IM-LPG: Inverse Modeling Approach to Laser Pulse Shape Generation in Inertial Confinement Fusion

Ricardo Luna Gutierrez[1*], **Vineet Gundecha**[1*], **Rahman Ejaz**[2*], **Varchas Gopalaswamy**[2*], **Riccardo Betti**[2*], **Sahand Ghorbanpour**[1*], **Aarne Lees**[1*], **Soumyendu Sarkar**[1*†]

[1]Hewlett Packard Enterprise [2]University of Rochester

{vineet.gundecha, rluna, sahand.ghorbanpour, soumyendu.sarkar}@hpe.com

{reja, vgopalas, betti, alee}@lle.rochester.edu

Abstract

Fusion energy remains one of the greatest scientific challenges of our time, with transformative potential for sustainable, carbon-free power. In Inertial Confinement Fusion (ICF), achieving successful implosions critically depends on the design of Laser Pulse (LP) shapes that can efficiently drive fusion targets within stringent physical constraints. Traditional LP design relies heavily on expensive simulations and manual iterative tuning, which limits scalability. We propose an Inverse Modeling Approach to Laser Pulse Shape Generation (IM-LPG) that maps target pellet parameters and desired fusion implosion outcomes directly to tailored LP shapes. IM-LPG supports both diffusion-based and autoregressive architectures, offering flexibility for diverse modeling scenarios. To balance accuracy and feasibility, we introduce a multi-objective training setup that produces LPs satisfying physical constraints while achieving <2% error on implosion outcomes. Furthermore, we incorporate constraint conditioning through inpainting and gradient-based editing, enabling fine-grained control of pulse characteristics during generation. Our framework provides a data-driven, flexible, and controllable solution to LP design in ICF, representing a step toward accelerating the path to practical fusion energy.

1 Introduction

Inertial Confinement Fusion (ICF) offers exceptional promise for clean energy but requires precise Laser Pulse Shape (LP) design, the temporal profile of laser energy that drives fuel pellet compression. Crafting effective LPs is challenging due to the intricate relationship between laser energy delivery and implosion dynamics.

LP design relies on computationally expensive simulations and manual trial-and-error exploration, often requiring weeks to identify a single high-performing LP. This forward design loop represents a significant bottleneck, necessitating an inverse design model that maps desired outcomes to promising LPs.

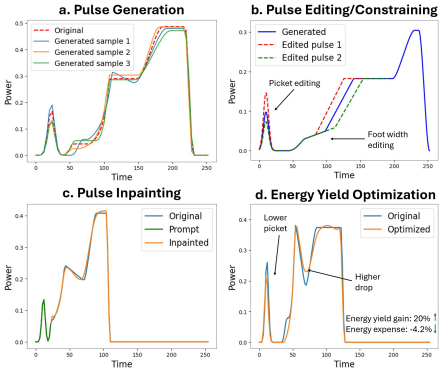


Figure 1: IM-LPG supports: (a) generation of diverse pulse shapes (b) constraint-based editing of specific pulse features, (c) inpainting specific regions (d) LP design for outcome-driven optimization.

*These authors contributed equally.

†Corresponding author.

AI for Science workshop (NeurIPS 2025).

In this work, we introduce the Inverse Modeling Approach to Laser Pulse Shape Generation (IM-LPG) a novel machine learning-based inverse modeling framework to address this critical challenge. IM-LPG enables scalable, data-driven LP design aligned with fusion objectives and experimental constraints. The main contributions of our work are:

- We investigate two inverse modeling generative paradigms for LP design: diffusion models and auto-regressive models.
- We introduce an auxiliary objective aligning generation with target implosion outcomes, yielding multiple valid LP candidates.
- We incorporate a physics-informed loss to ensure generated LPs are physically plausible and experimentally feasible.
- We develop conditioning mechanisms for direct control over specific LP attributes and regions, enabling editing, constrained generation, and inpainting.
- We demonstrate that IM-LPG can optimize LPs to improve implosion outcomes.

2 Background

2.1 ICF

ICF achieves nuclear fusion by using high-energy lasers to compress and heat fuel pellets (typically deuterium-tritium) to extreme temperatures and pressures, overcoming electrostatic repulsion between nuclei [1]. Achieving ignition requires precise, symmetric implosion controlled by the LP, the temporal profile governing compression over 3 nanoseconds.

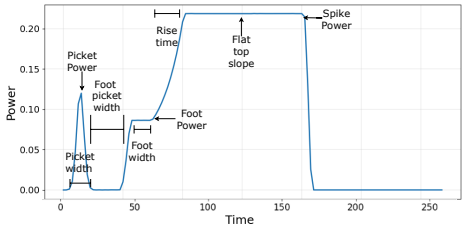


Figure 2: The LP is typically characterized by 12 parameters, some of which are annotated here.

The LP is carefully structured with distinct phases: a low-intensity foot launches timed shocks, a rising ramp controls pressure buildup, a high-intensity main pulse drives rapid implosion, and an optional tail sustains pressure (Figure 2). This multi-phase structure ensures symmetric compression and optimal fusion conditions.

Due to high experimental costs and limited access to ICF facilities, researchers rely on physics-based simulations to evaluate LP designs. LILAC is a prominent one-dimensional Lagrangian hydrodynamics ICF code [2–7], that maps LPs and target parameters to implosion outcomes (energy yield, velocity, areal density). This controlled environment enables systematic exploration of how variations in the LP influence fusion performance for a given target.

2.2 Denoising Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) [8] are generative models that use a forward diffusion process to gradually add Gaussian noise to a data sample $\mathbf{x}$ over N steps. This forward process is a Markov chain, where each step n is defined by $q(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \sqrt{1-\beta_n}\,\mathbf{x}_{n-1}, \beta_n\,\mathbf{I})$.

Denoising Diffusion Implicit Models (DDIMs) [9] accelerate sampling by replacing DDPM's stochastic reverse process with a deterministic update. The deterministic reverse update is given by $\mathbf{x}_{n-1} = \sqrt{\bar{\alpha}_{n-1}}\,\mathbf{x}' + \sqrt{1-\bar{\alpha}_{n-1}-\sigma_n^2}\,\epsilon_\theta(\mathbf{x}_n, n) + \sigma_n\epsilon_n$, where $\epsilon_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\sigma_n \in [0, 1]$ controls the stochasticity. The original input $\mathbf{x}'$ is stimated as:

$$\mathbf{x}' \approx \frac{\mathbf{x}_n - \sqrt{1-\bar{\alpha}_n}\,\epsilon_\theta(\mathbf{x}_n, n)}{\sqrt{\bar{\alpha}_n}}. \tag{1}$$

DDIM trains the noise prediction network $\epsilon_\theta(\mathbf{x}_n, n)$ following the noise prediction objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{n\sim[1,N],\,\mathbf{x}\sim q(\mathbf{x}),\,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\epsilon - \epsilon_\theta(x_n, n)\|^2\right]. \tag{2}$$

2.3 Auto-regressive Models

Auto-regressive models are a class of sequence generation models that predict each element in a sequence conditioned on all preceding elements. Given a sequence $\mathbf{x} = (x_1, x_2, ..., x_N)$, an auto-regressive model defines the probability of the sequence as a product of conditional probabilities:

$$p(\mathbf{x}) = \prod_{n=1}^{N} p(x_n | x_{1:n-1}),$$

where $x_{1:n-1}$ represents the subsequence from x_1 to x_{n-1}. Models like Long Short-Term Memory networks (LSTMs) [10] and Transformers [11] are commonly used to implement this approach. LSTMs are effective in learning dependencies over shorter sequences, while Transformers dominate modern NLP [12] and multi-modal tasks due to their scalability and capacity for long-range modeling.

3 Inverse Modeling Approach to Laser Pulse Shape Generation

We present the Inverse Modeling Approach to Laser Pulse Shape Generation (IM-LPG), an inverse modeling framework to generate LPs in ICF. IM-LPG takes desired implosion outcomes and target pellet parameters as input, aiming to generate LPs that achieve specified fusion objectives. An overview of the system architecture is provided in Figure 3. The following subsections describe each component of IM-LPG in detail.

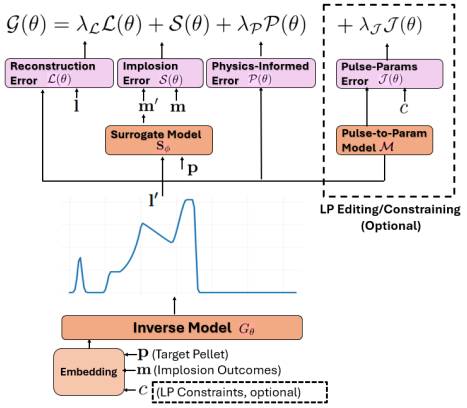


Figure 3: Training setup for the ICF IM-LPG. The loss is comprised of the reconstruction error, the implosion outcomes error, and the physics-informed error. The pulse-params error is only active when training the model for LP constrained design as described in the Pulse Shape Constrained Design section. The Surrogate and Pulse-to-Param model are frozen during training.

3.1 Inverse Modeling

Let $\mathbf{m} = \text{LILAC}(\mathbf{l}, \mathbf{p})$ denote the vector of implosion outcomes produced by the LILAC ICF simulator for a given LP ($\mathbf{l}$) and target pellet parameters ($\mathbf{p}$). These outcomes include quantities such as energy yield, areal density, burn width, and ion temperature. The pellet parameters are defined by attributes such as outer radius, ice thickness, ablator thickness, ice-tritium fraction, etc.

To construct an inverse design model, we first generate a dataset $D_F = \{(\mathbf{l}_i, \mathbf{p}_i, \mathbf{m}_i)\}_{i=1}^{I}$ of 1 Million examples via systematic simulation sweeps, capturing the relationship between pulse shapes, target parameters, and resulting implosion outcomes. The LP $\mathbf{l_i}$ is characterized as a real-valued sequence of length 256. $\mathbf{p}_i, \mathbf{m}_i$ are real-valued vectors of size 5 and 12, respectively.

Our goal is to learn a data-driven inverse mapping G_θ that designs a feasible LP $\mathbf{l}'$ given desired implosion outcomes $\mathbf{m}$ and pellet parameters $\mathbf{p}$:

$$\mathbf{l}' = G_\theta(\mathbf{m}, \mathbf{p}). \tag{3}$$

In practice, conditioning parameters such as $\mathbf{p}$ and $\mathbf{m}$ are embedded using a linear transformation when passed to the inverse model G_θ, resulting in the input form $G_\theta(\mathbf{m}', \mathbf{p}')$. For notational convenience, we use $G_\theta(\mathbf{m}, \mathbf{p})$ and $G_\theta(\mathbf{m}', \mathbf{p}')$ interchangeably throughout the paper, with the understanding that $\mathbf{m}$ and $\mathbf{p}$ implicitly refer to their embedded form unless otherwise specified.

This inverse design problem has multiple plausible solutions, since multiple pulse shapes can lead to similar outcomes. To address this, we consider two modeling approaches:

- Generative (Stochastic): Diffusion and auto-regressive generative models are trained to generate from a distribution of valid LPs, thereby providing diversity and robustness.
- Predictive (Deterministic): We also train predictive auto-regressive models to predict a single high-fidelity LP.

Diffusion Model Diffusion models sample from a learned distribution by denoising a latent representation over steps. This has fueled interest for their potential in different physics applications [13–16].

We leverage DDIM [9] to build G_θ, conditioning it on both the desired implosion outcomes $\mathbf{m}$ and the target parameters $\mathbf{p}$. During the forward diffusion process, noise is gradually added to the original LP $\mathbf{l}$ over N timesteps, resulting in a noisy latent variable $\mathbf{l}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The inverse model learns a conditional noise prediction network $\epsilon_\theta(\mathbf{l}_n, n, \mathbf{m}, \mathbf{p})$ that estimates the noise added during a forward process, conditioned on the noisy LP $\mathbf{l}_n$, the timestep n, the target implosion outcomes $\mathbf{m}$, and the target parameters $\mathbf{p}$. The training objective of the network ϵ_θ is:

$$\mathcal{L}(\theta) = \mathbb{E}_{n\sim[1,N],(\mathbf{l},\mathbf{m},\mathbf{p})\sim D_F}\left[\|\epsilon - \epsilon_\theta(\mathbf{l}_n, n, \mathbf{m}, \mathbf{p})\|^2\right] \tag{4}$$

$$\mathbf{l}_n = \sqrt{\bar{\alpha}_n}\,\mathbf{l} + \sqrt{1-\bar{\alpha}_n}\,\epsilon \tag{5}$$

During the generation phase, we start from a random noise vector $\mathbf{l}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively denoise it using the conditional noise prediction network via the DDIM update rule:

$$\mathbf{l}_{n-1} = \sqrt{\bar{\alpha}_{n-1}}\,\mathbf{l}' + \sqrt{1-\bar{\alpha}_{n-1}-\sigma_n^2}\epsilon_\theta(\mathbf{l}_n, n, \mathbf{m}, \mathbf{p}) + \sigma_n\epsilon_n, \tag{6}$$

where $\mathbf{l}'$ can be estimated as shown in Equation 1. Leveraging this formulation, IM-LPG generates diverse LPs consistent with target constraints, enabling exploration of physically meaningful designs.

3.1.1 Auto-regressive Model

Alternatively, LP design can be formulated auto-regressively, generating the pulse vector $\mathbf{l} = \{l_1, l_2, \ldots, l_T\}$ sequentially. At each timestep t, the model predicts l_t conditioned on the target parameters $\mathbf{p}$, the desired implosion outcomes $\mathbf{m}$, and the previously generated pulse values $\{l_1, \ldots, l_{t-1}\}$. This allows the model to capture temporal dependencies and auto-regressive correlations inherent in the pulse sequence. For the first timestep, since no past l_t values are available, the model instead conditions on an initialization vector constructed by applying linear transformations to both the target parameters $\mathbf{p}$ and the desired outcomes $\mathbf{m}$ (see Figure 3). This embedding serves as the initial conditioning input for the auto-regressive model.

We explore two paradigms of modeling for the auto-regressive models - predictive and generative. The predictive models try to reconstruct the ground-truth LP with high fidelity. The generative models trade-off between reconstruction fidelity and the ability to provide diverse LP candidates. The training objective for the predictive auto-regressive models is the minimization of the mean squared error between the predicted and true pulse values across all time-steps:

$$\mathcal{L}(\theta) = \frac{1}{T}\sum_{t=1}^{T}\|l_t - G_\theta(l_{<t}, \mathbf{m}, \mathbf{p})\|^2, \tag{7}$$

where $G_\theta(l_{<t}, \mathbf{m}, \mathbf{p})$ denotes the model's prediction for timestep t given the prior pulse values and conditioning inputs.

At each time step, the generative models output a probability distribution over the LP value at time t called $p_\theta(l_t)$. The training objective for the generative auto-regressive models is to minimize the negative log-likelihood of the true pulse under the predicted distribution $p_\theta(l_t)$:

$$\mathcal{L}(\theta) = \frac{1}{T}\sum_{t=1}^{T} log\; p_\theta(l_t|l_{1\ldots t}, \mathbf{m}, \mathbf{p}) \tag{8}$$

We explore 3 different modeling techniques for p_θ. The simplest technique is to model this as a Gaussian distribution where the model G_θ learns to predict the mean and variance. To offer more expressiveness, we also model p_θ as a weighted mixture of K Gaussians. Here, the model learns to predict K-length vectors for mean, variance and the mixing weights. We also try a discrete approach where the pulse l is quantized to 128 discrete values and p_θ is modeled as a Categorical distribution and trained with a cross-entropy loss.

3.2 Auxiliary Objectives

While the reconstruction objective enables LP reproduction, our broader goal is scientific exploration through plausible, diverse LP generation consistent with implosion outcomes. This provides a wider array of selectable, high-performing candidates and facilitates scientific discovery by potentially identifying novel LP configurations. To achieve this, we introduce an auxiliary objective that guides the inverse model to prioritize implosion fidelity and robustness over exact replication.

We train a surrogate model[3] $\mathbf{S}_\phi$ to approximate the LILAC simulator. This model learns the mapping from a LP $\mathbf{l}$ and target parameters $\mathbf{p}$ to the corresponding implosion outcomes $\mathbf{m}$. This can be represented as $\hat{\mathbf{m}} = \mathbf{S}_\phi(\mathbf{l}, \mathbf{p})$, where ϕ represents the learnable parameters of the surrogate model. The surrogate is trained on the simulation dataset D_F by minimizing the prediction error:

$$\text{LILAC}_{\text{surrogate}}(\phi) = \mathbb{E}_{(\mathbf{l},\mathbf{m},\mathbf{p})\sim D_F}\left[\|\mathbf{m} - \mathbf{S}_\phi(\mathbf{l}, \mathbf{p})\|^2\right]. \tag{9}$$

After training, $\mathbf{S}_\phi$ is kept frozen and is used during training of the inverse model to evaluate the physical fidelity of generated LPs. Given a generated pulse $\mathbf{l}' = G_\theta(\mathbf{m}, \mathbf{p})$, the surrogate model predicts the corresponding outcomes $\hat{\mathbf{m}} = \mathbf{S}_\phi(\mathbf{l}', \mathbf{p})$, which are then compared to the target outcomes $\mathbf{m}$. Our new objective can be defined as:

$$\mathcal{G}(\theta) = \lambda\mathcal{L}(\theta) + \mathcal{S}(\theta) \tag{10}$$

$$\mathcal{S}(\theta) = \mathbb{E}_{\mathbf{l}'\sim G_\theta,(\mathbf{m},\mathbf{p})\sim D_F}\left[\|\mathbf{m} - \mathbf{S}_\phi(\mathbf{l}', \mathbf{p})\|^2\right] \tag{11}$$

Here, λ_L controls the trade-off between faithfully reconstructing the original LP and generating physically consistent alternatives. A properly tuned λ_L allows the model to generalize beyond pure reconstruction, producing diverse yet plausible LPs tailored to the specified design goals. In practice, we first train only on the reconstruction loss $\mathcal{L}(\theta)$ since at the beginning of training G_θ can produce LPs that are incoherent and out-of-distribution for S_ϕ.

For our diffusion model, this loss formulation $\mathcal{S}(\theta)$ appears to require knowledge of the original LP $\mathbf{l}'$, at each reverse step. However, DDIM's non-Markovian formulation allows us to estimate $\mathbf{l}'$ at any intermediate timestep n (Equation 1).

3.3 Physics-Informed Loss

While our goal is to generate a diverse set of LPs for a given target configuration, the practical utility of these designs depends critically on their adherence to the fundamental physics of ICF and their feasibility under real-world experimental constraints. One essential constraint is energy conservation. Specifically, generated LPs should not exceed the total energy budget of corresponding reference pulses. For any $\mathbf{l}_i \in D_F$, we impose the following constraint on a generated LP $\mathbf{l}'_i$:

$$\int_0^T \mathbf{l}'_i(t)\,dt \le \int_0^T \mathbf{l}_i(t)\,dt, \quad t \in [0, T],$$

where T is the total pulse duration. To enforce this, we introduce a physics-based penalty term:

$$\mathcal{P}(\theta) = \mathbb{E}_{\mathbf{l}'\sim G_\theta,\mathbf{l}\sim D_F}\left[\left(\int_0^T \mathbf{l}'(t)\,dt - \int_0^T \mathbf{l}(t)\,dt\right)^+\right] \tag{12}$$

This term penalizes excess energy, encouraging designs within experimental and physical limits. Incorporating this constraint into our overall loss function, we define the complete training objective as:

$$\mathcal{G}(\theta) = \lambda_\mathcal{L}\mathcal{L}(\theta) + \lambda_\mathcal{S}\mathcal{S}(\theta) + \lambda_\mathcal{P}\mathcal{P}(\theta) \tag{13}$$

where $\lambda_\mathcal{P}$ and $\lambda_\mathcal{S}$ are hyperparameters that control the relative importance of each loss term in the overall objective. This joint objective enables the generation of LPs that are diverse, outcome-consistent, physically plausible, and experimentally viable.

[3]The surrogate achieves an error of 1.4% across the implosion outcomes $\mathbf{m}$. This can be considered as a lower bound on the implosion error that can be achieved by IM-LPG.

4 Experiments

4.1 Performance Evaluation

To evaluate our approach, we compare IM-LPG variants: Diffusion (IM-LPG$_{\text{Diff}}$), a predictive auto-regressive Transformer [11] (IM-LPG$_{\text{Transformer}}$), and an LSTM-based [10] model (IM-LPG$_{\text{LSTM}}$). The generative auto-regressive models are based on the LSTM architecture and are named IM-LPG$_{\text{GaussianAR}}$, IM-LPG$_{\text{MixtureOfGaussianAR}}$ and IM-LPG$_{\text{DiscreteAR}}$. We also include state-of-the-art generative models: Tabsyn [17] for tabular data and Variational Latent Diffusion (VLD) [16] for high-energy physics. Additionally, we evaluate ablated versions excluding the auxiliary loss $\mathcal{S}(\theta)$ ($w/o\ \mathcal{S}$) and the physics-informed loss ($w/o\ \mathcal{P}$)

We assess performance on four key metrics: implosion outcomes $\mathbf{m}$ error, reconstruction error, generation diversity, and energy conservation error. All models are evaluated over $R = 10$ random seeds on a test set $D_{F_{\text{test}}} = \{(\mathbf{l}_e, \mathbf{p}_e, \mathbf{m}_e)\}_{e=1}^{E}$, where $D_{F_{\text{test}}} \cap D_F = \emptyset$. The implosion outcomes $\mathbf{m}$ error is computed via the surrogate loss (Equation 11) and reported as mean average percentage error. Reconstruction error is: $\frac{1}{E}\sum_{e=1}^{E}\frac{1}{R}\sum_{r=1}^{R}\|\mathbf{l}_e - \mathbf{l'}_{e,r}\|$. Energy conservation is calculated in terms of percentage as: $\frac{1}{E}\sum_{e=1}^{E}\frac{1}{R}\sum_{r=1}^{R}\frac{\left(\int_0^J \mathbf{l'}_{,e,r}(j)\,dj - \int_0^J \mathbf{l}_e(j)\,dj\right)}{\int_0^J \mathbf{l}_e(j)\,dj} \times 100$. Diversity is measured by average pairwise L2 distance across generated samples: $\frac{1}{E}\sum_{e=1}^{E}\frac{1}{\binom{R}{2}}\sum_{1\le j<k\le R}\|\mathbf{l'}_{j,e} - \mathbf{l'}_{k,e}\|_2$. The estimated upper bound for diversity is 1.9, which is the diversity value obtained when comparing randomly selected LPs from $D_{F_{\text{test}}}$.

The evaluation results in Table 1 highlight the effectiveness of IM-LPG. Ablation studies confirm the critical role of both the auxiliary loss $\mathcal{S}(\theta)$ and the physics-informed loss $\mathcal{P}(\theta)$ in enhancing performance. Among the IM-LPG variants, the auto-regressive model IM-LPG$_{\text{LSTM}}$, achieves the highest reconstruction fidelity, accurately replicating original LPs. However, their deterministic nature yields zero diversity across outputs.

In contrast, generative variants (IM-LPG$_{\text{Diff}}$ and IM-LPG$_{*\text{AR}}$) balance performance and flexibility, maintaining low implosion outcome error while generating diverse, high-quality pulse shapes. Among the generative auto-regressive variants, IM-LPG$_{\text{MixtureOfGaussians}}$ offers the most diversity across samples due to better expressiveness of the multi-Gaussian distribution. This diversity provides a wider array of selectable, valid candidates and enables broader scientific discovery. The trade-off between precision and diversity allows scientists to select the model best suited to their design goals: predictive auto-regressive models for exact replication, and generative models for diverse, constraint-driven generation. Example LP generations for each model are included in Figures 5-10 of the Appendix.

Approach	**Diversity** ↑	**m Error** ↓	**Reconstruction Error** ↓	**Energy Conservation** ↓
IM-LPG$_{\text{Diff}}$	0.67	$1.95\% \pm 0.009$	0.007 ± 0.0001	$1.67\% \pm 0.005$
IM-LPG$_{\text{LSTM}}$	–	1.65%	0.0001	0.66%
IM-LPG$_{\text{Transformer}}$	–	1.94%	0.0008	0.95%
IM-LPG$_{\text{GaussianAR}}$	0.42	$1.89\% \pm 0.01$	$0.0005 \pm 2e^{-5}$	$0.58\% \pm 0.004$
IM-LPG$_{\text{MixtureOfGaussianAR}}$	0.56	$1.95\% \pm 0.09$	$0.0006 \pm 8e^{-5}$	$1.58\% \pm 0.006$
IM-LPG$_{\text{CategoricalAR}}$	0.39	$2.01\% \pm 0.04$	$0.0009 \pm 5e^{-5}$	$1.18\% \pm 0.08$
Tabsyn	0.69	$15.4\% \pm 0.2$	0.021 ± 0.0003	$5.67\% \pm 0.014$
VLD	0.39	$17.12\% \pm 0.3$	$0.016 \pm 9.7e^{-5}$	$1.88\% \pm 0.027$
IM-LPG$_{\text{Diff, w/o }\mathcal{S}}$	0.62	$4.5\% \pm 0.014$	$0.0057 \pm 8.8e^{-5}$	$1.71\% \pm 0.005$
IM-LPG$_{\text{LSTM, w/o }\mathcal{S}}$	–	3.9%	0.0004	0.0085
IM-LPG$_{\text{Transformer, w/o }\mathcal{S}}$	–	4.4%	0.001	0.0095
IM-LPG$_{\text{Diff, w/o }\mathcal{P}}$	0.69	$1.99\% \pm 0.01$	0.0084 ± 0.0001	3.21 ± 0.007
IM-LPG$_{\text{LSTM, w/o }\mathcal{P}}$	–	1.85%	0.0001	0.0078
IM-LPG$_{\text{Transformer, w/o }\mathcal{P}}$	–	2.1%	0.0009	0.0094

Table 1: ICF model performance. $\pm$ denotes standard deviation over seeds. Arrows denote desired improvement direction. For the predictive auto-regressive models (IM-LPG$_{\text{LSTM}}$, IM-LPG$_{\text{Transformer}}$), diversity is not defined since they are deterministic. The ablation experiments for the loss terms were only performed for a subset of the models due to computational constraints.

4.2 Pulse Shape Constrained Design

In many ICF experimental design scenarios, scientists require control over specific regions or attributes of the LP. A mapping function $\mathcal{M}$ (details in Appendix) is used to project an LP $\mathbf{l}$ onto $M = 12$ parameters $C = \mathcal{M}(\mathbf{l}) = \{c^1, \ldots, c^{12}\}$. Each c^m corresponds to a physically interpretable property of the LP (Figure 2). When constructing new designs, scientists fix one or two of these values within C, while allowing the remaining parameters, and thus the shape of the pulse, to vary.

However, our current inverse model G_θ lacks a mechanism to enforce such partial constraints during inference. A straightforward solution would be to take the constraint parameter $\mathbf{c}^m$ as an additional input during training, producing an LP conditioned on the desired implosion outcomes $\mathbf{m}$, target parameters $\mathbf{p}$, and the specified constraint as $\mathbf{l}' = G_\theta(\mathbf{m}, \mathbf{p}, \mathbf{c}^m)$. A constraining loss can then be introduced to enforce consistency between the specified constraint values and their corresponding values in the generated LP:

$$\mathcal{J}(\theta) = \mathbb{E}_{\mathbf{l}', c^m}\left[\|c^m_{l'} - c^m\|^2\right] \tag{14}$$

where $c^m_{l'} \in C_{l'}, C_{l'} = \mathcal{M}(\mathbf{l}')$. The constrained G_θ objective can be defined as:

$$\mathcal{G}(\theta) = \lambda_{\mathcal{L}}\mathcal{L}(\theta) + \mathcal{S}(\theta) + \lambda_{\mathcal{P}}\mathcal{P}(\theta) + \lambda_{\mathcal{J}}\mathcal{J}(\theta) \tag{15}$$

Diffusion-Based Adaptation The above approach enables G_θ to respect defined constraints. However, exhaustively training separate diffusion models for all combinations of one or two parameters in C is computationally infeasible due to combinatorial complexity.

To address this, inspired by few-shot learning in diffusion and inverse modeling [18–21], we propose a rapid adaptation strategy. Instead of retraining, we adapt a pre-trained model $G_\theta(\mathbf{m}, \mathbf{p})$ via a few gradient updates to satisfy constraint c.

Our adaptation approach employs an embedding network $\mathcal{B}_\delta$ that encodes constraint parameters $c \in C$ to condition LP generation. This results in a conditioned noise model: $\epsilon_\theta(\mathbf{l}_n, n', \mathbf{m}', \mathbf{p}', \mathcal{B}_\delta(c))$. To simplify conditioning, we enforce equal embedding dimensionality for n' and $c' = \mathcal{B}_\delta(c)$, enabling their combination into a single conditioning variable $u' = n' + c'$. This allows us to condition the model as $\epsilon_\theta(\mathbf{l}_t, u', \mathbf{m}', \mathbf{p}')$, requiring no architectural changes to the original model. During adaptation, only the parameters of $\mathcal{B}_\delta$ are fine-tuned, according to Equation 14, while the remaining weights are frozen. This strategy facilitates efficient, stable adaptation, preserving model integrity and accuracy while effectively enforcing constraints. Examples of this adaptation technique for picket power as a constraint are shown in Figure 4 (Appendix). Importantly, for a specific constraint c, the model is fine-tuned once, and this fine-tuned model can then be used to generate an arbitrary number of pulses with different specifications $\mathbf{m}$ and $\mathbf{p}$.

Gradient-Based Adaptation Alternatively, the LP can also be adapted at post-design using a model-agnostic technique without additional finetuning. To honor constraints in $\mathcal{C}$ while maintaining desired implosion outcomes, a loss combining the pulse constraining loss $\mathcal{J}(\theta)$ (Equation 14) and the LILAC surrogate loss $\mathcal{S}(\theta)$ (Equation 11) is formulated. Since both the losses are differentiable w.r.t the pulse $\mathbf{l}'$ (now referred to as $\mathcal{J}(\mathbf{l}')$, $\mathcal{S}(\mathbf{l}')$), we can perform gradient descent with respect to the following loss and update the pulse $\mathbf{l}'$:

$$\mathcal{T}(l') = \mathcal{S}(\mathbf{l}') + \lambda_{\mathcal{J}}\mathcal{J}(\mathbf{l}') \tag{16}$$

$$l' := l' - \alpha\nabla_{\mathbf{l}'}\mathcal{T}(\mathbf{l}') \tag{17}$$

where α is the learning rate. A disadvantage of this technique is that gradient descent can lead the LP into undesirable territory, such as negative values or peaks that violate the maximum power. However, in practice, we find that these issues can be effectively mitigated by clipping the pulse. Moreover, gradient-based adaptation must be performed for each unique constrained LP, whereas our Diffusion-Based Adaptation allows a fine-tuned model to be reused indefinitely.

Gradient-Based Adaptation We evaluate the ability of IM-LPG to rapidly adapt to specific design constraints c, using both Diffusion-Based and Gradient-Based adaptation strategies. For the Diffusion-Based approach, the model is fine-tuned for only 10 epochs on a new constraint parameter c^m, requiring minimal computational overhead. In parallel, we apply the Gradient-Based adaptation to the deterministic models IM-LPG$_{\text{LSTM}}$ and IM-LPG$_{\text{Transformer}}$, allowing for direct post-generation refinement.

We focus our evaluation on two physically meaningful parameters: picket power and foot power (see Figure 2). To quantify adaptation accuracy, we compute the mean absolute percentage error

(MAPE) between the specified constraint value c and the corresponding value extracted from the generated pulse (as defined in Equation 14). The Gradient-Based approach achieves a MAPE of $\mathbf{2.5\% \pm 0.12}$, while the Diffusion-Based method yields a MAPE of $\mathbf{3.4\% \pm 0.14}$. Importantly, both adaptation methods preserve IM-LPG's performance on the target implosion outcomes ($\mathbf{m}$-**Error** remains unaffected).

These constraint adaptation strategies empower scientists to generate diverse, high-quality pulse shapes that meet desired implosion outcomes and can be interactively tuned to satisfy evolving experimental or engineering constraints.

Inpainting/Prompting For additional controllability, we support inpainting-based generation, enabling scientists to specify desired LP design directly in the LP space. Rather than conditioning on the $\mathcal{C}$ constraining parameters, users can provide a partial LP such as prefixes or fixed regions. For the auto-regressive models, providing a LP prefix is akin to prompting it to complete the rest of the pulse. For diffusion, the model conditions its denoising process on specified LP segments, producing coherent, physically valid completions (details in Appendix). To evaluate auto-regressive models, a prefix comprising 10% of the pulse (this corresponds to the first peak) is provided as the prompt. For diffusion, random segments of the LP are specified, and the rest of the LP is masked. The autoregressive method achieves a reconstruction error of 0.0003 with an $\mathbf{m}$-**Error** of 1.55%, while the diffusion method yields a reconstruction error of 0.002 and an $\mathbf{m}$-**Error** of 2.0%. Examples of the LPs generated with this technique can be found in Figure 11 and Figure 12 (Appendix).

4.3 LP Optimization

A possible application of IM-LPG is energy-efficient optimization of initial laser pulse (LP) designs. Given initial parameters $\mathbf{p}_e$ and implosion outcomes $\mathbf{m}_e$, we define a new target output by increasing the energy yield by 10%, denoted as $\tilde{\mathbf{m}}_e^{\text{yield}} = 1.10 \times \mathbf{m}_e^{\text{yield}}$, where $\mathbf{m}_e^{\text{yield}}$ refers to the energy yield component. Using this new target, we generate an optimized pulse $\tilde{\mathbf{l}}'_e = G_\theta(\mathbf{p}_e, \tilde{\mathbf{m}}_e)$.

The objective is to produce a new LP $\tilde{\mathbf{l}}'_e$ that (i) achieves a predicted energy yield at least as high as $\tilde{\mathbf{m}}_e^{\text{yield}}$ and (ii) does not exceed the original total energy budget:

$$\mathbf{S}_\phi(\tilde{\mathbf{l}}'_e)^{yield} \geq \tilde{\mathbf{m}}_e^{\text{yield}} \quad \text{and} \quad \int_0^{\tilde{J}} \tilde{\mathbf{l}}'_e(j)\, dj \leq \int_0^J \mathbf{l}_e(j)\, dj.$$

Here, the surrogate model $\mathbf{S}_\phi$ is used to estimate energy yield. Because $\tilde{\mathbf{m}}_e$ may lie outside the training distribution, we include a frequency-domain similarity constraint to ensure the generated pulse remains close to the original design:

$$\frac{1}{J} \sum_{j=1}^{J} \Big| \, |\mathcal{F}(\mathbf{l}'_e)_j| - |\mathcal{F}(\tilde{\mathbf{l}}'_e)_j| \, \Big| < \tau,$$

where $\mathcal{F}$ denotes the discrete Fourier transform, J is the number of time-domain samples, τ is a similarity tolerance, and $\mathbf{l}'_e = G_\theta(\mathbf{p}_e, \mathbf{m}_e)$ is the generated LP for the original conditions. This constraint ensures that $\tilde{\mathbf{l}}'_e$ remains within a familiar region of the training distribution, improving reliability and physical plausibility.

For each initial design, we generate 20 candidate pulses (each from a different random seed) and select the one that maximizes energy gain while satisfying all constraints. We validate this approach using the **LILAC** simulator on 60 unique initial designs.

To compare design strategies, we evaluate the diffusion-based model IM-LPG$_{\text{Diff}}$ against the auto-regressive model IM-LPG$_{\text{LSTM}}$, which showed strong performance in the performance evaluation. IM-LPG$_{\text{Diff}}$ achieves a mean energy gain of **17.74%**, without increasing total energy input. In contrast, IM-LPG$_{\text{LSTM}}$ yields a modest improvement of only **2.79%**. Strikingly, IM-LPG$_{\text{Diff}}$ surpasses the 10% target, underscoring its efficiency in exploring and identifying superior LP designs for ICF optimization.

5 Conclusion

We presented IM-LPG, a data-driven inverse modeling framework designed to generate diverse ICF laser pulse shapes that meet both scientific objectives and physical constraints. The method enables

constrained pulse design as well as optimization of implosion outcomes. By providing a scalable and adaptable generative tool, IM-LPG accelerates fusion research through machine learning. This approach has the potential to substantially advance ICF studies, moving us closer to the realization of virtually limitless clean energy, a milestone with far-reaching environmental and societal benefits.

References

[1] R. Betti, O. A. Hurricane, Inertial-confinement fusion with lasers, Nature Physics 12 (2016) 435–448. URL: https://doi.org/10.1038/nphys3736. doi:10.1038/nphys3736.

[2] E. C. Hansen, D. H. Barnak, R. Betti, E. M. Campbell, P.-Y. Chang, J. R. Davies, V. Y. Glebov, J. P. Knauer, J. Peebles, S. P. Regan, A. B. Sefkow, Measuring implosion velocities in experiments and simulations of laser-driven cylindrical implosions on the omega laser, Plasma Physics and Controlled Fusion 60 (2018) 054014. URL: https://dx.doi.org/10.1088/1361-6587/aab73f. doi:10.1088/1361-6587/aab73f.

[3] LLE Review, Volume 85, Technical Report DOE/SF/19460-378, University of Rochester, Laboratory for Laser Energetics, 2000. URL: https://www.lle.rochester.edu/media/publications/lle_review/documents/v85/85_Review.pdf, quarterly Report.

[4] J. Delettrez, R. Epstein, M. C. Richardson, P. A. Jaanimagi, B. L. Henke, Effect of laser illumination nonuniformity on the analysis of time-resolved x-ray measurements in uv spherical transport experiments, Phys. Rev. A 36 (1987) 3926–3934. URL: https://link.aps.org/doi/10.1103/PhysRevA.36.3926. doi:10.1103/PhysRevA.36.3926.

[5] A. Lees, R. Betti, J. P. Knauer, V. Gopalaswamy, D. Patel, K. M. Woo, K. S. Anderson, E. M. Campbell, D. Cao, J. Carroll-Nellenback, R. Epstein, C. Forrest, V. N. Goncharov, D. R. Harding, S. X. Hu, I. V. Igumenshchev, R. T. Janezic, O. M. Mannion, P. B. Radha, S. P. Regan, A. Shvydky, R. C. Shah, W. T. Shmayda, C. Stoeckl, W. Theobald, C. Thomas, Experimentally inferred fusion yield dependencies of omega inertial confinement fusion implosions, Phys. Rev. Lett. 127 (2021) 105001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.105001. doi:10.1103/PhysRevLett.127.105001.

[6] V. Gopalaswamy, R. Betti, J. P. Knauer, N. Luciani, D. Patel, K. M. Woo, A. Bose, I. V. Igumenshchev, E. M. Campbell, K. S. Anderson, K. A. Bauer, M. J. Bonino, D. Cao, A. R. Christopherson, G. W. Collins, T. J. B. Collins, J. R. Davies, J. A. Delettrez, D. H. Edgell, R. Epstein, C. J. Forrest, D. H. Froula, V. Y. Glebov, V. N. Goncharov, D. R. Harding, S. X. Hu, D. W. Jacobs-Perkins, R. T. Janezic, J. H. Kelly, O. M. Mannion, A. Maximov, F. J. Marshall, D. T. Michel, S. Miller, S. F. B. Morse, J. Palastro, J. Peebles, P. B. Radha, S. P. Regan, S. Sampat, T. C. Sangster, A. B. Sefkow, W. Seka, R. C. Shah, W. T. Shmyada, A. Shvydky, C. Stoeckl, A. A. Solodov, W. Theobald, J. D. Zuegel, M. G. Johnson, R. D. Petrasso, C. K. Li, J. A. Frenje, Tripled yield in direct-drive laser fusion through statistical modelling, Nature 565 (2019) 581–586.

[7] V. Gopalaswamy, C. A. Williams, R. Betti, D. Patel, J. P. Knauer, A. Lees, D. Cao, E. M. Campbell, P. Farmakis, R. Ejaz, K. S. Anderson, R. Epstein, J. Carroll-Nellenbeck, I. V. Igumenshchev, J. A. Marozas, P. B. Radha, A. A. Solodov, C. A. Thomas, K. M. Woo, T. J. B. Collins, S. X. Hu, W. Scullin, D. Turnbull, V. N. Goncharov, K. Churnetski, C. J. Forrest, V. Y. Glebov, P. V. Heuer, H. McClow, R. C. Shah, C. Stoeckl, W. Theobald, D. H. Edgell, S. Ivancic, M. J. Rosenberg, S. P. Regan, D. Bredesen, C. Fella, M. Koch, R. T. Janezic, M. J. Bonino, D. R. Harding, K. A. Bauer, S. Sampat, L. J. Waxer, M. Labuzeta, S. F. B. Morse, M. Gatu-Johnson, R. D. Petrasso, J. A. Frenje, J. Murray, B. Serrato, D. Guzman, C. Shuldberg, M. Farrell, C. Deeney, Demonstration of a hydrodynamically equivalent burning plasma in direct-drive inertial confinement fusion, Nature Physics 20 (2024) 751–757.

[8] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: Conference on Neural Information Processing Systems (NeurIPS), 2020.

[9] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, in: International Conference on Learning Representations, 2021.

[10] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, volume 30, 2017.

[12] A. Radford, K. Narasimhan, Improving language understanding by generative pre-training, 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[13] C. Liu, C. Wu, shihui cao, M. Chen, J. C. Liang, A. Li, M. Huang, C. Ren, Y. N. Wu, D. Liu, T. Geng, Diff-PIC: Revolutionizing particle-in-cell nuclear fusion simulation with diffusion models, in: The Thirteenth International Conference on Learning Representations, 2025. URL: https://openreview.net/forum?id=c9z65sDx6M.

[14] F. Wu, S. Z. Li, Diffmd: a geometric diffusion model for molecular dynamics simulations, in: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'23/IAAI'23/EAAI'23, AAAI Press, 2023. URL: https://doi.org/10.1609/aaai.v37i4.25663. doi:10.1609/aaai.v37i4.25663.

[15] M. Petersen, G. Roig, R. Covino, Dynamicsdiffusion: Generating and rare event sampling of molecular dynamic trajectories using diffusion models, in: NeurIPS 2023 AI for Science Workshop, 2023. URL: https://openreview.net/forum?id=pwYCCq4xAf.

[16] A. Shmakov, K. Greif, M. J. Fenton, A. Ghosh, P. Baldi, D. Whiteson, End-to-end latent variational diffusion models for inverse problems in high energy physics, in: Conference on Neural Information Processing Systems (NeurIPS), 2023.

[17] H. Zhang, J. Zhang, Z. Shen, B. Srinivasan, X. Qin, C. Faloutsos, H. Rangwala, G. Karypis, Mixed-type tabular data synthesis with score-based diffusion in latent space, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=4Ay23yeuz0.

[18] S. Bartunov, D. Vetrov, Few-shot generative modelling with generative matching networks, in: A. Storkey, F. Perez-Cruz (Eds.), Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, volume 84 of *Proceedings of Machine Learning Research*, 2018, pp. 670–678.

[19] A. Netanyahu, Y. Du, A. Bronars, J. Pari, J. Tenenbaum, T. Shu, P. Agrawal, Few-shot task learning through inverse generative modeling, in: Conference on Neural Information Processing Systems (NeurIPS), 2024.

[20] G. Giannone, D. Nielsen, O. Winther, Few-shot diffusion models, 2022. URL: https://arxiv.org/abs/2205.15463. arXiv:2205.15463.

[21] T. Hu, J. Zhang, R. Yi, Y. Du, X. Chen, L. Liu, Y. Wang, C. Wang, Anomalydiffusion: Few-shot anomaly image generation with diffusion model, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, AAAI Press, 2024, pp. 8526–8534. doi:10.1609/aaai.v38i8.28696.

[22] G. Daras, H. Chung, C.-H. Lai, Y. Mitsufuji, J. C. Ye, P. Milanfar, A. G. Dimakis, M. Delbracio, A survey on diffusion models for inverse problems, 2024. URL: https://arxiv.org/abs/2410.00083. arXiv:2410.00083.

[23] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, J. C. Ye, Diffusion posterior sampling for general noisy inverse problems, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=OnD9zGAGT0k.

[24] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, J. Tamir, Robust compressed sensing mri with deep generative priors, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 14938–14954. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/7d6044e95a16761171b130dcb476a43e-Paper.pdf.

[25] J. Song, A. Vahdat, M. Mardani, J. Kautz, Pseudoinverse-guided diffusion models for inverse problems, in: International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=9_gsMA8MRKQ.

[26] H. Chung, J. Kim, S. Kim, J. C. Ye, Parallel diffusion models of operator and image for blind inverse problems, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.

[27] M. Mardani, J. Song, J. Kautz, A. Vahdat, A variational perspective on solving inverse problems with diffusion models, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=1YO4EE3SPB.

[28] J. Oscanoa, C. Alkan, D. Abraham, A. Nurdinova, D. Ennis, S. Vasanawala, M. Mardani, J. M. Pauly, Variational diffusion models for MRI blind inverse problems, in: NeurIPS 2023 Workshop on Deep Learning and Inverse Problems, 2023. URL: https://openreview.net/forum?id=Cei9ee2zfJ.

[29] H. Chihaoui, A. Lemkhenter, P. Favaro, Zero-shot image restoration via diffusion inversion, 2024. URL: https://openreview.net/forum?id=ZnmofqLWMQ.

[30] B. L. Trippe, J. Yim, D. Tischer, D. Baker, T. Broderick, R. Barzilay, T. S. Jaakkola, Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=6TxBxqNME1Y.

[31] B. Jing, G. Corso, J. Chang, R. Barzilay, T. S. Jaakkola, Torsional diffusion for molecular conformer generation, in: A. H. Oh, A. Agarwal, D. Belgrave, K. Cho (Eds.), Advances in Neural Information Processing Systems, 2022. URL: https://openreview.net/forum?id=w6fj2r62r_H.

[32] G. Duthé, I. Abdallah, E. Chatzi, Graph transformers for inverse physics: reconstructing flows around arbitrary 2d airfoils, 2025. URL: https://arxiv.org/abs/2501.17081. arXiv:2501.17081.

[33] D. Luo, Z. Chen, J. Carrasquilla, B. K. Clark, Autoregressive neural network for simulating open quantum systems via a probabilistic formulation, Physical Review Letters 128 (2022). URL: http://dx.doi.org/10.1103/PhysRevLett.128.090501. doi:10.1103/physrevlett.128.090501.

[34] Y. Yu, F. Zhan, R. WU, J. Pan, K. Cui, S. Lu, F. Ma, X. Xie, C. Miao, Diverse image inpainting with bidirectional and autoregressive transformers, in: Proceedings of the 29th ACM International Conference on Multimedia, MM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 69–78. URL: https://doi.org/10.1145/3474085.3475436. doi:10.1145/3474085.3475436.

[35] M. J. Buehler, Generative pretrained autoregressive transformer graph neural network applied to the analysis and discovery of novel proteins, Journal of Applied Physics 134 (2023) 084902. URL: https://doi.org/10.1063/5.0157367. doi:10.1063/5.0157367. arXiv:https://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0157367/18100893/084902_1_5.0157

[36] N. Geneva, N. Zabaras, Transformers for modeling physical systems, Neural Networks 146 (2022) 272–289. URL: https://www.sciencedirect.com/science/article/pii/S0893608021004500. doi:https://doi.org/10.1016/j.neunet.2021.11.022.

[37] D. Ray, D. V. Patel, H. Ramaswamy, A. Oberai, Efficient posterior inference & generalization in physics-based bayesian inference with conditional GANs, in: NeurIPS 2021 Workshop on Deep Learning and Inverse Problems, 2021. URL: https://openreview.net/forum?id=VC7UtQ2j0XW.

[38] V. Shah, C. Hegde, Solving linear inverse problems using gan priors: An algorithm with provable guarantees, 2018. URL: https://arxiv.org/abs/1802.08406. arXiv:1802.08406.

[39] R. Anirudh, J. J. Thiagarajan, B. Kailkhura, T. Bremer, An unsupervised approach to solving inverse problems using generative adversarial networks, 2018. URL: https://arxiv.org/abs/1805.07281. arXiv:1805.07281.

[40] A. Heyrani Nobari, W. Chen, F. Ahmed, Pcdgan: A continuous conditional diverse generative adversarial network for inverse design, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21, ACM, 2021, p. 606–616. URL: http://dx.doi.org/10.1145/3447548.3467414. doi:10.1145/3447548.3467414.

[41] N. Wang, H. Chang, D. Zhang, Deep-learning-based inverse modeling approaches: A subsurface flow example, Journal of Geophysical Research: Solid Earth 126 (2021). URL: http://dx.doi.org/10.1029/2020JB020549. doi:10.1029/2020jb020549.

[42] S. Kamyab, Z. Azimifar, R. Sabzi, P. Fieguth, Survey of deep learning methods for inverse problems, 2021. URL: https://arxiv.org/abs/2111.04731. arXiv:2111.04731.

[43] J. Adler, O. Öktem, Solving ill-posed inverse problems using iterative deep neural networks, Inverse Problems 33 (2017) 124007. URL: http://dx.doi.org/10.1088/1361-6420/aa9581. doi:10.1088/1361-6420/aa9581.

[44] S. Antholzer, M. Haltmeier, J. Schwab, Deep learning for photoacoustic tomography from sparse data, 2018. URL: https://arxiv.org/abs/1704.04587. arXiv:1704.04587.

[45] K. H. Jin, M. T. McCann, E. Froustey, M. Unser, Deep convolutional neural network for inverse problems in imaging, IEEE Transactions on Image Processing 26 (2017) 4509–4522. doi:10.1109/TIP.2017.2713099.

[46] K. Fan, Q. Wei, W. Wang, A. Chakraborty, K. Heller, Inversenet: Solving inverse problems with splitting networks, 2017. URL: https://arxiv.org/abs/1712.00202. arXiv:1712.00202.

[47] A. Mousavi, R. G. Baraniuk, Learning to invert: Signal recovery via deep convolutional networks, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 2272–2276. doi:10.1109/ICASSP.2017.7952561.

[48] K. Ishitsuka, W. Lin, Physics-informed neural network for inverse modeling of natural-state geothermal systems, Applied Energy 337 (2023) 120855. URL: https://www.sciencedirect.com/science/article/pii/S0306261923002192. doi:https://doi.org/10.1016/j.apenergy.2023.120855.

[49] H. R. Tamaddon-Jahromi, N. K. Chakshu, I. Sazonov, L. M. Evans, H. Thomas, P. Nithiarasu, Data-driven inverse modelling through neural network (deep learning) and computational heat transfer, Computer Methods in Applied Mechanics and Engineering 369 (2020) 113217. URL: https://www.sciencedirect.com/science/article/pii/S0045782520304023. doi:https://doi.org/10.1016/j.cma.2020.113217.

[50] Z. He, F. Ni, W. Wang, J. Zhang, A physics-informed deep learning method for solving direct and inverse heat conduction problems of materials, Materials Today Communications 28 (2021) 102719. URL: https://www.sciencedirect.com/science/article/pii/S235249282100711X. doi:https://doi.org/10.1016/j.mtcomm.2021.102719.

[51] S. Lu, Q. Zhou, X. Chen, Z. Song, J. Wang, Inverse design with deep generative models: next step in materials discovery, National Science Review 9 (2022) nwac111. URL: https://doi.org/10.1093/nsr/nwac111. doi:10.1093/nsr/nwac111. arXiv:https://academic.oup.com/nsr/article-pdf/9/8/nwac111/45957070/nwac111.pdf.

[52] T. Long, Y. Zhang, H. Zhang, Generative deep learning for the inverse design of materials, 2024. URL: https://arxiv.org/abs/2409.19124. arXiv:2409.19124.

[53] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, Computer Methods in Applied Mechanics and Engineering 379 (2021) 113741. URL: https://www.sciencedirect.com/science/article/pii/S0045782521000773. doi:https://doi.org/10.1016/j.cma.2021.113741.

[54] V. Puzyrev, Deep learning electromagnetic inversion with convolutional neural networks, Geophysical Journal International 218 (2019) 817–832. URL: https://doi.org/10.1093/gji/ggz204. doi:10.1093/gji/ggz204. arXiv:https://academic.oup.com/gji/article-pdf/218/2/817/28693772/ggz204.pdf.

[55] M. Araya-Polo, J. Jennings, A. Adler, T. Dahlke, Deep-learning tomography, The Leading Edge (2018).

[56] B. Moseley, T. Nissen-Meyer, A. Markham, Deep learning for fast simulation of seismic waves in complex media, Solid Earth 11 (2020) 1527–1549. URL: http://dx.doi.org/10.5194/se-11-1527-2020. doi:10.5194/se-11-1527-2020.

[57] B. Wu, D. Meng, L. Wang, N. Liu, Y. Wang, Seismic impedance inversion using fully convolutional residual network and transfer learning, IEEE Geoscience and Remote Sensing Letters 17 (2020) 2140–2144. doi:10.1109/LGRS.2019.2963106.

[58] S. Ren, A. Mahendra, O. Khatib, Y. Deng, W. J. Padilla, J. M. Malof, Inverse deep learning methods and benchmarks for artificial electromagnetic material design, Nanoscale 14 (2022) 3958–3969. URL: http://dx.doi.org/10.1039/D1NR08346E. doi:10.1039/D1NR08346E.

[59] T. Hong, S. H. Lee, Integrating physics-based models with sensor data: An inverse modeling approach, Building and Environment 154 (2019) 23–31. URL: https://www.sciencedirect.com/science/article/pii/S036013231930160X. doi:https://doi.org/10.1016/j.buildenv.2019.03.006.

[60] K. Xu, A. M. Tartakovsky, J. Burghardt, E. Darve, Inverse modeling of viscoelasticity materials using physics constrained learning, 2020. URL: https://arxiv.org/abs/2005.04384. arXiv:2005.04384.

[61] N. Linde, P. Renard, T. Mukerji, J. Caers, Geological realism in hydrogeological and geophysical inverse modeling: A review, Advances in Water Resources 86 (2015) 86–101. URL: https://www.sciencedirect.com/science/article/pii/S0309170815002262. doi:https://doi.org/10.1016/j.advwatres.2015.09.019.

[62] D. R. Ferreira, P. J. Carvalho, Deep learning for plasma tomography in nuclear fusion, in: NeurIPS 2020 Workshop on Deep Learning and Inverse Problems, 2020. URL: https://openreview.net/forum?id=fuB3vZdTh0.

[63] M. Leigh, D. Sengupta, J. A. Raine, G. Quétant, T. Golling, Faster diffusion model with improved quality for particle cloud generation, Phys. Rev. D 109 (2024) 012010. URL: https://link.aps.org/doi/10.1103/PhysRevD.109.012010. doi:10.1103/PhysRevD.109.012010.

[64] E. Öztürk, R. Akers, S. Pamela, T. M. Team, P. Peers, A. Ghosh, Inverse rendering of fusion plasmas: Inferring plasma composition from imaging systems, 2024. URL: https://arxiv.org/abs/2408.07555. arXiv:2408.07555.

[65] M. Leigh, D. Sengupta, G. Quétant, J. A. Raine, K. Zoch, T. Golling, PC-JeDi: Diffusion for particle cloud generation in high energy physics, SciPost Phys. 16 (2024) 018. doi:10.21468/SciPostPhys.16.1.018. arXiv:2303.05376.

[66] V. Mikuni, B. Nachman, M. Pettee, Fast point cloud generation with diffusion models in high energy physics, Physical Review D 108 (2023). URL: http://dx.doi.org/10.1103/PhysRevD.108.036025. doi:10.1103/physrevd.108.036025.

[67] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[68] J. W. Cooley, J. W. Tukey, An algorithm for the machine calculation of complex fourier series, Mathematics of Computation 19 (1965) 297–301. doi:10.2307/2003354.

[69] M. Bellanger, Digital processing of speech signals, IEEE Transactions on Acoustics, Speech, and Signal Processing 28 (1980) 118–119. doi:10.1109/TASSP.1980.1163362.

[70] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (1945) 80–83.

[71] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

A Related Work

A.1 Inverse Modeling

Inverse modeling is foundational in scientific discovery and engineering design. Recent advances in machine learning have significantly expanded its applicability across diverse scientific domains. Among these, diffusion models have emerged as particularly effective due to their ability to model complex, high-dimensional distributions and generate diverse, plausible solutions [22–31].

Autoregressive approaches have also gained prominence for their ability to capture long-range dependencies and generate outputs that satisfy intricate input constraints [32–36]. In parallel, Generative Adversarial Networks (GANs) have been widely applied to inverse problems, especially in imaging [37–40].

Beyond these, a range of other deep learning paradigms—including convolutional networks, recurrent architectures, and physics-informed neural networks—have also been explored for inverse modeling [41–48].

A.2 Inverse Modeling in Physical Sciences

In the physical sciences, inverse modeling enables the inference of hidden or causal system parameters from observable data and has been widely adopted across domains. In thermodynamics, it supports the estimation of material properties and optimization of thermal processes [48–50]. In materials science, inverse methods facilitate the discovery of materials with tailored properties [49, 51–53]. In geophysics, inverse modeling has been employed for seismic imaging, subsurface property estimation, and tomography [54–57]. Other fields, including chemistry, biomechanics, and hydrology, also have leveraged inverse methods to tackle complex modeling tasks [58–61].

In high-energy and plasma physics, inverse modeling has been used for reconstructing particle dynamics, inferring jet structures, and enhancing simulations [16, 62–66]. However, existing methods are often domain-specific and not applicable to ICF. No prior work addresses the unique physical constraints and data characteristics of ICF pulse design. Our work fills this gap by introducing a machine learning-based inverse modeling framework tailored for the high-dimensional, complex and constrained nature of LP design in ICF.

B Experimental Details

Computation was performed on a server with four H-100 GPUs for 360 hours at around 80% capacity. The ICF dataset consisted of 1 million ICF simulations. We used a 70%-30% train-test split for training and evaluation of all the approaches. The hyper-parameters used in our loss function $\mathcal{G}(\theta)$ are shown in Table 4. For the learning rate (LR) we use Pytorch's CosineAnnealing LR scheduler. We used Adam as optimizer. We use $\tau = 0.1$ for the implosion outcome optimization experiments in Section LP Optimization. We used a linear noise scheduler to train the IM-LPG$_{\text{Diff}}$. For the forward process we set $N = 100$, while we set $N = 25$ during generation.

The hyperparameters of the denoising network ϵ_θ are shown in Table 2 and for the auto-regressive LSTM are shown in Figure 3. The hyperparameters where selected using the optuna library [67] using the following ranges. Transformer layers $[2:4]$, Attention heads: $[4:8]$, Feed forward dimension: $[512:2048]$, Embedding layers: $[2:4]$, Embedding dimension: $[512:2048]$, Dropout: $[0.1:0.5]$, Learning rate: $[1e^{-6}, 1e^{-2}]$, Batch size: $\{64, 128, 256, 512\}$.

Transformer layers	4
Attention heads	4
Feed forward dimension	2048
Embedding layers	2
Embedding dimension	1024
Dropout	0.1
Learning rate	$5e^{-5}$
Batch size	128

Table 2: Network Hyperparameters ϵ_θ used in all diffusion models.

Num of layers	4
Number of hidden units	512
Learning rate	$1e^{-5}$
Batch size	128

Table 3: Network Hyperparameters used in the LSTM auto-regressive model.

$\lambda_{\mathcal{L}}$	0.5
$\lambda_{\mathcal{P}}$	0.25
$\lambda_{\mathcal{E}}$	0.25

Table 4: Hyperparameters used for our inverse model loss function $\mathcal{G}(\theta)$.

The code used for the Tabsyn experiments can be found in `https://github.com/amazon-science/tabsyn` and for VLD in `https://openreview.net/forum?id=v7WWesSiOu&noteId=HimuianB99`, we only employ their end-to-end architecture, but we do not use their consistency loss, since it does not apply to the ICF problem.

Optimization Filtering: For the filtering method used in 4.3 we evaluated three different methods to compare LPs $\mathbf{l'}_e$ and $\tilde{\mathbf{l}}'_e$: the discrete Fourier transform [68], MSE and cross-correlation [69]. We evaluate the approaches using the LILAC simulator in a set of 16 LPs, and the selected the approach which had least amount of false positives.

B.1 Additional Evaluation

We conducted an additional evaluation of IM-LPG, comparing it against two state-of-the-art generative baselines: VLD and TabSyn. All models were evaluated using 10 random seeds to ensure statistical robustness. To determine whether the observed performance differences are statistically significant, we use the *Wilcoxon signed-rank test* [70]. The test is applied to the **mError** obtained across the 10 seeds, using IM-LPG as the reference baseline.

Table 5 presents the mean prediction error and standard deviation (denoted by $\pm$) for each method, along with the corresponding p-values from the Wilcoxon test. The resulting p-values (0.002) indicate that the performance differences are statistically significant at conventional significance levels (e.g., $\alpha = 0.05$), suggesting that the improvements achieved by IM-LPG are unlikely to be due to chance.

B.2 LILAC surrogate

The LILAC surrogate ($\mathbf{S}_\phi$) is a MLP with the following hyper-params. It is trained with the Adam optimizer [71], with a batch size of 128, learning rate of $1e^{-5}$ for 100 epochs.

The surrogate has an average error of 1.4% across all the outputs.

B.3 Mapping Function

The mapping function ($\mathcal{M}$) is a 1 layer bidirectional LSTM with 64 hidden units. We concatenate the hidden state from both directions, before passing it through an output linear layer. It is trained with the Adam optimizer, with a batch size of 64, learning rate of $1e^{-5}$ for 250 epochs. The model achieves an average error of 1.1% across all 12 parameters in C

C Diffusion Models

Denoising Diffusion Probabilistic Models (DDPMs) [8] are generative models based on a forward diffusion process that gradually adds Gaussian noise to data $\mathbf{x}'$ over N steps. This forward process is a Markov chain where each step n depends only on the previous step $n-1$, defined by the conditional distribution:

$$q(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \sqrt{1-\beta_n}\,\mathbf{x}_{n-1}, \beta_t\,\mathbf{I}) \tag{18}$$

Approach	Prediction Error (%) ↓	p-value vs Ours
IM-LPG	$1.95\% \pm 0.009$	-
VLD	$17.12\% \pm 0.3$	0.002
TabSyn	$15.4\% \pm 0.2$	0.002

Table 5: Comparison of pulse generation methods across 10 random seeds. Statistical significance tested using Wilcoxon signed-rank test with IM-LPG as baseline method.

Num. of layers	4
Num. of hidden units	256
Activation function	ReLU

Table 6: Hyperparameters used for the LILAC surrogate $\mathbf{S}_\phi$

where $\beta_n \in (0, 1)$ is a variance schedule controlling the amount of noise added at each step. The schedule $\{\beta_n\}_{t=1}^N$ is typically either linear, quadratic, or cosine, carefully designed to smoothly increase the noise variance from a small value to near 1 across N timesteps.

We can express $\mathbf{x}_n$ given $\mathbf{x}'$ using:

$$\mathbf{x}_n = \sqrt{\bar{\alpha}_n}\,\mathbf{x}' + \sqrt{1-\bar{\alpha}_n}\,\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{19}$$

where $\alpha_n = 1 - \beta_n$ and $\bar{\alpha}_n = \prod_{i=1}^n \alpha_i$.

The reverse process, parameterized by a neural network, iteratively denoises samples from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ back to an estimate of the original data $\mathbf{x}'$ over N steps by learning to approximate the conditional distributions $p_\theta(\mathbf{x}_{n-1}|\mathbf{x}_n)$.

Denoising Diffusion Implicit Models (DDIM) [9] accelerates sampling, enabling faster generation without significantly sacrificing sample quality. DDIM defines a deterministic reverse update rule $\mathbf{x}_{n-1}$ conditioned on $\mathbf{x}_n$:

$$\mathbf{x}_{n-1} = \sqrt{\bar{\alpha}_{n-1}}\,\mathbf{x}' + \sqrt{1-\bar{\alpha}_{n-1}-\sigma_n^2}\epsilon_\theta(\mathbf{x}_n, n) + \sigma_n\epsilon_n, \tag{20}$$

where $\mathbf{x}'$ is estimated via:

$$\mathbf{x}' \approx \frac{\mathbf{x}_n - \sqrt{1-\bar{\alpha}_n}\,\epsilon_\theta(\mathbf{x}_n, n)}{\sqrt{\bar{\alpha}_n}}, \tag{21}$$

Here, ϵ_n is Gaussian noise, $\epsilon_\theta(\mathbf{x}_n, n)$ is the noise predicted by a neural network trained to estimate the noise added in the forward process, and $\sigma \in [0, 1]$ controls the stochasticity of the reverse process. Setting $\sigma = 0$ results in a fully deterministic sampling path, while increasing σ interpolates toward the stochasticity of DDPM. With DDIM's update rule, the reverse process no longer explicitly depends on the full sequence of forward diffusion steps. As a result, we can consider forward processes with fewer steps than the original N. This means that during sampling, the reverse process can be performed over a subset of the original time steps $NS = \{1, 2, \ldots, N\}$, selecting a reduced set of steps $\tilde{NS} \subseteq NS$ with $|\tilde{NS}| < N$.

This property allows DDIM to accelerate sampling by skipping intermediate steps in the diffusion chain, effectively generating high-quality samples in significantly fewer iterations than DDPM without retraining the model. The determinism of the fixed update rule ensures that even when subsampling timesteps, the resulting trajectories remain coherent.

DDIM trains the noise prediction network $\epsilon_\theta(\mathbf{x}_n, n)$ following the noise prediction objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{n\sim[1,N],\,\mathbf{x}'\sim q(\mathbf{x}'),\,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\epsilon - \epsilon_\theta(x_n, n)\|^2\right]. \tag{22}$$

Conditional Generation The generative capabilities of diffusion models can be extended to synthesize data conditioned on external information $\mathbf{y}$. The noise prediction network is then $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{y})$, taking the noisy sample $\mathbf{x}_t$, timestep t, and conditioning $\mathbf{y}$ as input. The training objective becomes the following.

$$\mathcal{L}(\theta) = \mathbb{E}_{n,\mathbf{x}',\epsilon,\mathbf{y}}\left[\|\epsilon - \epsilon_\theta(x_n, n, \mathbf{y})\|^2\right]. \tag{23}$$

During conditional generation, the learned $\epsilon_\theta(\mathbf{x}_n, n, \mathbf{y})$ guides the reverse denoising process based on $\mathbf{y}$.

D Discrete Fourier Transform for Pulse Shape Comparison

The discrete Fourier transform (DFT) is a fundamental tool in signal processing that transforms a sequence of time-domain samples into its frequency-domain representation. Given a discrete signal $x = [x_0, x_1, \ldots, x_{N-1}]$ of length N, its DFT is defined as:

$$\mathcal{F}(x)_k = X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi i k n/N}, \quad k = 0, 1, \ldots, N-1,$$

where i is the imaginary unit, and X_k represents the complex amplitude of the k-th frequency component in the signal.

The magnitude spectrum, $|X_k|$, captures the strength of the frequency components, while the phase spectrum (i.e., $\arg(X_k)$) encodes timing information. In our work, we use the magnitude spectrum to compare two LPs $\mathbf{l}'$ and $\tilde{\mathbf{l}}'$ in the frequency domain, ensuring that the generated pulses remain close to the original designs in spectral content.

To enforce this spectral similarity, we define a frequency-domain distance:

$$\Delta_{\text{freq}} = \frac{1}{N}\sum_{k=0}^{N-1} \Big|\, |\mathcal{F}(\mathbf{l}')_k| - |\mathcal{F}(\tilde{\mathbf{l}}')_k| \,\Big|,$$

and accept the pulse generated $\tilde{\mathbf{l}}'$ only if $\Delta_{\text{freq}} < \tau$, where τ is a small threshold. This regularization encourages coherence in the overall pulse structure, which is particularly important in physical systems like ICF, where abrupt or unphysical changes in the frequency domain can lead to invalid or unstable designs.

We computed the DFT using the Fast Fourier Transform (FFT) algorithm, which has a computational complexity of $\mathcal{O}(N \log N)$, allowing efficient integration into the generation loop.

E Inpainting in Pulse Space.

To enhance controllability during generation, IM-LPG supports inpainting as a mechanism for partial conditioning directly in the laser pulse space. This allows domain experts to specify only certain segments of a laser pulse, while leaving the remaining parts to be completed by the model in a physically consistent manner.

Formally, let $\mathbf{l} \in \mathbb{R}^T$ denote a laser pulse of length T, and let $\mathbf{b} \in \{0, 1\}^T$ be a binary mask that defines the constraint region. Each element $\mathbf{b}_t$ indicates whether the t-th component of the pulse is observed ($\mathbf{b}_t = 1$) or masked ($\mathbf{b}_t = 0$). Let $\mathbf{l}_{\text{obs}}$ represent the observed (non-masked) entries of the pulse.

To incorporate inpainting with Diffusion, we modify the reverse diffusion process so that known regions remain fixed at each step. Let $\hat{\mathbf{l}}_n$ be the predicted intermediate pulse at timestep n from the standard DDIM update rule (Equation 6). We define the inpainted latent state as:

$$\mathbf{l}_n^{\text{inpaint}} = \mathbf{b} \odot \mathbf{l}_{\text{obs}} + (1 - \mathbf{b}) \odot \hat{\mathbf{l}}_n, \tag{24}$$

where $\odot$ denotes elementwise multiplication. This formulation ensures that the observed parts of the pulse are preserved throughout the denoising process, while the model only modifies the masked (unknown) components.

In practice, the mask $\mathbf{b}$ can be flexibly defined to support a wide range of expert interactions, for example, preserving a fixed peak intensity, pre-specified rising edge, or constraints learned from past experiments. Combined with the implosion outcomes and target parameters conditioning via $\mathbf{m}$ and $\mathbf{p}$, inpainting enables IM-LPG to support fine-grained, interpretable, and editable pulse design.

F Additional Results

F.1 LP Constraining

Table 7 shows the error before and after the quick adaptation approach for LP constraining. We can see that our method is able to significantly reduce the error in all of the C parameters.

	c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}	c^{11}
Before DiffAdapt	0.002	0.009	0.0121	0.0176	0.0075	0.0166	0.0358	0.0851	0.0768	0.0025	0.0943
After DiffAdapt	0.0002	0.0005	0.0018	0.0037	0.0033	0.0027	0.0038	0.0016	0.0009	0.0012	0.0031

Table 7: Diffusion-Based adaptation for LP constraining results. MSE Before and After fine tuning for different C parameters which constraint different parts of the generated LP.

F.2 Implosion outcome optimization

In addition to the energy yield. We also tested IM-LPG$_{\text{Diff}}$ capabilities to increase other implosion outcomes following the approach described in 4.3. We ran this evaluation using surrogate $\mathcal{S}_\phi$. Table 8 presents the average increase achieved on the different implosion outcomes. Here each index m for $\mathbf{m}^m$ represents a different implosion outcome.

	$\mathbf{m}^2$	$\mathbf{m}^3$	$\mathbf{m}^4$	$\mathbf{m}^5$	$\mathbf{m}^6$	$\mathbf{m}^7$	$\mathbf{m}^8$
Percentage Improvement	24.1%	10.09%	11.78%	15.89%	6.32%	24.5%	7.57%

Table 8: IM-LPG$_{\text{Diff}}$ optimization percentage improvement on different implosion outcomes

G Example Pulse Shape Generation

We present some of the pulses generated by the different approaches. Figure 5 for IM-LPG$_{\text{Diff}}$, Figure 6a for IM-LPG$_{\text{LSTM}}$, Figure 6b for IM-LPG$_{\text{Transformer}}$, Figure 7a for IM-LPG$_{\text{GaussianAR}}$, Figure 7b for IM-LPG$_{\text{MixtureOfGaussianAR}}$

Figure 8b for Tabsyn, and Figure 8a for VLD. WE can observe that both Tabsyn and VLD fail to design meaningful LPs.

Figure 9a and 9b show LPs generated by IM-LPG$_{\text{Diff}}$ when asked to increase energy yield generation and areal density by 10%. We can observe that IM-LPG$_{\text{Diff}}$ was able to generate novel LPs, while maintaining energy expenditure equal or lower than the original design. All the pulses presented were validated on the real LILAC simulator. We got an average of 21% energy yield gain compared to the original design.

Figure 13 shows LPs generated by IM-LPG$_{\text{Diff}}$ after being finetuned using the real experimental data.

G.1 Pulse Shape Constraining

Figure 4 shows examples before and after LPs constraining using the approach presented in Section 4.2. Here, we aim to constrain on picket power. Figure 10a show examples of LPs that have been constrained on picket power, and Figure 10b LPs that are constrained on foot power.

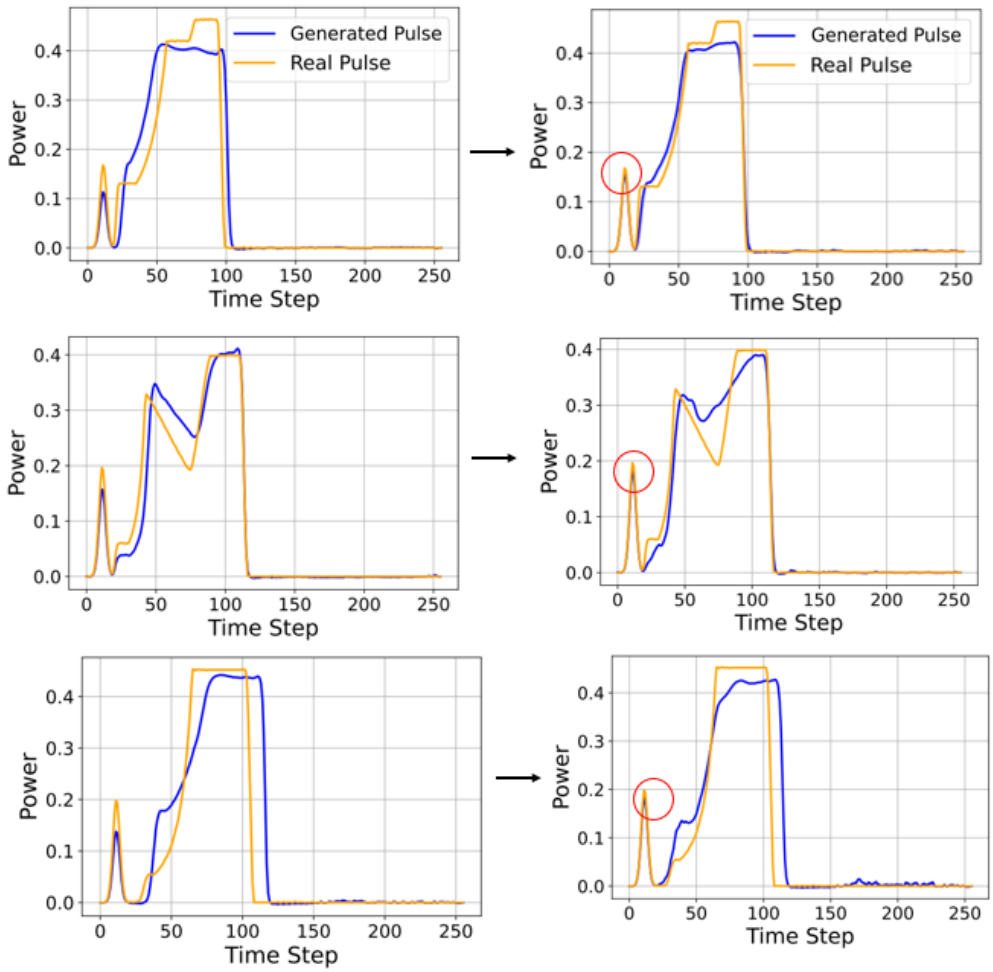


Figure 4: LPs constraining on Picket Power (red highlight). The model successfully constrains the picket power to the target value while allowing variation in the rest of the LP.

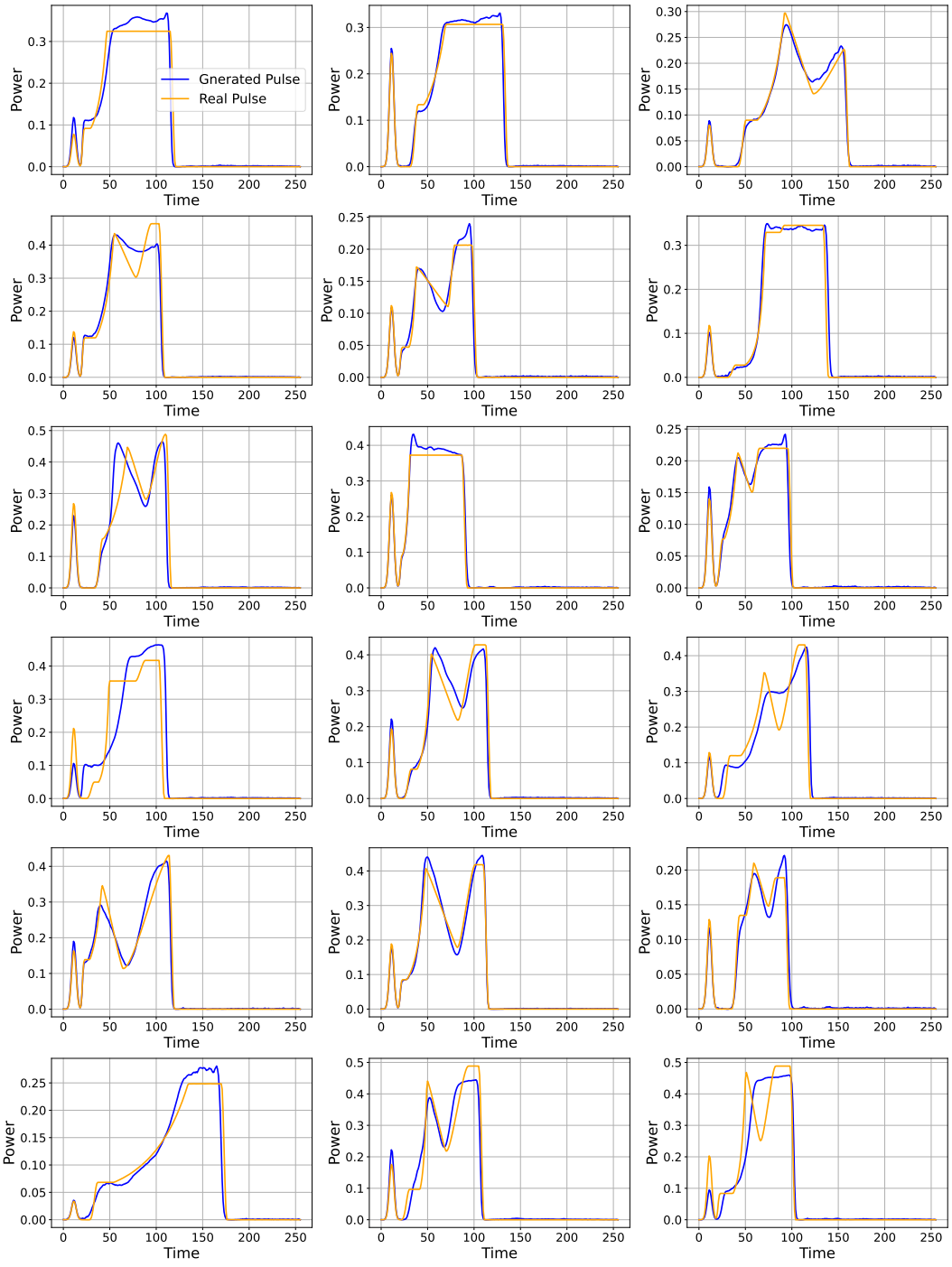
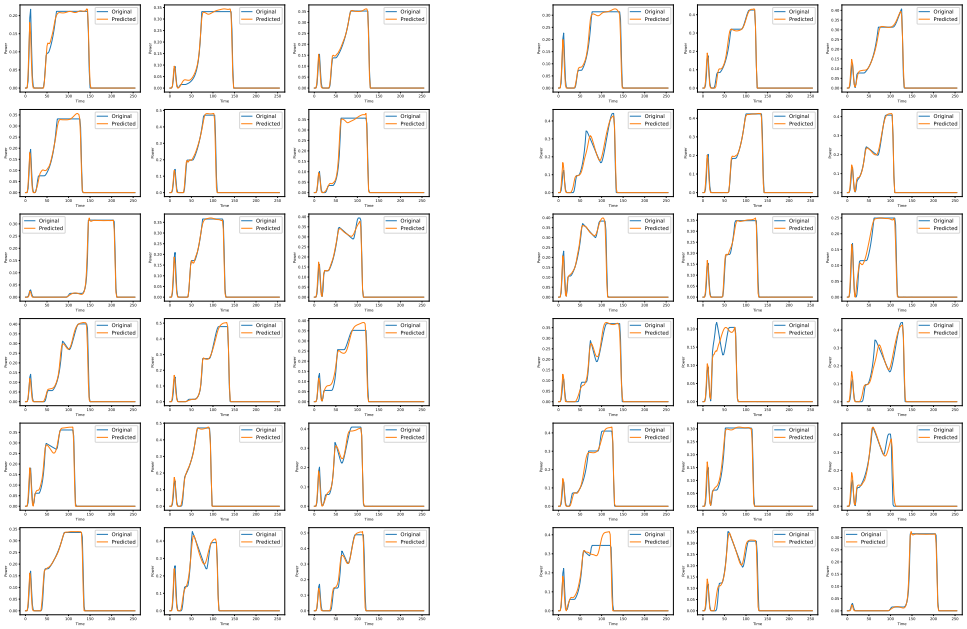


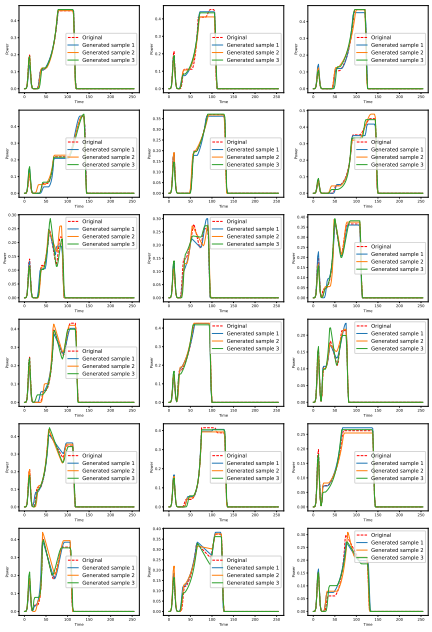
Figure 5: Pulses Generated by IM-LPG$_{\text{Diff}}$.



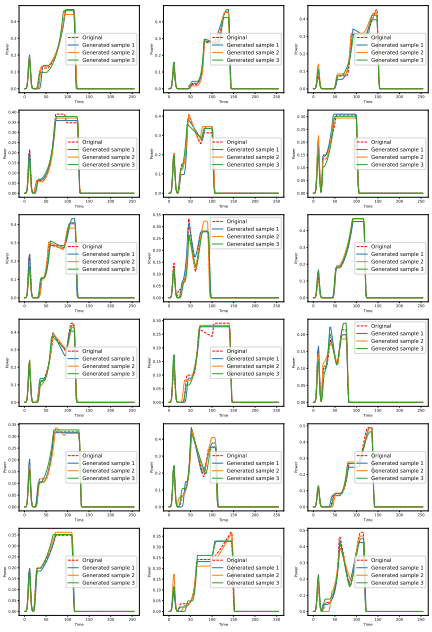
(a) LPs generated by IM-LPG$_{\text{LSTM}}$ model

(b) LPs generated by the IM-LPG$_{\text{Transformer}}$ model

Figure 6: LPs generated by predictive predictive auto-regressive models

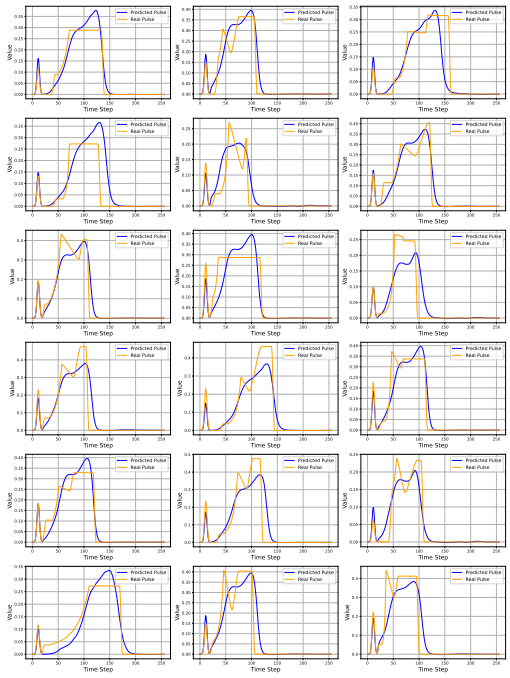


(a) LPs generated by IM-LPG$_{\text{GaussianAR}}$ model

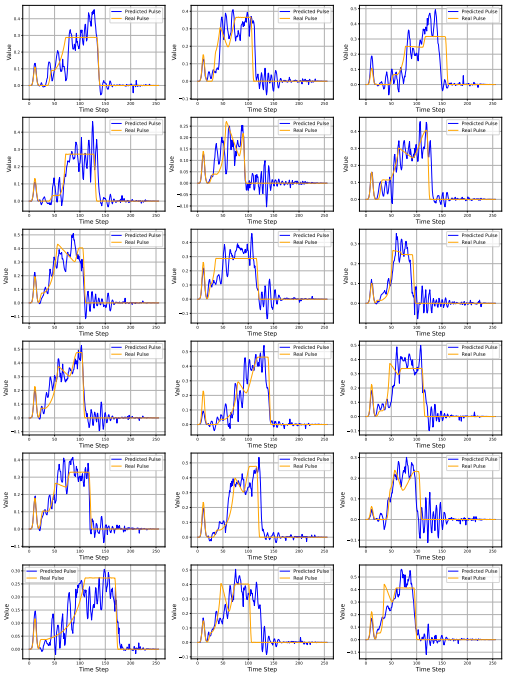


(b) LPs generated by the IM-LPG$_{\text{MixtureofGaussianAR}}$ model

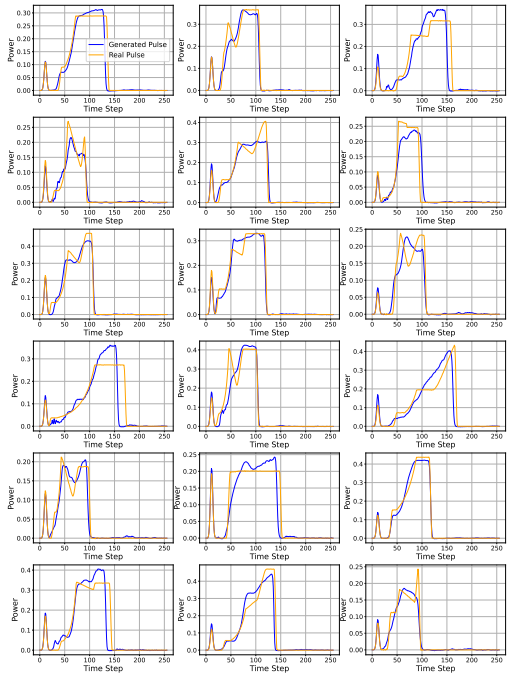
Figure 7: LPs generated by generative auto-regressive model



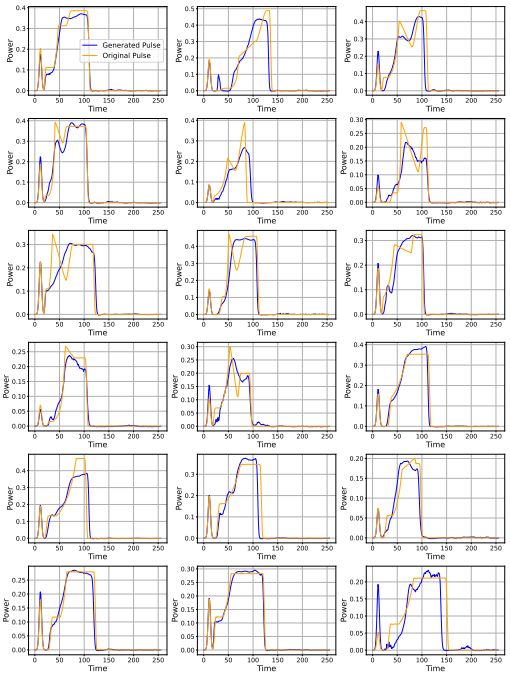
(a) LPs Generated by VLD.



(b) LPs Generated by Tabsyn.

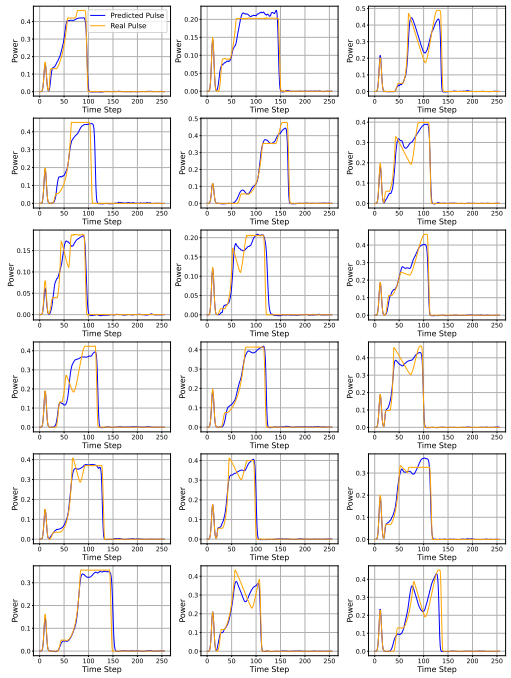


(a) LPs Generated by IM-LPG$_{\text{Diff}}$ after optimizing for enhancing energy yield.

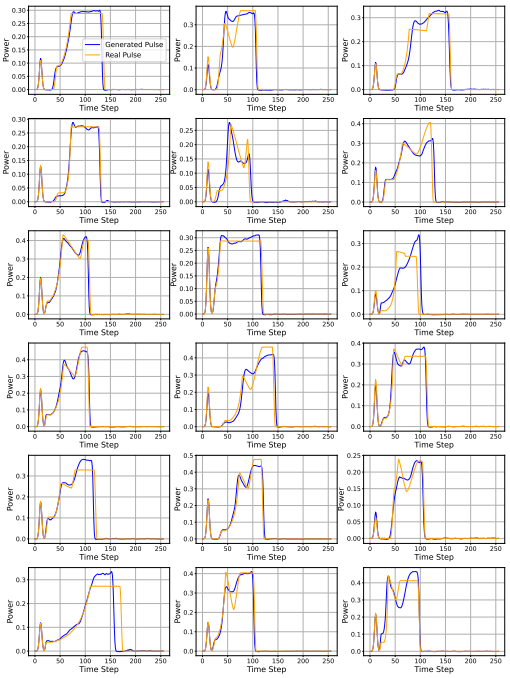


(b) LPs Generated by IM-LPG$_{\text{Diff}}$ after optimizing for enhancing areal density.

Figure 9: LPs generate by IM-LPG$_{\text{Diff}}$ implosion outcome optimization.



(a) LPs constraining on Picket Power.



(b) LPs constraining on Foot Power.

Figure 10: LPs generate by IM-LPG$_{\text{Diff}}$ when doing parameter constraining.

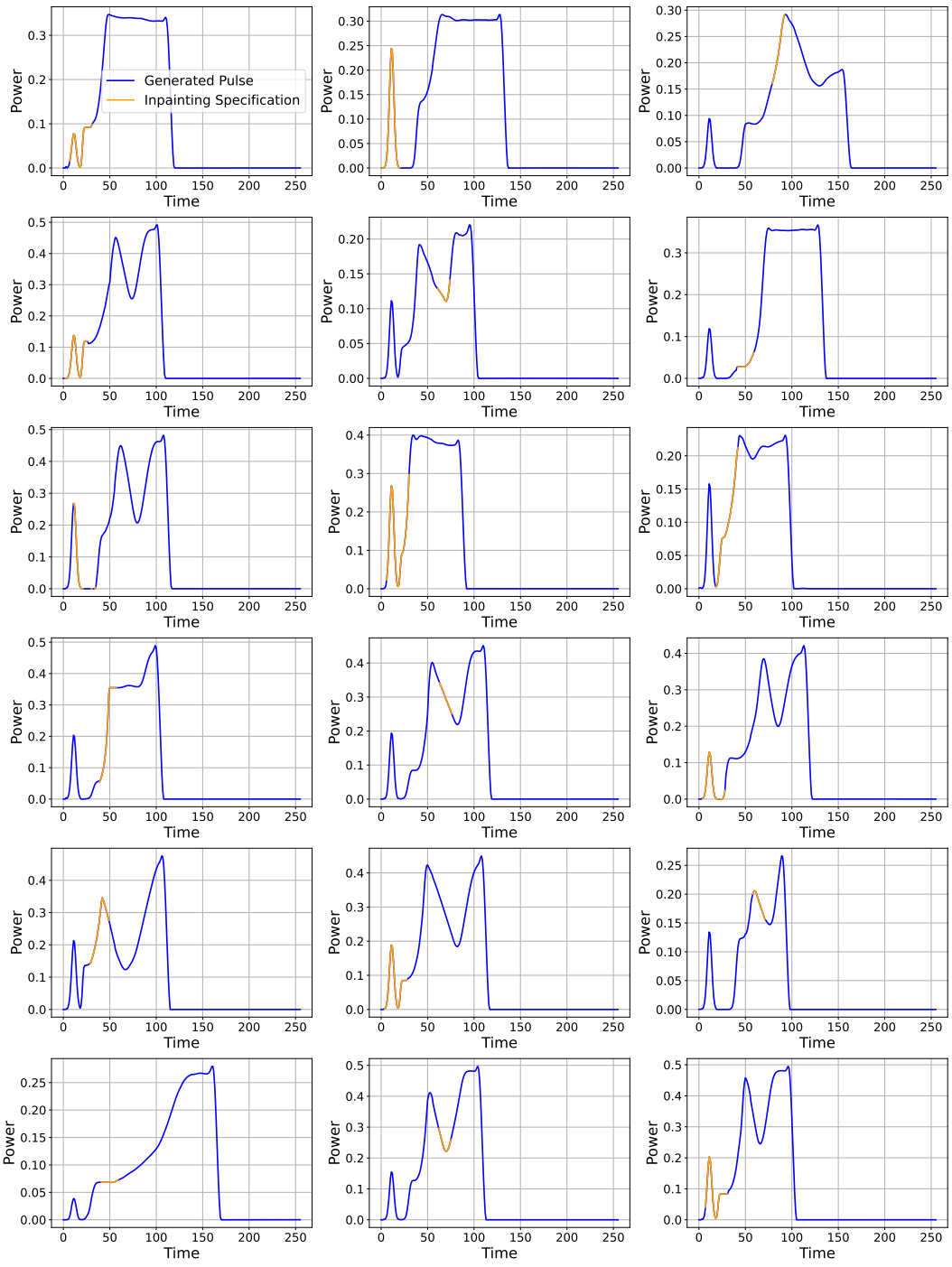


Figure 11: LPs generated by the model when doing inpainting using IM-LPG$_{\text{Diff}}$.

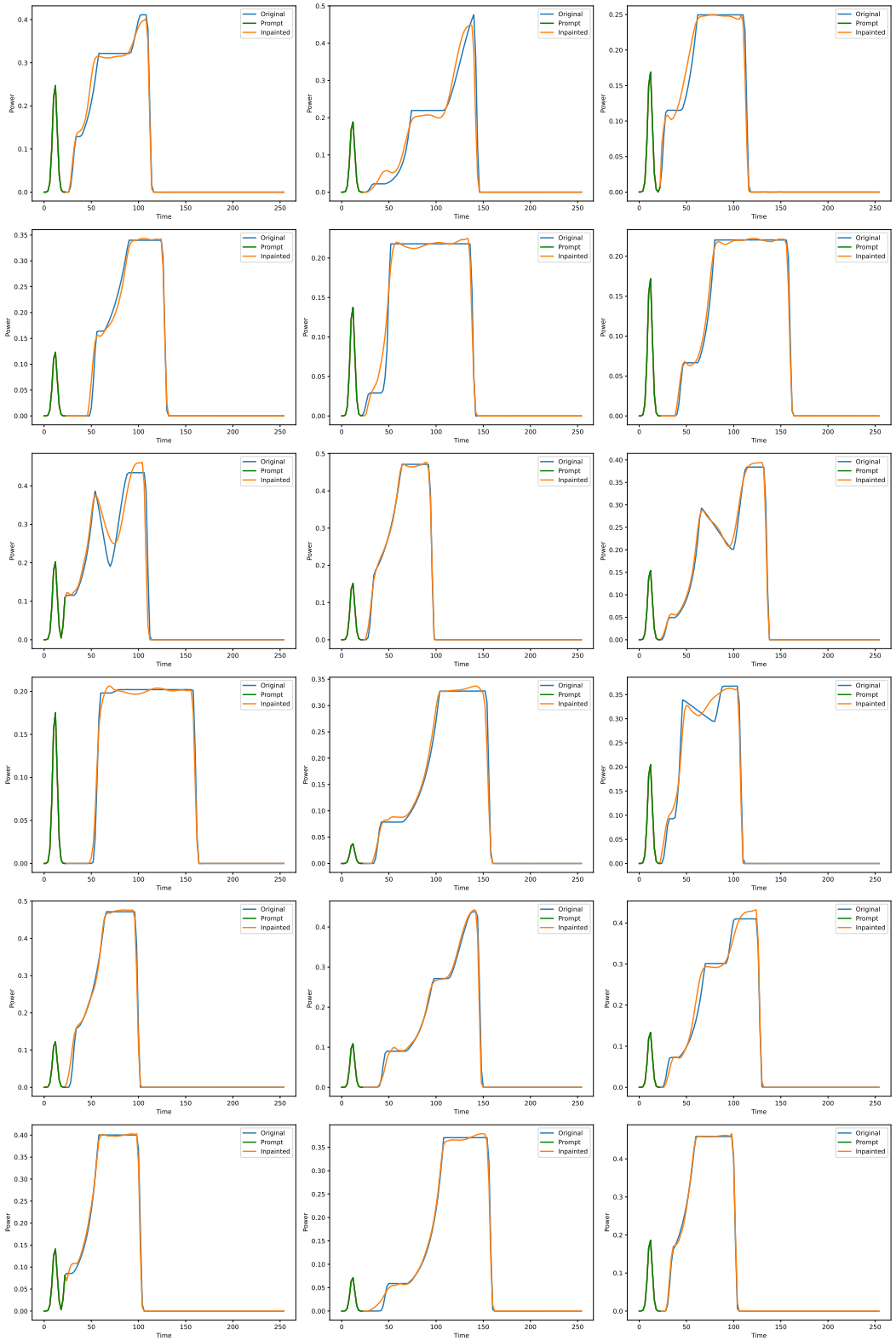


Figure 12: LPs generated by the model when doing inpainting using IM-LPG$_{\text{LSTM}}$.

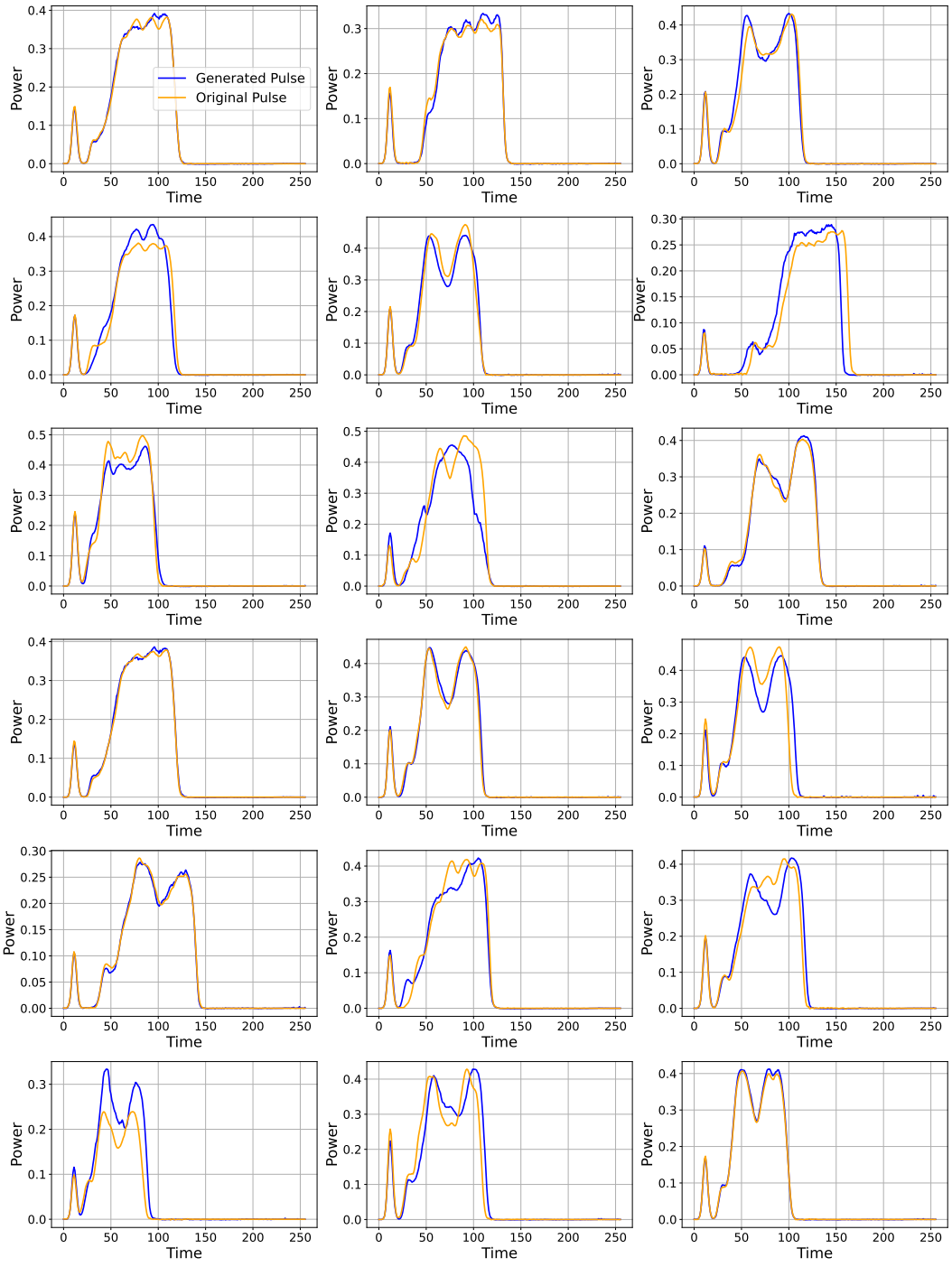


Figure 13: LPs generated by the model when finetuned in the real experimental data.