.

Strickland D, Mourou G (1985) Compression of amplified chirped optical pulses. *Optics Communications*, 55(6):447–449. DOI :10.1016/0030-4018(85)90151-8.

Sugioka K, Cheng Y (2014) Ultrafast lasers—reliable tools for advanced materials processing. *Nature - Light : Science & Applications*, 3(4):e149–e149.

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. *ImageNet Large Scale Visual Recognition Challenge*.

Thibault P, Dierolf M, Bunk O, Menzel A, Pfeiffer F (2009) Probe retrieval in ptychographic coherent diffractive imaging. *Ultramicroscopy*, 109(4):338–343. DOI :10.1016/j.ultramic.2008.12.011.

Trebino R (2002) *Frequency-Resolved Optical Gating : The Measurement of Ultrashort Laser Pulses*. Springer.

Trebino R, DeLong KW, Fittinghoff DN, Sweetser JN, Krumbügel MA, Richman BA (1997) Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating. *Review of Scientific Instruments*, 68(9):3277–3295. DOI :10.1063/1.1148286.

Trebino R, Jafari R, Piksarv P, Bowlan P, Valtna-Lukner H, Saari P, Guang Z, Steinmeyer G (2021) *Handbook of Laser Technology and Applications*, chapitre The Measurement of Ultrashort Laser Pulses. CRC Press.

van der Valk NCJ, Wenckebach T, Planken PCM (2004) Full mathematical description of electro-optic detection in optically isotropic crystals. *Journal of the Optical Society of America B*, 21(3):622–631. DOI :10.1364/OL.28.002229.

Vodungbo B, Gautier J, Lambert G, Sardinha AB, Lozano M, Sebban S, Ducousso M, Boutu W, Li K, Tudu B, Tortarolo M, Hawaldar R, Delaunay R, López-Flores V, Arabski J, Boeglin C, Merdji H, Zeitoun P, Lüning J (2012) Laser-induced ultrafast demagnetization in the presence of a nanoscale magnetic domain network. *Nature Communications*, 3(999). DOI :10.1038/ncomms2007.

von Korff Schmising C, Pfau B, Schneider M, Günther CM, Giovannella M, Perron J, Vodungbo B, Müller L, Capotondi F, Pedersoli E, Mahne N, Lüning J, Eisebitt S (2014) Imaging ultrafast demagnetization dynamics after a spatially localized optical excitation. *Physical Review Letters*, 112(21). DOI :10.1103/PhysRevLett.112.217203.

Wada O (2004) Femtosecond all-optical devices for ultrafast communication and signal processing. *New Journal of Physics*, 6(1):183. DOI :10.1088/1367-2630/6/1/183.

Yao Z, Wang Y, Wu H, Wang J, Long M (2023) Modernn : Harnessing spatiotemporal mode collapse in unsupervised predictive learning. *IEEE Transactions on Pattern Analysis and Machine Learning*, 45(11). DOI :10.1109/TPAMI.2023.3293145.

Yun SH, Kwok SJ (2017) Light in diagnosis, therapy and surgery. *Nature Biomedical Engineering*, 1(1):0008. DOI :10.1038/s41551-016-0008.

Zahavy T, Dikopoltsev A, Moss D, Haham GI, Cohen O, Mannor S, Segev M (2018) Deep learning reconstruction of ultrashort pulses. *Optica*, 5(5):2334–2536. DOI :10.1364/OPTICA.5.000666.

Zhang K (2021) On mode collapse in generative adversarial networks. *Artificial Neural Networks and Machine Learning – ICANN*, volume 12892.

Zhang T, Zhang K, Lin J, Louie WYG, Huang H (2022) Sim2real learning of obstacle avoidance for robotic manipulators in uncertain environments. *IEEE Robotics and Automation Letters*, 7(1):65–72. DOI :10.1109/LRA.2021.3116700.

Zhao W, Queralta JP, Westerlund T (2020) Sim-to-real transfer in deep reinforcement learning for robotics : a survey. *IEEE Symposium Series on Computational Intelligence*.

7 APPENDICES

7.1 Appendix A : Codes

In this appendix, the codes used in this work are presented, organized by their function. In Section 7.1.1, the codes used to generate the pulses and switch are shown. Section 7.1.2 presents the DNN architectures used. In Section 7.1.3, the training loop and validation processes are presented. Finally, PtyChoPy's pre-processing and applying the ptychographic constraint are shown in Section 7.1.4. It should be noted that this code has been simplified to be more easily readable. For this reason, the process cannot be directly implemented from what has been provided.

7.1.1 Data Generation

```
def generate_FTL_pulse(FWHM):
    # calculate spread
    sigma = (FWHM / (2 * np.sqrt(2 * np.log(2))))

    # generate Gaussian curve
    intensity = (1 / (sigma * np.sqrt(2 * np.pi))) * np.exp(-0.5*(t / sigma) ** 2)

    # return amplitude of Gaussian
    return np.sqrt(intensity)
```

Figure 7.1 : Generating an FTL Pulse.

This script shows the process used to generate the Gaussian envelop that makes up the FTL pulse. The width of the pulse varies with the input, which is the desired FWHM in fs. The function outputs a real-valued pulse.

```
def apply_phase(phase_coefficients):
    # make polynomial from random phase coefficients
    phase = np.polyval(phase_coefficients, omega)

    # take spectrum of FTL
    FTL_spectra = fft(generate_FTL_pulse(FWHM))
    chirped_spectra = (FTL_spectra * np.exp(1j * phase))

    # Convert chirped pulse back to time
    return ifft(chirped_spectra)

```

Figure 7.2 : Applying Spectral Phase to FTL Pulse.

This function shows the process used to apply the random spectral phase function to the FTL pulse acquired from the previous function. The input is the phase coefficients, which are evaluated as points of a polynomial. The output is a complex chirped pulse.

```

def generate_switch(FWHM):
    # calculate spread and generate Gaussian curve
    sigma = FWHM / (2 * np.sqrt(2 * np.log(2)))
    gauss = (np.exp(-0.5*(t ** 2 / (sigma ** 2))))

    # integrate over vector to increasing curve
    switch = np.zeros(len(gauss))
    for i in range(switch.size - 1):
        switch[i + 1] = switch[i] + gauss[i]
    switch = np.abs(switch)**2
    switch /= max(switch)
    return switch

```

Figure 7.3 : Generating the Switch Function.

The procedure used to generate the perfect switch functions. The Gaussian curve is generated with the desired FWHM, in fs, which is input to the function. The output is a real-valued, normalized, and integrated Gaussian curve.

7.1.2 Architecture

The codes generated to produce the architectures described in this work are presented below. This includes both the aparametric and parametric architectures.

```
def FROStNET(pulse, switch):
    # generating shifted versions of the switch
    switch_shift = numpy.sliding_window_view(switch, window_shape = len(pulse))

    # get FROSt trace
    return np.abs(fft(switch_shift * pulse))**2
```

Figure 7.4 : FROStNET Architecture.

The matrix operations used to generate the FROStNET architecture. The pulse and switch are input. These are manipulated to generate the real-valued FROSt trace, which is output.

```
def MR_block(inputs, N_filters, MR_kernels = [1, 3, 5], C_kernel = 3):
    feature_maps = []              # apply looped MR layers
    for i in range(len(MR_kernels)):
        x = tf.keras.layers.Conv2D(filters = N_filters, kernel_size = MR_kernels[i],
                                   padding = 'same', activation = 'relu')(inputs)
        feature_maps.append(x)

    # concatenated feature maps in a single matrix
    concat_maps = tf.keras.layers.Concatenate(axis=-1)(feature_maps)

    # apply lone convolution to concatenated feature maps
    stride = (concat_maps.shape[1] - 1) // (concat_maps.shape[1]//2 - 1)
    output = tf.keras.layers.Conv2D(filters = 2,
                                   kernel_size = C_kernel, strides = stride,
                                   padding = 'same', activation = 'relu')(concat_maps)
    return output

```

Figure 7.5 : Multi-Resolution Block Architecture.

The keras architecture used to generate the MR block. The three parallelized convolutions are applied first, followed by a second convolution after concatenation. The stride is calculated for each convolutional layer using Equation 2.13 The inputs are the FROSt trace and the desired number of filters in each block. The output is the resulting feature maps that were generated from the convolutions.

```python
def MR_model(stddev = 0, dropout_rate = 0, dense_neurons = 512, dense_layers = 1):

    # Establish dimensionality of FROSt inputs
    inputs = tf.keras.Input(shape=(points, points, 1))

    # Add white Gaussian noise to the input
    noisy_input = tf.keras.layers.GaussianNoise(stddev)(inputs)

    # Establish 3 MR blocks in series
    MR_1 = MR_block(noisy_input, N_filters = noisy_input.shape[-1])
    MR_2 = MR_block(MR_1, N_filters = MR_1.shape[-1])
    MR_3 = MR_block(MR_2, N_filters = MR_2.shape[-1])

    # Flatten MR_3 output and dropout
    x = tf.keras.layers.Flatten()(MR_3)
    x = tf.keras.layers.Dropout(dropout_rate)(x)

    # Connect to dense layers (repeat for the number of layers)
    for _ in range(dense_layers):
        x = tf.keras.layers.Dense(dense_neurons, activation = 'relu')(x)

    # Final dense layers for pulse and switch outputs
    switch_output = tf.keras.layers.Dense(points, activation = 'linear')(x)
    pulse_output = tf.keras.layers.Dense(points, activation = 'linear')(x)

    # Compile the model from the above layers
    return tf.keras.Model(inputs = inputs, outputs = [pulse_output, switch_output])

```

Figure 7.6 : Global Multi-Resolution Model Architecture.

The global architecture, made up of several MR blocks, is shown. The inputs are variables trialed through the `Optuna` study. The output is the compiled model. This process includes applying the WGN, passing through the three MR blocks, and the dense layers.

7.1.3 Training

The homemade training loop is presented here, as well as the experimental validation function. This was the function minimized during the `Optuna` trial, which is also shown.

```
def train(model_parameters, batch_size, learning_rate, hybrid):
    # Establish loss function, optimizer, and model
    optimizer = tf.keras.optimizers.Adam(learning_rate)
    loss_fun = tf.keras.losses.MeanAbsoluteError()
    model = create_MR_model(model_parameters)

    # Set up callbacks for early stopping
    early_stop = tf.keras.callbacks.EarlyStopping()
    callbacks = tf.keras.callbacks.CallbackList([early_stop])
    for epoch in range(epochs):         # start training loop
        for i in range(0, train_size, batch_size):
            # Get electric field and switch labels batch from file
            label_pulse, label_switch = batch_data(dataset_folder, i, batch_size)
            label_FROSt = FROStNET(label_pulse, label_switch)       # FROSt trace
            label_spectrum = np.abs(fft(label_pulse))**2            # Spectrum for XPC

            # Generate prediction on FROSt
            with tf.GradientTape(persistent = True) as tape:
                pred_phase, pred_switch = model(label_FROSt, training = True)
                # Reconstruct predicted pulse with XP constraint
                pred_pulse = ifft(tf.sqrt(label_spectrum)*tf.exp(pred_phase))
                # calculate error on each prediction with loss function
                switch_loss = loss_fun(label_switch, pred_switch)
                pulse_loss = loss_fun(label_pulse, pred_pulse)
                # calculate UL error
                trace_loss = np.abs(switch_loss*pulse_loss)**2
                total_loss = switch_loss + pulse_loss + hybrid*trace_loss

            # Generate loss gradient and back-propagate it
            gradient = tape.gradient(total_loss, model.trainable_weights)
            optimizer.apply_gradients(zip(gradient, model.trainable_weights))
        errors = validation(model)              # Validation loop
        if early_stop.stopped_epoch > 0:        # early stopping if training stagnates
            break
        model.save_weights(save_path)           # save the model at end of training
    return model

```

Figure 7.7 : Training Loop.

The homemade training loop is shown. This includes batching the data, generating traces, training, and validation. After the epochs are executed, the model is saved.

```python
def xp_validation(xp_FROSt, xp_pulse, xp_switch, model):
    # interpolate FROSt to 128 points and make prediction
    xp_FROSt = tf.image.resize(xp_FROSt, (N_t, N_w))
    pred_phase, pred_switch = model.predict(xp_FROSt)

    # reconstruct pulse using XPC and FROSt trace
    xp_spectrum = fft(xp_FROSt)
    pred_pulse_t = ifft(tf.sqrt(xp_spectrum)*tf.exp(pred_phase))
    pred_FROSt = FROStNET(pred_pulse_t, pred_switch)

    # calculate error on each prediction
    switch_loss = loss_fun(xp_switch, pred_switch)
    pulse_loss = loss_fun(xp_pulse, pred_pulse_t)
    trace_loss = np.sqrt(1 - np.sum(xp_FROSt*pred_FROSt)**2 /
                            np.sum(xp_FROSt**2)*np.sum(pred_FROSt**2))

    return switch_loss, pulse_loss, trace_loss
```

Figure 7.8 : Experimental Validation.

The experimental validation process is shown. A model generates predictions from experimental FROSt traces and the error on the switch, pulse, and FROSt trace is returned.

```python
def objective(trial):
    # Define the hyperparameters to tune
    num_layers = trial.suggest_int("num_layers", 1, 5)
    num_neurons = trial.suggest_int("num_neurons", 256, 2048)
    dropout_rate = trial.suggest_float("dropout_rate", 0.0, 0.5)
    learning_rate = trial.suggest_float("learning_rate", 1e-5, 1e-1)
    batch_size = trial.suggest_categorical("batch_size", [8, 16, 32, 64])
    wgn = trial.suggest_int("wgn", 5, 15)
    hybrid = trial.suggest_float("hybridization", 0.1, 0.5)

    # assign variables to training loop, perform validation
    parameters = [dropout_rate, wgn, num_layers, num_neurons]
    trained_model = training_loop(parameters, batch_size, learning_rate, hybrid)
    _, pulse_loss, _ = xp_validation(trained_model)
    return pulse_loss  # return variable for Optuna to minimize

```

Figure 7.9 : `Optuna` Objective.

The optimization study was executed to minimize the experimental pulse error. The varied parameters are presented, where the trial is used to generate the values for each parameter.

7.1.4 Ptychography

In this Section, we include some portions of the PtyChoPy program. Because much of it was implemented in C and wrapped in Python using `Cython`, it is not intuitive to read. As such, we've included only the portions written in Python (including the pre-processing and the ptychographic constraint).

```
1
2  def PtyChoPy_preprocessing(xp_trace, delay, freq):
3      # crop temporally
4      xp_trace = xp_trace[:, N_crop[0]:N_crop[-1]]
5      delay_crop = delay[N_crop[0]:N_crop[-1]]
6
7      # Remove outliers
8      threshold = np.mean(np.min(xp_trace))
9      ind_outliers = np.where(xp_trace < 4 * threshold)[0]
10     for index in ind_outliers:
11         xp_trace[index] = np.mean(xp_trace[index - 1], xp_trace[index + 1])
12
13     # Background removal
14     background = grey_opening(xp_trace, size = (threshold, threshold))
15     xp_trace = xp_trace - background
16
17     # Spectral filtering with hyper gaussian of order = 80
18     hypergauss_filter = np.exp(-0.5(freq / N_filter) ** 80)
19     xp_trace = ifft(fft(xp_trace) * hypergauss_filter)
20
21     # Temporal smoothing with gaussian curve of order 6
22     smoothing_vector = np.exp(-0.5(t - N_smoothing) / N_smoothing) ** 6)
23     xp_trace = convolve2d(xp_trace, smoothing_vector)
24
25     # Interpolate in time delay and frequency
26     new_f = np.linspace(min(freq), max(freq), new_N_f)
27     new_delay = np.linspace(min(delay), max(delay), new_N_delay)
28     processed_trace = interp2d(freq, delay, xp_trace, kind='cubic')(new_f, new_delay)
29     return new_f, new_delay, processed_trace
30
```

Figure 7.10 : Trace Pre-Processing.

The pre-processing steps taken to prepare the FROSt trace for PtyChoPy recovery are shown. Each trace was processed individually with parameters selected for its properties. The input is the raw trace and its axes, and the processed version of each is returned.

```python

def ptychography_constraint(trace, pulse, time, delay):
    # declare variables
    N = trace.shape[0]
    N0 = trace.shape[1]
    dt = time[1] - time[0]
    dt0 = delay[1] - delay[0]

    # shift E field vector
    N_pts_plus_decalage = int((N0 - 1)*(dt0 / dt))
    moitie = N_pts_plus_decalage//2
    M_decalage_t0 = np.zeros((N0, N), dtype=pulse.dtype)
    for row in range(N0):
        ind = int(row * dt0 / dt) + 1
        # First range of columns
        for col in range(ind - 1):
            if moitie <= col <= moitie + N - 1:
                M_decalage_t0[row, col - moitie] = pulse[0]
        # Second range of columns
        for n in range(ind - 1, ind + N - 2):
            if moitie <= n <= moitie + N - 1:
                M_decalage_t0[row, n - moitie] = pulse[n - ind + 1]
        # Third range of columns
        for n in range(ind + N - 2, N + N_pts_plus_decalage):
            if moitie <= n <= moitie + N - 1:
                M_decalage_t0[row, n - moitie] = pulse[N - 1]

    # Solve for switch
    M_Obj = np.sum(M_Obj, axis = 0)
    numerator = np.zeros((N0, N), dtype = np.complex128)
    for i in range(N0):
        numerator[i, :] = np.conj(M_Obj) * ifft(trace)[:, i]
    Obj_t = np.sum(numerator, axis = 1) / np.sum(np.abs(M_Obj) ** 2)
    Obj_t /= np.max(Obj_t)
    return Obj_t

```

Figure 7.11 : Application of Ptychographic Constraint.

The implementation of the ptychographic constraint is shown. The pulse is shifted relative to the FROSt trace, as is shown in Equation 5.1. The output is an estimate of the complex object.

7.2 Appendix B : Further Predictions

In addition to the results presented in the main body of this work, we include some additional examples of experimental predictions made by DFROStNET. A variety of traces were selected from the experimental dataset to demonstrate the versatility of DFROStNET.

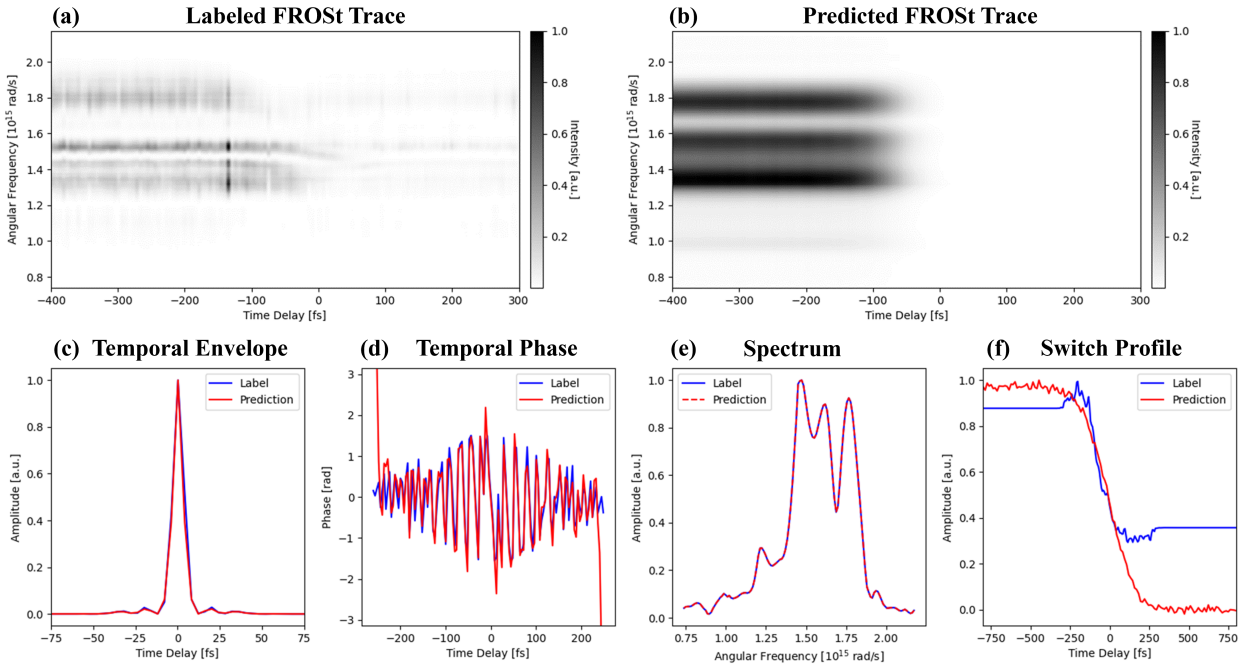

Figure 7.12 : Example 1 of Experimental Prediction by DFROStNET.

An experimental FROSt trace was taken as shown in (a). The reconstructed trace, generated from the DNN predictions, is shown in (b). The labeled (in blue) and predicted (in red) vectors for the pulse envelope and phase are shown in (c) and (d), respectively. The spectrum of the labeled and predicted electric field is presented in (e). The temporal amplitude of the labeled and predicted switch is shown in (f). For this case, the reconstruction errors on the switch, pulse, and trace are 21.34%, 3.71%, and 60.45%, respectively.

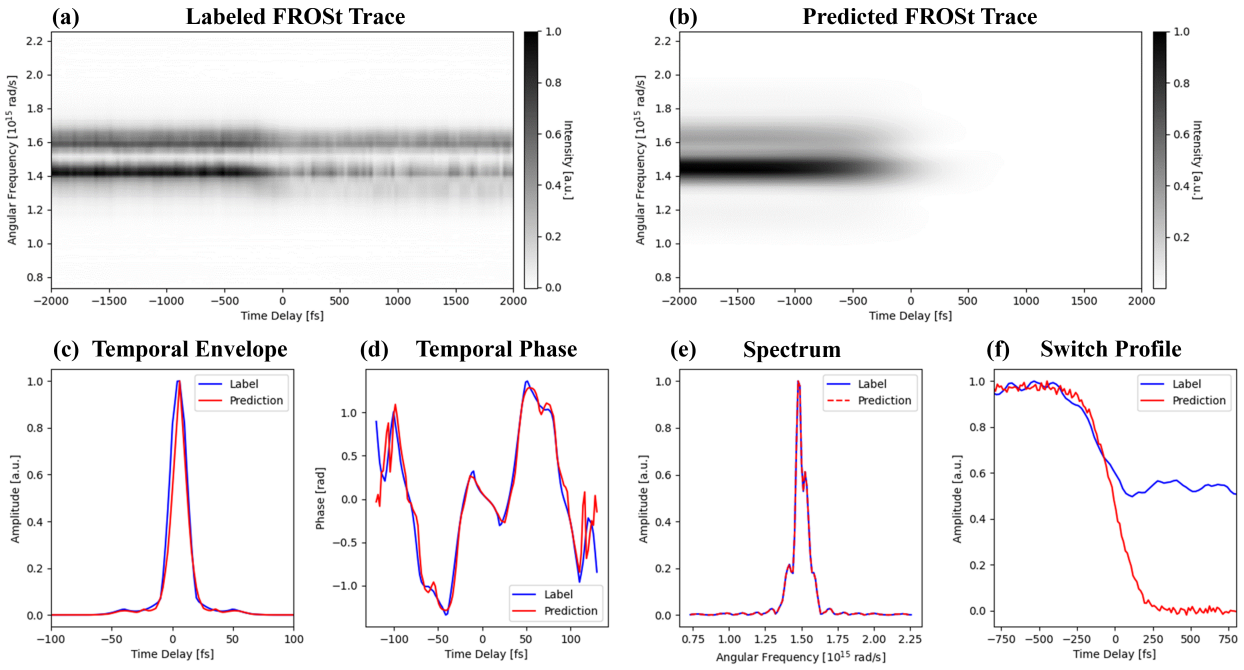

Figure 7.13 : Example 2 of Experimental Prediction by DFROStNET.

An experimental FROSt trace was taken as shown in (a). The switch was generated by a relatively low-intensity pump. The reconstructed trace, generated from the DNN predictions, is shown in (b). The labeled (in blue) and predicted (in red) vectors for the pulse envelope and phase are shown in (c) and (d), respectively. The spectrum of the labeled and predicted electric field is presented in (e). The temporal amplitude of the labeled and predicted switch is shown in (f). The reconstruction errors on the switch, pulse, and trace are 25.35%, 14.39%, and 63.42%, respectively.

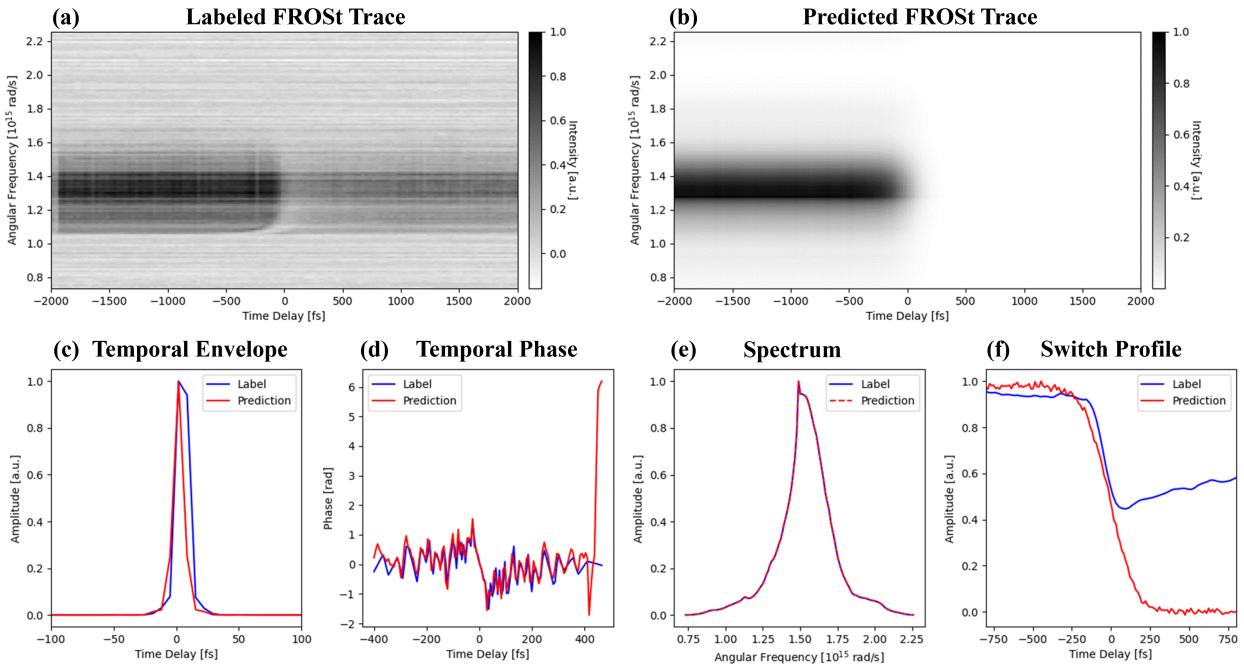

Figure 7.14 : Example 3 of Experimental Prediction by DFROStNET.

An experimental FROSt trace was taken as shown in (a). In this case, the trace was generated using a Si switch and a low-intensity pump. The reconstructed trace, generated from the DNN predictions, is shown in (b). The labeled (in blue) and predicted (in red) vectors for the pulse envelope and phase are shown in (c) and (d), respectively. The spectrum of the labeled and predicted electric field is presented in (e). The temporal amplitude of the labeled and predicted switch is shown in (f). The errors on the switch, pulse, and trace are 29.05%, 2.65%, and 56.09%, respectively.

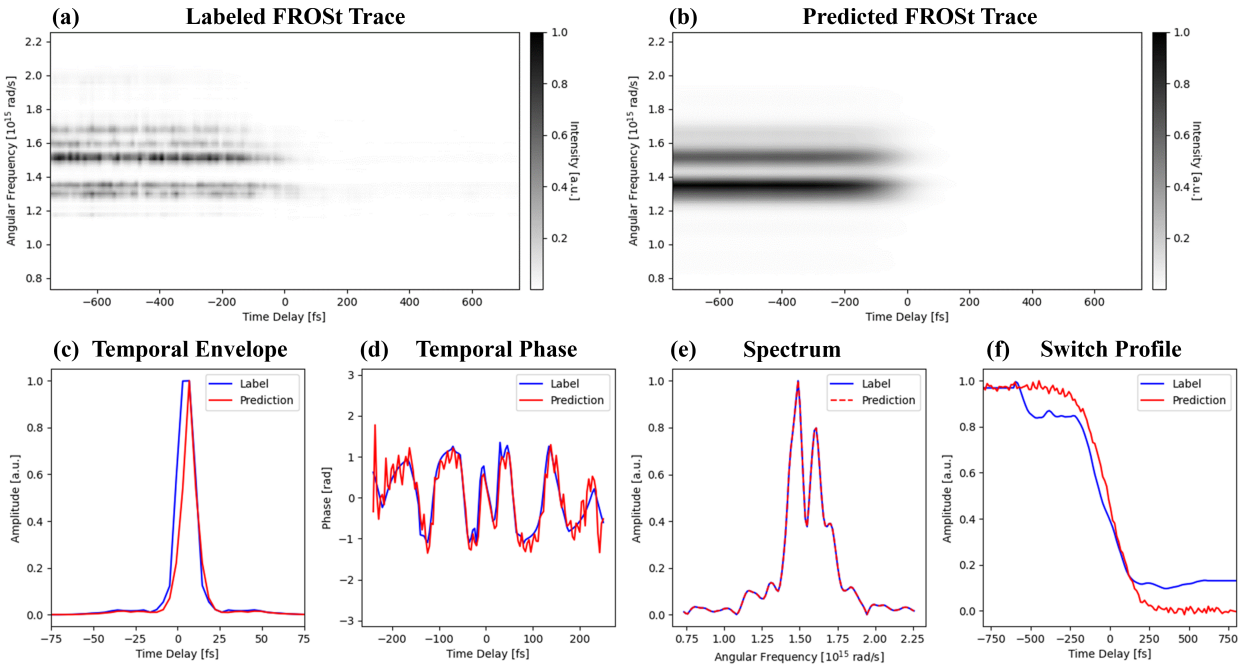

Figure 7.15 : Example 4 of Experimental Prediction by DFROStNET.

An experimental FROSt trace was taken as shown in (a). The reconstructed trace, generated from the DNN predictions, is shown in (b). The labeled (in blue) and predicted (in red) vectors for the pulse envelope and phase are shown in (c) and (d), respectively. The spectrum of the labeled and predicted electric field is presented in (e). The temporal amplitude of the labeled and predicted switch is shown in (f). The reconstruction errors on the switch, pulse, and trace are 7.44%, 6.28%, and 60.81%, respectively.

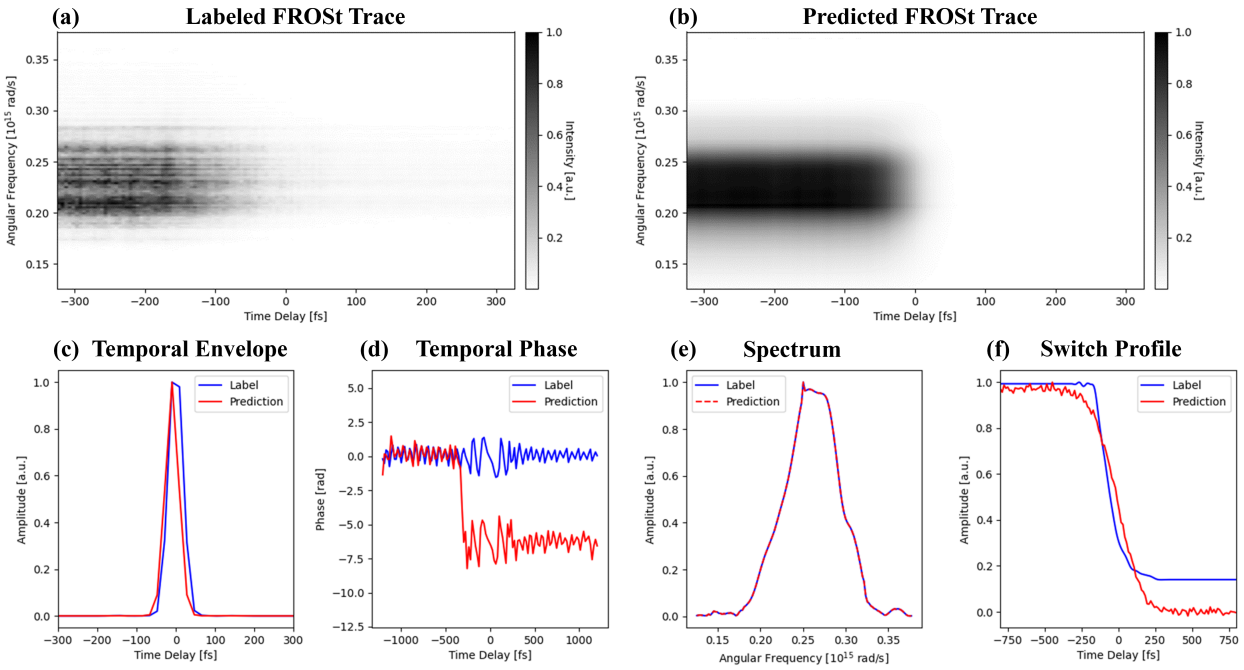

Figure 7.16 : Example 5 of Experimental Prediction by DFROStNET.

An experimental FROSt trace was taken as shown in (a). In this case, the switch was Ge at 1 mm thick. The reconstructed trace, generated from the DNN predictions, is shown in (b). The labeled (in blue) and predicted (in red) vectors for the pulse envelope and phase are shown in (c) and (d), respectively. The spectrum of the labeled and predicted electric field is presented in (e). The temporal amplitude of the labeled and predicted switch is shown in (f). The reconstruction errors on the switch, pulse and trace are 8.88%, 3.14%, and 41.67%, respectively.