CONSTRAINT-AWARE FLOW MATCHING: DECISION ALIGNED END-TO-END TRAINING FOR CONSTRAINED SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep generative models provide state-of-the-art performance across a wide array of applications, with recent studies showing increasing applicability for science and engineering. Despite a growing corpus of literature focused on the integration of physics-based constraints into the generation process, existing approaches fail to enforce strict constraint satisfaction while maintaining sample quality. In particular, training-free constrained sampling methods, while providing per-sample feasibility guarantees, introduce a fundamental mismatch between the training objective and the constrained sampling procedure, often leading to performance degradation. Identifying this training-sampling misalignment as a central limitation of current constrained generative modeling approaches, this paper proposes *Constraint-Aware Flow Matching*, a novel end-to-end framework that explicitly incorporates constraint projections into the training objective. By aligning the model's learned dynamics with the constrained sampling process, the proposed method mitigates distributional shift induced by projection-based corrections, enabling high-quality constrained generation. The proposed approach is evaluated on three challenging real-world benchmarks, illustrating the generality and efficacy of the method.

1 INTRODUCTION

Flow matching and diffusion generative models provide state-of-the-art performance across a wide array of settings, representing the forefront of content creation for image and video generation (Rombach et al., 2022; Ifriqi et al., 2025; Jin et al., 2024), engineering (Wang et al., 2023), material science (Cui et al., 2026; Christopher et al., 2025a; Wu et al., 2026), and other scientific applications (Christopher et al., 2025b; Anand & Achim, 2022; Hoogeboom et al., 2022). Despite the promising potential of these models, the stochastic nature of these approaches often results in generations which resemble the training distribution but lack integral requirements of real-world data. Particularly within scientific domains, where samples are required to precisely satisfy domain-specific constraints and physical laws to maintain any real meaning, a critical barrier to widespread adoption is the inability of generative models to adhere to these requirements.

In response to these challenges, recent literature has started exploring constrained generative modeling for scientific domains. This work can be partitioned into three classes: (1) *Constraint Guidance* approaches condition the generation on the constraint set, but these methods provide no formal certificate of feasibility, thus relying heavily on rejection sampling (Carvalho et al., 2023; Mazé & Ahmed, 2023; Botteghi et al., 2023); (2) *Physics-Informed Training* augments generative model training with a physical residual loss, aligning the sampling distribution with the constraint set but failing to enforce per-sample satisfaction (Warner et al., 2026; Bastek et al., 2024); (3) *Constrained Sampling* provides per-sample guarantees for constraint satisfaction, and, thus, these approaches have been most widely adopted (Christopher et al., 2024; Liang et al., 2025a; Utkarsh et al., 2025; Zampini et al., 2025; Cheng et al., 2024). However, while constrained sampling shows significant promise, generation quality often degrades as compared to the unconstrained sampling approaches.

This paper argues that this performance trade-off stems from the misalignment between the training objective and the constrained sampling process. Existing constrained sampling approaches are training-free, modifying only the sampling phase and leveraging pretrained flow matching or diffusion

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

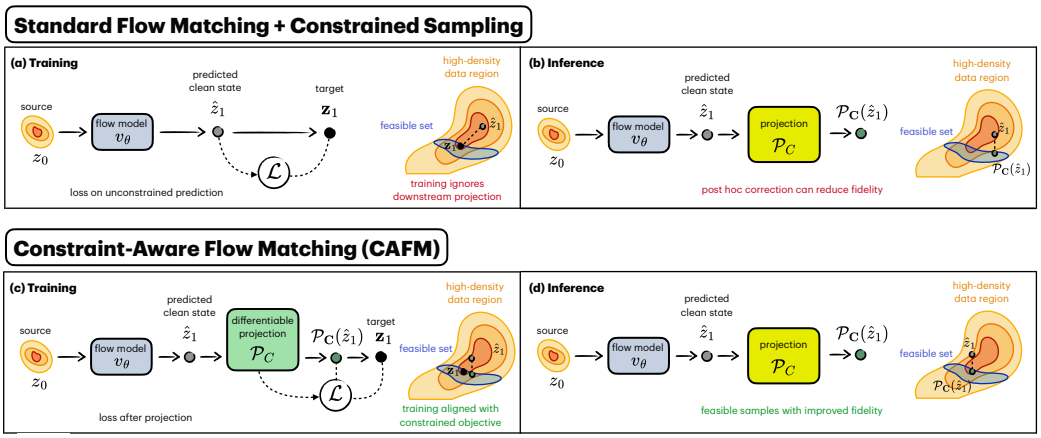


Figure 1: Visualization of **Constraint-Aware Flow Matching** compared to standard flow matching. While the clean state prediction from standard flow matching, $\hat{z}_1$, falls in a high-density region of the distribution, the projection degrades fidelity. Conversely, our constraint-aware objective optimizes the *downstream task*, learning to predict $\hat{z}_1$ such that the projection falls in high-density regions.

models. While this makes the integration of constraints inexpensive, requiring no model retraining, it also results in a model which is trained to sample in a fundamentally different way from how it is used at deployment. Generally, state-of-the-art methods rely on the integration of projections onto the feasible set during sampling, pushing the samples towards low density regions which may not have been learned during the training process (Christopher et al., 2024; Liang et al., 2025a; Utkarsh et al., 2025). Figure 1 (b) provides an illustration of this: while the initial predicted sample falls in a high-density region of the data distribution, the projection pushes the sample to lower density regions when restoring feasibility. The paper addresses this gap by introducing a constraint-aware training procedure, ensuring alignment between training and sampling.

Contributions. This work addresses the disconnect between the training and sampling processes of constrained flow matching approaches by making the following contributions: **(1)** It presents a novel end-to-end formulation of the constrained generation task, analyzing the failure modes of training-free constrained modeling (visualized in Figure 1). **(2)** It derives a constraint-aware training objective, inspired by the recent progress in the areas of differentiable optimization and decision-focused learning (Wilder et al., 2019; Wang et al., 2019; Ferber et al., 2020b; Elmachtoub & Grigas, 2022; Agrawal et al., 2019; Berthet et al., 2020; Blondel et al., 2020a; Ferber et al., 2023; Blondel et al., 2020b; Mandi et al., 2024a), aligning the training process with the downstream generative task and providing the first end-to-end training approach for constrained sampling approaches. **(3)** It presents empirical evaluation on three real-world scientific settings encompassing PDE-constrained generation, microweather forecasting, and microstructure generation, with the introduced *Constraint-Aware Flow Matching (CAFM)* reporting state-of-the-art performance across these domains.

2 Related Work

Constraint Guidance and Physics-Informed Training. Early applications of generative models for constrained domains sought to control sample dynamics using model conditioning approaches, relying on learning-based approaches which either trained an independent classifier or leveraged the diffusion model directly to steer the generation (Ho et al., 2020; Ho & Salimans, 2022). As these methods were increasingly applied for scientific tasks (Wang et al., 2023; Carvalho et al., 2023; Mazé & Ahmed, 2023), more formal notions of constraint-aware training began to be explored within the literature, applying principles from physics-informed neural networks to generative contexts (Raissi et al., 2019; Raissi & Karniadakis, 2018). Bastek et al. (2024) proposed incorporating a physical constraint directly into the training objective of the diffusion model. Warner et al. (2026) demonstrated that physical and statistical constraints could be adopted as a residual loss when learning a latent space for latent flow matching (Dao et al., 2023). However, while these works provide a stronger alternative to conditioning-based constraint enforcement, they fail to provide per-sample

guarantees, align the sampling process only at a distributional-level, and rely heavily on neural networks to approximate the true constraints.

Constrained Sampling. More recently, Christopher et al. (2024) introduced constrained sampling approaches for diffusion models, incorporating constraint correction during the generation process by projecting onto the feasible set. Subsequent studies have demonstrated that constrained sampling provides state-of-the-art performance in a broad range of domains including robotics (Liang et al., 2025a), biology (Christopher et al., 2025b), and material science (Cui et al., 2026). Sampling-time approaches have been extended to latent diffusion models (Zampini et al., 2025), as well as to flow matching parameterizations (Utkarsh et al., 2025; Liang et al., 2025b). However, while this class of methods contribute the strongest performance for constrained generative modeling tasks, they remain training-free by design, and this disconnect between training and sampling often results in degraded sample quality by conventional metrics.

Differentiable Optimization. Our method builds on the general principle that optimization procedures can be embedded within differentiable computational pipelines. Differentiation through constrained optimization problems was identified as of significant importance for machine learning settings, dating back to Gould et al. (2016). This line of research has been predominantly applied in end-to-end learning settings, enabling machine learning models to integrate constraint programming into their training processes. Amos & Kolter (2017) proposed using a quadratic programming layer for solving constrained optimization problems during the forward pass of the network. While this approach leveraged implicit differentiation of KKT conditions for the backward pass, differentiable optimization layers have been widely generalized (Agrawal et al., 2019; Kotary et al., 2023; Wilder et al., 2019; Mandi & Guns, 2020; Ferber et al., 2020a; Pogančić et al., 2019; Sahoo et al., 2022; Niepert et al., 2021). While many of these methods were limited to convex constraint sets, Kotary et al. (2023) and Blondel et al. (2022) demonstrated applicability for nonconvex constraints using sequential quadratic programming and fixed-point differentiation, enabling greater generality of applicable use cases.

3 Preliminaries: Flow Matching

Flow Matching (Lipman et al., 2022). Flow matching is a generative modeling framework where a neural network learns a time-dependent velocity field $v_\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ that defines a deterministic flow $\psi_t : \mathbb{R}^d \to \mathbb{R}^d$ transporting samples from a base distribution p_0 to a target distribution p_1. Given an initial condition $z_0 \sim p_0$, the generative trajectory is obtained by solving the ODE

$$\frac{d}{dt}\psi_t(z_0) \;=\; v_\theta\big(\psi_t(z_0), t\big), \qquad \psi_0(z_0) = z_0, \tag{1}$$

and the generated sample is $z_1 = \psi_1(z_0)$. Training in flow matching is performed using a *reference transport path* constructed from paired endpoints $(\mathbf{z}_0, \mathbf{z}_1)$, where $\mathbf{z}_0 \sim p_0$ and $\mathbf{z}_1 \sim p_1$ are coupled (e.g., via an optimal transport plan). A time-t point on this reference path is denoted $\mathbf{z}_t$ (e.g., $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\mathbf{z}_1$, optionally with small noise), and the corresponding reference velocity for the path at time t is $u_t(\mathbf{z}_t|\mathbf{z}_0, \mathbf{z}_1)$. The velocity network v_θ is then trained with this reference velocity:

$$\mathcal{L}_{FM} = \| \overbrace{v_\theta(\mathbf{z}_t, t)}^{\text{predicted}} - \overbrace{u_t(\mathbf{z}_t|\mathbf{z}_0, \mathbf{z}_1)}^{\text{target velocity}} \|^2. \tag{2}$$

Notation. Bold variables $\mathbf{z}_0, \mathbf{z}_t, \mathbf{z}_1$ denote states induced by the *training* coupling/path (constructed directly from paired endpoints), whereas non-bold variables z_0, z_t, z_1 denote states induced by the *learned generative flow* in (1). This distinction is important because training evaluates v_θ on $\mathbf{z}_t$ sampled from the reference path, while at generation time the model is queried on z_t produced by integrating the learned dynamics.

4 Constraint-Aware Sampling

The goal of constrained generation is to produce samples $z_1 \sim p_{\text{data}}$ subject to a set of feasibility requirements C. While often times the true data distribution p_{data} may already satisfy the relevant constraints by construction (e.g., physical systems typically respect the underlying governing laws),

the practical challenge is that a learned generative model v_θ, despite approximating p_{data}, may assign probability mass to infeasible regions of the ambient space. If $\mathbb{1}(\cdot)$ denotes an indicator function, the *idealized constrained target* is:

$$p_C(z_1) \propto p_{\text{data}}(z_1)\mathbb{1}\{z_1 \in C\}, \tag{3}$$

with the practical objective obtained by replacing p_{data} with the learned approximation obtained by v_θ (Christopher et al., 2025b). Across applications, the constraint set C is selected on the basis of domain-specific criteria, but, to ensure generality in the presentation, it can be arbitrarily defined as the intersection of a series of equality constraints $\mathbf{h}(z_1) = 0$ and inequality constraints $\mathbf{g}(z_1) \leq 0$, imposed on the ambient space.

As discussed in Section 2, constrained sampling approaches provide state-of-the-art performance for this class of generation task. However, these methods suffer from their own set of challenges. Seminal works focus on correcting intermediate or partially denoised states to allow constraint enforcement to guide the generation process (Christopher et al., 2024; Liang et al., 2025a). Yet, enforcing constraints on noisy representations can bias the sampling trajectory, distort the learned dynamics, or over-correct due to high-variance noise (Blanke et al., 2025). More recent studies have demonstrated that enforcing feasibility on clean state predictions results in a reduction of this bias, particularly in high-dimensional settings within more restrictive feasibility criteria (Utkarsh et al., 2025; Christopher et al., 2025b). In consequence, this study adopts a final state projection method formalized below:

$$\hat{z}_1 = \text{ODESolve}\big(z_t, v_\theta, t, 1\big) \qquad \text{(Forward Solve)}$$

$$z_1 = \mathcal{P}_C\big(\hat{z}_1\big) \qquad \text{(Constraint Projection)}$$

$$z_s = \text{ODESolve}\Big(z_1, -\big(z_1 - z_0\big), 1, s\Big) \qquad \text{(Reverse Update)}$$

where the projection operator $\mathcal{P}_C(y) := \arg\min_{x \in C} \|x - y\|_2^2$ returns the nearest feasible point, the ODESolve follows the trajectory defined in Equation (1), and timestep $s = t + \Delta$ for a step size $\Delta > 0$. This formulation follows the sampling approach proposed by (Utkarsh et al., 2025), which removes the intermediate feasibility assumption presented in prior work, solely enforcing *terminal state feasibility*. At each sampling stage, the *Forward Solve* predicts a clean z_1, which is projected onto the constraint manifold. Then, the learned flow is updated such that the end point enforces feasibility.

5 Aligning Training and Inference

Within the existing constrained sampling literature, the training-free property of the sampling approaches is often regarded as a merit of the methodology. However, existing diffusion and flow matching objectives are designed to sample from p_{data}, thus remaining misaligned with the downstream task of sampling from p_C. Indeed, the standard training objective optimizes the unconstrained generative path (Figure 1 (a)), while the actual deployment objective is strongly influenced by the feasibility correction (Figure 1 (b)). These observations suggest that sampling-time enforcement alone creates a disconnect between what the models is trained to optimize and what is ultimately evaluated in deployment.

While this challenge remains unaddressed for constrained generative modeling, machine learning for optimization is an area of the literature where end-to-end learning has been explored extensively (Mandi et al., 2024b). This area of research formalizes a downstream task:

$$\mathbf{x}^\star(\mathbf{c}) = \arg\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{c}) \tag{4a}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x}, \mathbf{c}) \leq 0; \quad \mathbf{h}(\mathbf{x}, \mathbf{c}) = 0 \tag{4b}$$

where $\mathbf{x}^\star(\mathbf{c})$ is an optimal decision of a constrained optimization problem parameterized by $\mathbf{c}$, a prediction by an ML model. While earlier works focused on prediction-focused learning (Hu et al., 2016), minimizing the mean squared error between the predicted parameters $\hat{\mathbf{c}}$ and the true parameters $\mathbf{c}$, later studies showed that decision-focused learning (Wilder et al., 2019), minimizing the downstream loss,

$$\mathcal{L}_{\text{DFL}} := f(\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) - f(\mathbf{x}^\star(\mathbf{c}), \mathbf{c}), \tag{5}$$

results in improved downstream performance. Theoretically, this improvement provided by decision-focused learning, Equation (5), can be characterized by a discontinuity between the the downstream and the prediction-focused loss; while the global optimum between the two losses is aligned, sharp changes in $\mathbf{x}^\star(\hat{\mathbf{c}})$ can result in suboptimality increasing even when prediction quality improves (Demirović et al., 2019).

5.1 Constraint-Aware Flow Matching Objective

Building from the problem formulation presented in Equation (4), this section extends the end-to-end formulation leveraged by decision-focused learning to the setting of constrained generation. Bridging the decision-focused learning formulation to the constrained generation task in Equation (3), let $\mathbf{x}^\star(\mathbf{c}) := z_1$, $\mathbf{c} := \mathbf{z}_1$, and $f(\mathbf{x}, \mathbf{c}) := \|\mathbf{x} - \mathbf{c}\|^2$, the objective of the projection operator, recovering the optimization:

$$\mathbf{x}^\star(\mathbf{c}) = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{c}\|^2 \tag{6a}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x}) \leq 0; \quad \mathbf{h}(\mathbf{x}) = 0 \tag{6b}$$

Assuming $\mathbf{z}_1 \in C$, this formulation recovers the training sample exactly, analogous to decision-focused learning under the true parameterization of $\mathbf{c}$. At training time, then, the standard flow matching objective aligns with prediction-focused learning; specifically, it optimizes the prediction task directly, as we define $\hat{\mathbf{c}} := \hat{z}_1$, the unconstrained prediction.

As a result, training-free constrained sampling approaches realize a gap between the prediction-focused task and the downstream decision-focused task, mirroring the challenges that have been exhaustively studied within decision-focused learning. While this discontinuity especially emerges when applying nonconvex constraints, where small changes in the prediction task can result in larges changes in the minimizer of the projection, the gap is present even in convex cases as illustrated in Figure 2. Indeed, the example shows that even in simple convex cases, predictions $\hat{z}_1$ with identical mean square error can result in very different downstream decisions z_1.

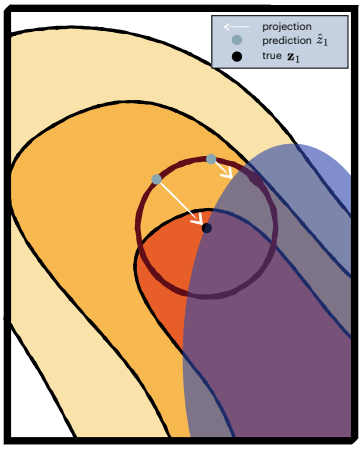


Figure 2: Example of two predictions with identical prediction-focused losses resulting in substantially different decision-focused losses.

A natural objection could be that different parameterizations of generative models may allow training-free constrained sampling to address this gap. For instance, considering the loss landscape in Figure 2, one may propose adopting a proximal method to sample from high density regions while satisfying the constraint. As such, prior works have leveraged score-matching as an approach for estimating the density function gradient, enabling proximal gradient optimization (Christopher et al., 2024; Blanke et al., 2025). However, the disconnect between the training and sampling process remains the central challenge in this case as well; as the proximal algorithm diverges from the forward diffusion trajectory, score estimation becomes increasingly inaccurate due to the absence of training data in these low density regions (Christopher et al., 2025b; Song & Ermon, 2019). Hence, a critical step to closing the gap with standard generative methods remains aligned, end-to-end training processes for constrained samplers.

In order to derive this end-to-end training objective, first consider an end-to-end variant of the standard flow matching objective, which backpropagates through the Euler rollout in Equation (Forward Solve). Indeed, recent approaches have explored this formulation for post-training flow matching models, where $\hat{z}_1$ is useful to define a reward function (Liang et al., 2026; Wang et al., 2026):

$$\mathcal{L}_{Rollout} = \| \overbrace{\left(\mathbf{z}_t + \int_t^1 v_\theta(z_s, s)\right)}^{\text{integrated endpoint } \hat{z}_1} - \mathbf{z}_1\|^2 = \|\hat{z}_1 - \mathbf{z}_1\|^2 \;=\; \|\hat{\mathbf{c}} - \mathbf{c}\|^2 \quad \text{(Prediction-Focused Loss)}$$

where z_s, for $s \in [t, 1]$, is a state obtained by integrating the learned velocity field forward from the training sample $z_t = \mathbf{z}_t$. Next, to align the flow matching loss with the downstream task, the mean square error optimization is imposed on the optimal solution $\mathbf{x}^\star(\hat{\mathbf{c}}) := z_1$ rather than $\hat{\mathbf{c}} := \hat{z}_1$,

restoring alignment with the true endpoint. The loss $\mathcal{L}_{Rollout}$ is thus extended under this framing to yield the *constraint-aware flow matching objective*:

$$\mathcal{L}_{CAFM} = \Big\| \overbrace{\Big(\mathcal{P}_C\big(\mathbf{z}_t + \int_t^1 v_\theta(z_s, s)\big) - \mathbf{z}_0\Big)}^{\text{projected integrated endpoint } z_1} - \mathbf{z}_1 \Big\|^2 = \|\mathbf{x} - \mathbf{c}\|^2 - 0 = f(\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) - f(\mathbf{x}^\star(\mathbf{c}), \mathbf{c})$$

(Decision-Focused Loss)

recovering the decision-focused loss in Equation (5).

5.2 Accelerated and Stable Training with Rectified Flows

While this end-to-end training framework provides an analytically aligned framing for the constraint-aware loss, it is impractical to adopt this training objective directly. Indeed, backpropagation through the forward ODE solve remains plagued by prohibitive memory cost, high runtime requirements, and training instability (Liang et al., 2026; Wang et al., 2026). In practice, this results in an expensive training regime which may even result in quality regression as compared to standard flow matching due to gradient explosion when computing the backward pass through the ODE. Instead, we propose a scalable alternative which leverages local approximations of the forward ODE solve to stabilize the backward pass.

We begin by revisiting the flow matching objective, defining the conditional path $u_t(\mathbf{z}_t|\mathbf{z}_0, \mathbf{z}_1)$ as a linear interpolation between $\mathbf{z_1}$ and $\mathbf{z_0}$, equivalent to *1-rectified flow* training (Liu et al., 2022):

$$\begin{aligned}
\mathcal{L}_{FM} = \| \overbrace{v_\theta(\mathbf{z}_t, t)}^{\text{predicted}} - \overbrace{(\mathbf{z}_1 - \mathbf{z}_0)}^{\text{target velocity}} \|^2 &\propto \overbrace{\|(1-t)v_\theta(\mathbf{z}_t, t) - (1-t)(\mathbf{z}_1 - \mathbf{z}_0)\|^2}^{\text{time-scaled weighting}} \\
&= \|(1-t)v_\theta(\mathbf{z}_t, t) - (\mathbf{z}_1 - \mathbf{z}_t)\|^2 \\
&= \| \underbrace{\mathbf{z}_t + (1-t)v_\theta(\mathbf{z}_t, t)}_{\hat{z}_1^{\text{inst}} \text{ approximation}} - \mathbf{z}_1 \|^2
\end{aligned}$$

Importantly, as the flow matching loss approaches zero, $\hat{z}_1^{\text{inst}} = \mathbf{z}_t + (1-t)v_\theta(\mathbf{z}_t, t)$ approaches the true training end point $\mathbf{z}_1$. As rectification reduces variation in the velocity along the induced trajectory, the instantaneous endpoint approximation $\hat{z}_1^{\text{inst}}$ becomes increasingly consistent with the integrated endpoint $\hat{z}_1^{\text{ODE}} = \mathbf{z}_t + \int_t^1 v_\theta(z_s, s)ds$ following this linear path.

In practice, this reduces the divergence between $\hat{z}_1^{\text{ODE}}$ and $\hat{z}_1^{\text{inst}}$ such that the local approximation can be substituted for the integrated endpoint, removing the need to backpropagate through the ODE. This divergence can be quantified as

$$\hat{z}_1^{\text{ODE}} - \hat{z}_1^{\text{inst}} = \int_t^1 \big[v_\theta(z_s, s) - v_\theta(\mathbf{z}_t, t)\big] ds,$$

which approaches zero as the velocity becomes constant across the generated trajectory. While empirically, we observe that 1-rectified flow training is sufficient for closing this gap, the discrepancy can be improved further through continual training. Starting from a learned flow, *2-rectified flow* pairs each sample $\mathbf{z}_0$ with the endpoint obtained under the current model $\hat{z}_1^{\text{ODE}}$ and retrains the velocity field under the linear interpolation of these updated endpoints (Liu et al., 2022).

Ultimately, this rectified flow enables us to efficiently approximate to $z_1 \approx \mathcal{P}_C(\hat{z}_1^{\text{inst}})$, removing the dependency on the ODE rollout:

$$\mathcal{L}_{CAFM+} = \| \overbrace{\mathcal{P}_C\big(\mathbf{z}_t + (1-t)v_\theta(\mathbf{z}_t, t)\big)}^{\text{projected approximate endpoint}} - \mathbf{z}_1 \|^2. \tag{10}$$

Appendix E provides a qualitative comparison of the performance across 1-rectified flow training, 2-rectified flow training, and the constrained rollout training derived in the previous section.

5.3 Differentiable Projection Operators

Having justified the adoption of the constraint-aware flow matching objective, the next challenge is defining the projection problem such that it can be treated as a differentiable operation. The key

observation is that, when implemented by an iterative optimization algorithm, the projection can be unrolled into the computational graph. To facilitate end-to-end training, the operator $\mathcal{P}_C$ is integrated as a *differentiable projection layer*: a parameterized mapping from the predicted clean state $\hat{z}_1$ to the projected state $z_1 = \mathcal{P}_C(\hat{z}_1)$. However, rather than requiring a closed-form projector, $\mathcal{P}_C$ is implemented by an iterative optimization routine. Let Φ denote a single update of this optimization algorithm, starting from the initialization $\mathbf{x}^{(0)}(\hat{\mathbf{c}}) = \hat{z}_1$, and

$$\mathbf{x}^{(k+1)}(\hat{\mathbf{c}}) = \Phi\big(\mathbf{x}^{(k)}, \hat{\mathbf{c}}\big), \quad k = 0, \dots, K-1.$$

Then, the projection layer used during training is composed of a series of iterative updates:

$$\mathcal{P}_C\big(\hat{z}_1\big) = \mathbf{x}^\star(\hat{\mathbf{c}}) \approx \mathbf{x}^{(K)}(\hat{\mathbf{c}}) = \big(\Phi \circ \cdots \circ \Phi\big)\big(\hat{z}_1\big)$$

where $\mathbf{x}^\star(\hat{\mathbf{c}})$ denotes the fixed-point of the update map, satisfying

$$\mathbf{x}^\star(\hat{\mathbf{c}}) = \Phi\big(\mathbf{x}^\star, \hat{\mathbf{c}}\big).$$

Because the optimization routine is run for only K iterations, the layer uses $\mathbf{x}^{(K)}$ as a finite-step approximation to this fixed-point. Under this view, the unrolled operations are added to the graph via the chain rule, enabling gradient flow from the downstream objective.

6 Experiments

Three real-world problems are selected for the downstream evaluation of the proposed methodology:

1. **Partial Differential Equation** constrained physical systems, comprising three distinct problems,
2. **Wind Velocity Field Estimation** over spatiotemporal horizons under a prescribed coherence,
3. **Microstructure Inverse-Design** for material discovery under porosity constraints.

For transparent evaluation, CAFM uses an identical sampling setup to Physics-Constrained Flow Matching (PCFM) (Utkarsh et al., 2025) in all settings, isolating the impact of end-to-end training. Additional constrained sampling approaches are included when applicable, and Functional Flow Matching (FFM) (Lipman et al., 2022) serves as a reference for unconstrained generative performance. The method performance is evaluated by *accuracy* and *feasibility* of the outputs, with domain-specific evaluation metrics selected for each task. Extended details are provided in Appendix B.

6.1 Application 1: Partial Differential Equations

The first setting benchmarks CAFM across a set of challenging partial differential equations. Evaluations on PDE governed systems have recently been adopted as a standard stress test for constrained generative models, providing real-world scientific complexity that remains difficult for neural operators and generative models (Utkarsh et al., 2025; Bastek et al., 2024).

The selected settings introduce increasing constraint complexity, as implemented by (Utkarsh et al., 2025). Navier–Stokes imposes initial vorticity (IC) and linear global mass conservation constraints (CL). Burgers BC incorporates localized, pointwise boundary conditions increasing structural rigidity, while Burgers IC enforces initial conditions (IC) and both nonlinear global nonlinear mass conservation and local conservation updates (CL). Following Utkarsh et al. (2025), solution subsets are constrained on held-out IC or BC, allowing the experiment to evaluate the performance of constraint-aware training when the test-time constraints have not appeared in the training set.

Several clear trends emerge across the settings reported in Table 1. First, while FFM (Kerrigan et al., 2023) reports strong reconstruction metrics in applications like Navier–Stokes, the high constraint residuals demonstrate that unconstrained approaches are not viable for these complex PDE settings. Next, while the ECI framework (Cheng et al., 2024) is generally competetive with the other constrained samplers for constraint satisfaction, it generally reports less competitive accuracy metrics as compared to the other constrained approaches. Finally, and most notably, the table shows that CAFM consistently outperforms its constrained sampling counterpart PCFM, illustrating that the end-to-end training improves both accuracy and feasibility. These results offer strong evidence that CAFM can effectively scale across increasingly difficult constraint sets. Furthermore, the strong performance illustrates that this method provides strong generalization to out-of-distribution constraint sets, given the evaluation is conducted over unseen initial conditions and boundary conditions.

Table 1: Generative performance for zero-shot methods on constrained PDEs. Navier–Stokes enforces global conservation laws (CL) as linear constraints along with initial condition (IC) constraints, while Burgers applies nonlinear conservation-law constraints together with either IC or boundary condition (BC) constraints. Lower values indicate better performance. Best and second-best mean results are shown in **bold** and underlined, respectively, considering only methods that satisfy the benchmark constraints.

Metric	Benchmark	CAFM	PCFM	ECI	FFM
MMSE	Navier–Stokes	$\mathbf{(2.83 \pm 0.34)e{-1}}$	$(3.04 \pm 0.36)e{-1}$	$(2.84 \pm 0.17)e{-1}$	$(2.01 \pm 0.12)e{-1}$
SMSE		$(1.78 \pm 0.54)e{-1}$	$(2.01 \pm 0.22)e{-1}$	$\mathbf{(6.40 \pm 0.62)e{-2}}$	$(4.42 \pm 0.20)e{-2}$
CV (IC)		$\mathbf{(0.00 \pm 0.00)e{+0}}$	$\mathbf{(0.00 \pm 0.00)e{+0}}$	$(8.30 \pm 0.03)e{-17}$	$(2.06 \pm 0.03)e{-1}$
CV (CL)		$(1.32 \pm 0.30)e{-16}$	$(1.45 \pm 0.04)e{-16}$	$\mathbf{(6.74 \pm 0.77)e{-17}}$	$(1.23 \pm 0.15)e{-2}$
MMSE	Burgers BC	$\mathbf{(6.12 \pm 3.55)e{-4}}$	$(1.89 \pm 1.04)e{-3}$	$(3.42 \pm 0.39)e{-3}$	$(1.34 \pm 0.27)e{-2}$
SMSE		$(1.05 \pm 0.21)e{-3}$	$\mathbf{(4.60 \pm 0.70)e{-4}}$	$(9.97 \pm 0.56)e{-3}$	$(1.70 \pm 0.11)e{-2}$
CV (BC)		$\mathbf{(4.44 \pm 0.00)e{-16}}$	$\mathbf{(4.44 \pm 0.00)e{-16}}$	$(4.48 \pm 0.00)e{-16}$	$(5.88 \pm 0.54)e{-2}$
CV (CL)		$(3.55 \pm 0.23)e{-15}$	$\mathbf{(3.27 \pm 0.08)e{-15}}$	$(1.21 \pm 0.03)e{-14}$	$(2.50 \pm 0.44)e{-4}$
MMSE	Burgers IC	$\mathbf{(3.56 \pm 0.38)e{-2}}$	$(3.75 \pm 0.23)e{-2}$	$(7.73 \pm 0.18)e{-2}$	$(4.83 \pm 0.15)e{-2}$
SMSE		$(5.75 \pm 0.77)e{-2}$	$(1.01 \pm 0.03)e{-1}$	$\mathbf{(1.72 \pm 0.24)e{-2}}$	$(4.03 \pm 0.27)e{-2}$
CV (IC)		$(4.67 \pm 1.83)e{-22}$	$\mathbf{(3.26 \pm 0.34)e{-22}}$	$(4.31 \pm 0.03)e{-17}$	$(7.92 \pm 0.39)e{-2}$
CV (CL)		$\mathbf{(3.20 \pm 0.19)e{-15}}$	$(3.27 \pm 0.24)e{-15}$	$(8.52 \pm 2.14)e{-12}$	$(1.51 \pm 0.32)e{-3}$

6.2 Application 2: Microweather Wind Velocity Field Estimation

The next evaluation assesses the proposed method for wind velocity field prediction. Motivated by the real-world gap in prediction of microweather, localized weather conditions which are computationally prohibitive to explicitly compute with fluid dynamics, this experiment requires that the generative model predicts mean and variance wind fields while maintaining realistic spatiotemporal coherence (Shah et al., 2025; Nithya et al., 2024). Due to the prohibitive cost of collecting this wind velocity data, which is dependent on the deployment of sensors, the training set is composed of sparse synthetic measurements to mimic the challenges of collecting spatially-dense wind velocity data with deployed sensors.

The evaluation targets a fixed-grid version of the wind velocity field estimation setting described by Warner et al. (2026), where the goal is to generate realistic wind samples from sparse spatiotemporal observations while preserving a prescribed coherence structure. As in the original setup, wind velocity is modeled using the standard decomposition (Carassale & Solari, 2006):

$$\mathbf{U}(\mathbf{r}, \omega) = \boldsymbol{\mu}_u(\mathbf{r}) + \mathbf{W}(\mathbf{r}, \omega)$$

where $\mathbf{r} = [x_1, x_2, x_3, t]$ represents a static spatiotemporal point, $\boldsymbol{\mu}_u$ is a deterministic mean function over $u \in \mathbb{R}^{x_1 \times x_2 \times x_3 \times t}$, and $\mathbf{W}(\mathbf{r}, \omega)$ is a zero-mean, stationary Gaussian process. The application considers a spatiotemporal correlation constraint, which is defined between two grid locations $\mathbf{r}$ and $\mathbf{r}'$ by the coherence function:

$$\mathbf{Coh}(\mathbf{r}, \mathbf{r}', n) = \exp\Big(-\frac{n\|\mathbf{d}^T(\mathbf{r} - \mathbf{r}')\|}{\|\boldsymbol{\mu}_u(\mathbf{r}) - \boldsymbol{\mu}_u(\mathbf{r}')\|}\Big)$$

where n is frequency and $\mathbf{d}$ contains directional decay coefficients. Following Warner et al. (2026), the evaluation focuses on a two-dimensional spatial plane over time, $\mathbf{r} = [x_2, x_3, t]$. In contrast to the continuous functional representation used in the original setup, our implementation operates on a fixed grid with $N_{x_2} = 10$, $N_{x_3} = 10$, and $N_t = 256$, so each wind sample is represented as an array $u \in \mathbb{R}^{N_{x_2} \times N_{x_3} \times N_t} = \mathbb{R}^{10 \times 10 \times 256}$.

Empirical results are reported in Table 2a. The unconstrained baseline, Functional Flow Matching (FFM) (Kerrigan et al., 2023), is ineffective in this experimental setting, reporting the worst performance across all metrics; the higher MMSE and Variance MSE suggests that poor adherence to the coherence constraint results in wind velocities which fail to align with the training data. Conversely, the Projected Diffusion Model (PDM) (Christopher et al., 2024), which we extend to operate with the flow matching backbone as in (Utkarsh et al., 2025), and the PCFM model report the strongest results for Variance MSE, while improving the MMSE and CV metrics over FFM. In contrast, CAFM reports the strongest MMSE and constraint violation, lowering the coherence residual by an order of

Table 2: Final metric summaries for the two generative modeling settings. Lower values indicate better performance, with best results in **bold** and second-best results underlined.

	Shared Sampler		*Other Samplers*	
Metric	**CAFM (Ours)**	**PCFM**	**PDM**	**FFM**
MMSE	**4.60e-02**	4.85e-02	4.85e-02	5.05e-02
Variance MSE	4.73e-01	4.17e-01	**4.13e-01**	4.85e-01
CV (Coherence)	**3.60e-04**	4.59e-03	2.68e-03	5.136e-01

(a) Fixed-grid wind modeling.

	Shared Sampler		*Other Samplers*	
Metric	**CAFM (Ours)**	**PCFM**	**PDM**	**FM**
MSE	**3.69e-01**	5.00e-01	4.32e-01	4.51e-01
NNMSE	**9.05e-02**	9.59e-02	1.41e-01	1.45e-01
CV (Porosity)	**0.00e+00**	**0.00e+00**	**0.00e+00**	9.38e-01

(b) Microstructure inverse design.

magnitude. While the variance trade-off is interesting to note, as it suggests that the constraint-aware objective may suppress high-variance structures to avoid projection instability, the significant improvements in feasibility and reconstruction accuracy mark a pronounced improvement over existing constrained sampling approaches.

6.3 Application 3: Microstructure Inverse-Design

The final setting benchmarks the proposed method on a material science inverse-design task. Due to the prohibitive cost of collecting microstructure imaging data, generative models have been viewed as a highly promising direction for obtaining necessary samples for discovering structure-property linkages (Christopher et al., 2024; Zampini et al., 2025; Chun et al., 2020). For this evaluation, a sparse dataset of Bentheimer sandstone imaging data is adopted (Li et al., 2022) and subsampled to create 256×256 patches rescaled to $[-1, 1]$.

This experiment adopts a convex constraint set, following the definition provided by (Zampini et al., 2025). Specifically, let $\mathbf{r}_{i,j}$ be the pixel value for row i and column j, where $\mathbf{r}_{i,j} \in [-1, 1]$ for all values of i and j. The porosity constraint is then,

$$\textit{porosity} = \frac{1}{n \times m} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{1}(\mathbf{r}_{i,j} < 0),$$

where our constraint enforces a strict target porosity value for each sample (e.g. $\textit{porosity} = 0.5$).

The downstream evaluation metrics are reported in Table 2b. The results for this task parallel the previous two settings in the general trends: FFM struggles to adhere to the prescribed constraint, and the performance suffers because of this; PDM and PCFM remain competitive with one another, consistently improving over the FFM baseline. Meanwhile, CAFM outperforms both PCFM and all other methods across accuracy metrics, while maintaining perfect constraint satisfaction. The empirical evaluation for this setting reiterates the benefit of aligning model training with the downstream objective, providing state-of-the-art performance when utilizing this end-to-end training paradigm.

7 Conclusion

Motivated by the transformative potential of constraint-aware diffusion and flow matching models for scientific applications, this paper focuses on addressing a fundamental misalignment between the training and sampling process of constrained sampling approaches. To this end, *Constraint-Aware Flow Matching* is proposed, a novel end-to-end framework for constrained generative modeling that directly addresses the mismatch between the learned dynamics and downstream objective. The resulting model is able to generate samples that satisfy constraints while improving data fidelity. The empirical results on three challenging real-world benchmarks demonstrate both the effectiveness and the generality of the proposed approach, reporting state-of-the-art performance across increasingly complex constraint sets and data distributions. Beyond improving performance in the studied settings, these findings highlight the importance of explicitly accounting for constraint-enforcement mechanisms during training, rather than treating them solely as a post hoc correction at inference time. More broadly, this paper marks a promising step towards building generative models that are both physically consistent and practically reliable in scientific and engineering applications.

AI USE STATEMENT

In this paper, we used AI tools to assist in creating assets for diagrams and figures and accelerating code development pipelines. These tools were also used in a limited role for refining the written exposition. We have not used generative AI tools in any way to support development of the methodology, in any role contributing to mathematical correctness of the work, nor as an author of the paper. All AI assisted work has been carefully reviewed, and we take full responsibility for the content of the work.

REPRODUCIBILITY STATEMENT

A complete documentation of implementation details is included Appendix B for transparent reproduction of the experiments in this paper. Code is included with the submission and links to anonymized HuggingFace models are submitted with the repository.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pp. 136–145. PMLR, 2017.

Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M Kochmann. Physics-informed diffusion models. *arXiv preprint arXiv:2403.14404*, 2024.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9508–9519, 2020.

Matthieu Blanke, Yongquan Qu, Sara Shamekh, and Pierre Gentine. Strictly constrained generative modeling via split augmented langevin sampling. *arXiv preprint arXiv:2505.18017*, 2025.

Mathieu Blondel, André FT Martins, and Vlad Niculae. Learning with fenchel-young losses. *J. Mach. Learn. Res.*, 21(35):1–69, 2020a.

Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pp. 950–959. PMLR, 2020b.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *Advances in Neural Information Processing Systems*, 35:5230–5242, 2022.

Nicolò Botteghi, Federico Califano, Mannes Poel, and Christoph Brune. Trajectory generation, control, and safety with denoising diffusion probabilistic models. *arXiv preprint arXiv:2306.15512*, 2023.

Luigi Carassale and Giovanni Solari. Monte carlo simulation of wind velocity fields on complex structures. *Journal of Wind Engineering and Industrial Aerodynamics*, 94(5):323–339, 2006.

Joao Carvalho, An T Le, Mark Baierl, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1916–1923. IEEE, 2023.

Chaoran Cheng, Boran Han, Danielle C Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W Mahoney, and Yuyang Wang. Gradient-free generation for hard-constrained systems. *arXiv preprint arXiv:2412.01786*, 2024.

Jacob K Christopher, Stephen Baek, and Nando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Physics-aware diffusion models for micro-structure material design. In *AI for Materials Science Workshop at NeurIPS 2024*. AI For Material Science workshop, at NeurIPS-24, 2025a.

Jacob K Christopher, Austin Seamann, Jingyi Cui, Sagar Khare, and Ferdinando Fioretto. Constrained diffusion for protein design with hard structural constraints. *arXiv preprint arXiv:2510.14989*, 2025b.

Sehyun Chun, Sidhartha Roy, Yen Thi Nguyen, Joseph B Choi, Holavanahalli S Udaykumar, and Stephen S Baek. Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials. *Scientific reports*, 10(1):13307, 2020.

Jingyi Cui, Jacob K Christopher, Ankita Biswas, Prasanna V Balachandran, and Ferdinando Fioretto. Constrained diffusion for accelerated structure relaxation of inorganic solids with point defects. *arXiv preprint arXiv:2602.19153*, 2026.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Emir Demirović, Peter J Stuckey, James Bailey, Jeffrey Chan, Chris Leckie, Kotagiri Ramamohanarao, and Tias Guns. An investigation into prediction+ optimisation for the knapsack problem. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pp. 241–257. Springer, 2019.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1): 9–26, 2022.

Aaron Ferber, Bryan Wilder, Bistra Dilkina, and Milind Tambe. Mipaal: Mixed integer program as a layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1504–1511, 2020a.

Aaron M. Ferber, Bryan Wilder, Bistra Dilkina, and Milind Tambe. Mipaal: Mixed integer program as a layer. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 1504–1511. AAAI Press, 2020b.

Aaron M. Ferber, Taoan Huang, Daochen Zha, Martin Schubert, Benoit Steiner, Bistra Dilkina, and Yuandong Tian. Surco: Learning linear surrogates for combinatorial nonlinear optimization problems. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10034–10052. PMLR, 2023.

Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Wuhua Hu, Ping Wang, and Hoay Beng Gooi. Toward optimal energy management of microgrids via robust two-stage optimization. *IEEE Transactions on smart grid*, 9(2):1161–1174, 2016.

Tariq Berrada Ifriqi, John Nguyen, Karteek Alahari, Jakob Verbeek, and Ricky TQ Chen. Flowception: Temporally expansive flow matching for video generation. *arXiv preprint arXiv:2512.11438*, 2025.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional flow matching. *arXiv preprint arXiv:2305.17209*, 2023.

James Kotary, Jacob Christopher, My H Dinh, and Ferdinando Fioretto. Analyzing and enhancing the backward-pass convergence of unrolled optimization. *arXiv preprint arXiv:2312.17394*, 2023.

R. Li, I. Shikhov, and C. Arns. Bentheimer sandstone image data, March 2022. URL https://doi.org/10.17612/1J6K-SH07. Dataset.

Jinhao Liang, Jacob K Christopher, Sven Koenig, and Ferdinando Fioretto. Simultaneous multi-robot motion planning with projected diffusion models. *arXiv preprint arXiv:2502.03607*, 2025a.

Jinhao Liang, Yixuan Sun, Anirban Samaddar, Sandeep Madireddy, and Ferdinando Fioretto. Chance-constrained flow matching for high-fidelity constraint-aware generation. *arXiv preprint arXiv:2509.25157*, 2025b.

Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. Leapalign: Post-training flow matching models at any generation step by building two-step trajectories. *arXiv preprint arXiv:2604.15311*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Jayanta Mandi and Tias Guns. Interior point solving for lp-based prediction+ optimisation. *Advances in Neural Information Processing Systems*, 33:7272–7282, 2020.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 81:1623–1701, 2024a. doi: 10.48550/arXiv.2307.13565.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024b.

François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 9108–9116, 2023.

Mathias Niepert, Pasquale Minervini, and Luca Franceschi. Implicit mle: backpropagating through discrete exponential family distributions. *Advances in Neural Information Processing Systems*, 34: 14567–14579, 2021.

D S Nithya, Giuseppe Quaranta, Vincenzo Muscarello, and Man Liang. Review of wind flow modelling in urban environments to support the development of urban air mobility. *Drones*, 8 (4), 2024. ISSN 2504-446X. doi: 10.3390/drones8040147. URL https://www.mdpi.com/2504-446X/8/4/147.

Egon Peršak and Miguel F Anjos. Decision-focused forecasting: A differentiable multistage optimisation architecture. *arXiv preprint arXiv:2405.14719*, 2024.

Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. Differentiation of blackbox combinatorial solvers. In *International conference on learning representations*, 2019.

Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Subham Sekhar Sahoo, Anselm Paulus, Marin Vlastelica, Vít Musil, Volodymyr Kuleshov, and Georg Martius. Backpropagation through combinatorial algorithms: Identity with projection works. *arXiv preprint arXiv:2205.15213*, 2022.

Tristan A. Shah, Michael C. Stanley, and James E. Warner. Generative modeling of microweather wind velocities for urban air mobility. In *2025 IEEE Aerospace Conference*, pp. 1–17, 2025. doi: 10.1109/AERO63441.2025.11068624.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. *arXiv preprint arXiv:2506.04171*, 2025.

Po-Wei Wang, Priya Donti, Bryan Wilder, and Zico Kolter. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, pp. 6545–6554. PMLR, 2019.

Ruoyu Wang, Boye Niu, Xiangxin Zhou, Yushi Huang, Tongliang Liu, and Chi Zhang. Exploring the design space of reward backpropagation for flow matching. *arXiv preprint arXiv:2606.11075*, 2026.

Tsun-Hsuan Johnson Wang, Juntian Zheng, Pingchuan Ma, Yilun Du, Byungchul Kim, Andrew Spielberg, Josh Tenenbaum, Chuang Gan, and Daniela Rus. Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. *Advances in Neural Information Processing Systems*, 36:44398–44423, 2023.

James E. Warner, Tristan A. Shah, Patrick E. Leser, Geoffrey F. Bomarito, Joshua D. Pribe, and Michael C. Stanley. Latent generative modeling of random fields from limited training data, 2026. URL https://arxiv.org/abs/2505.13007.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1658–1665, 2019.

Liming Wu, Rui Jiao, Qi Li, Mingze Li, Songyou Li, Shifeng Jin, and Wenbing Huang. Dmflow: Disordered materials generation by flow matching. *arXiv preprint arXiv:2602.04734*, 2026.

Stefano Zampini, Jacob K Christopher, Luca Oneto, Davide Anguita, and Ferdinando Fioretto. Training-free constrained generation with stable diffusion models. *arXiv preprint arXiv:2502.05625*, 2025.

A Discussion

Connection to Multistage Optimization. While classical decision-focused learning formulations consider a single downstream optimization problem, many real-world systems are more accurately framed as sequential or multistage decision-making, where predictions inform a sequence of optimization steps rather than a single decision. Recent work has targeted such applications, leveraging a predictive model which, rather than parameterizing a single optimization problem, induces a policy through repeated applications of an optimization operator over a temporal horizon (Peršak & Anjos, 2024). Formally, let $\mathbf{x}_{(k)}$ denote the state at iteration k, and let $\hat{\mathbf{c}}_{(k)}$ denote the predicted parameters at that stage. A sequential decision process can be written:

$$\mathbf{x}_{(k+1)} = \mathcal{O}\big(\mathbf{x}_{(k)}, \hat{\mathbf{c}}_{(k)}\big)$$

where $\mathcal{O}$ is a forward operator, coupling the prediction and optimization steps. The final decision is then given by the trajectory endpoint. This perspective generalizes Equation (5) by replacing a single optimization problem with a policy induced by the repeated optimization. This formulation naturally extends to our methodology, which we believe is a fitting perspective for understanding how the single-stage optimization in Equation (6), which is isolated at training time, generalizes to the multistage sampling process.

Out-of-Distribution Generation. One of the strongest arguments for the adoption of constrained sampling algorithms, as opposed to constraint-aware training, is the native ability of sampling-time generalization to arbitrary constraint sets. The evaluation has demonstrated that CAFM effectively handles held-out constraint specifications (Section 6.1) and under-represented design properties (Section 6.3), demonstrating that our training-aligned sampler remains robust to out-of-distribution tasks and constraints. Furthermore, it is worth noting that additional sampling-time constraints can be seamlessly incorporated into CAFM, creating a hybrid between CAFM and PCFM which remains effective due to the sampling-time projections. As such, the proposed approach remains general in its handling of domain shift, avoiding this pitfall of existing constraint-aware training methods.

Training Complexity. Analogous to decision-focused learning approaches, CAFM's improved downstream performance is provided by incorporating the optimization problem into the training objective. While existing decision-focused learning literature has provided an array of efficiency imporvements that are leveraged by our implementation, the additional computational overhead introduced to solve the projection for each forward pass during training should be acknowledged. To reduce the additional training requirements, the CAFM objective can be applied as a finetuning step, initializing from the pretrained flow matching weights, decreasing the total number of training steps needed to reach convergence. Appendix D provides an additional analysis of this finetuning approach to illustrate the efficiency gains provided by warm starting with pretrained weights.

B Experimental Setups and Reproducibility

This section is dedicated to documenting the specific evaluation settings, hyperparameters, and reproducibility details. All experiments were conducted on a single NVIDIA A100-80GB GPU. For transparency, we additionally provide our code base and trained model weights with the submission, encouraging reviewers to refer to the exact code implementation for additional implementation details.

Across all settings, CAFM trains the flow model with the same sampling-time projection mechanism used during constrained inference. Unless otherwise stated, optimization is performed using Adam, with $\beta = (0.9, 0.999)$ and no weight decay. This is implemented with a differentiable equality-constrained SQP projection layer. Given a predicted endpoint u, the projector computes the constraint residual $h(u)$, forms the damped normal matrix $JJ^\top + \epsilon I$, solves for the Lagrange multiplier update, and applies the correction. The default damping is $\epsilon = 10^{-6}$, with an adaptive Cholesky fallback. The projected velocity target is computed as

$$\frac{u_1^{\text{proj}} - u_t}{\max(1 - t, 10^{-3})}.$$

For projected training losses, we use residual regularization weight 10^{-3}, sample times $t \in [0, 0.95]$, and a projection clamp of 10^{-3}.

B.1 Partial Differential Equations

Evaluation Metrics. The selected evaluation criteria follows the metrics used by Utkarsh et al. (2025). Each is described below:

- **Mean Squared Errors of the Mean (MMSE):** Measures the squared error between the generated mean field and the ground-truth mean field. It evaluates how accurately the model reproduces the expected or average behavior of the target distribution.
- **Standard Deviation (SMSE):** Measures the squared error between the generated standard deviation field and the ground-truth standard deviation field. It evaluates how well the model captures the spatial distribution of uncertainty or variability.
- **Constraint Violation of Boundary / Initial Conditions (CV (BC/IC)):** Measures the constraint violation of the boundary conditions or initial conditions, depending on which are relevant for the problem. Navier–Stokes and Burgers IC enforce initial conditions, while Burgers BC enforce boundary conditions.
- **Constraint Violation of Conservation Laws (CV (CL)):** Measures the constraint violation of the conservation laws, which are detemined by the specific problem. These are linear for Navier–Stokes and nonlinear for Burgers.

Baselines. The following baselines are adopted for comparison in this experimental setting:

- **Physics-Constrained Flow Matching (PCFM) (Utkarsh et al., 2025):** Uses an identical sampling setup to CAFM, isolating the impact of end-to-end training.
- **Extrapolation, Correction, and Interpolation (ECI) (Cheng et al., 2024):** Uses a similar final state correction to PCFM and CAFM, but operates in a gradient-free manner. Strong performance reported for this PDE setting, making it a relevant baseline.
- **Functional Flow Matching (FFM) (Kerrigan et al., 2023):** Unconstrained generative flow matching baseline, adapted from standard flow matching for spatiotemporal function spaces.

Implementation. The PDE experiments evaluate constrained generation across Navier–Stokes, Burgers BC, and Burgers IC systems. If not specified, all implementation details follow the implementation from (Utkarsh et al., 2025). The flow model uses an FNO-based encoder for PDE fields. Unless otherwise specified, the differentiable projection layer uses `max_iter=3` SQP/Gauss–Newton correction steps with exact Jacobians.

For Navier–Stokes, models are trained for 50,000 iterations with batch size 16 and learning rate 3×10^{-5}. Samples have dimensions $32 \times 32 \times 25$. Because of the larger dimensionality of this setting, projection uses `max_iter=1` with `jacobian_mode=penalty`, `penalty_steps=5`, and `penalty_step_size=1e-2`. For Burgers with boundary-condition constraints, models are trained for 20,000 iterations with batch size 256 and learning rate 3×10^{-5}. We use exact Jacobians, `max_iter=3`. Samples again have dimensions 101×101. For Burgers with initial-condition constraints, models are trained for 20,000 iterations with batch size 256 and learning rate 3×10^{-5}. We use exact Jacobians and `max_iter=3`. Samples have dimensions 101×101.

B.2 Microweather Wind Velocity Field Estimation

Evaluation Metrics. The selected evaluation criteria follows the metrics used by Warner et al. (2026). Each is described below:

- **Mean Squared Errors of the Mean (MMSE):** Measures the squared error between the generated mean field and the ground-truth mean field. It evaluates how accurately the model reproduces the expected or average behavior of the target distribution.
- **Mean Squared Errors of the Variance (Variance MSE):** Measures the squared error between the generated variance field and the ground-truth variance field. It evaluates how accurately the model reproduces the second-order statistics (variance structure) of the target distribution.
- **Constraint Violation of Coherence Function (CV (Coherence)):** Measures constraint violation of the prescribed coherence function.

Baselines. The following baselines are adopted for comparison in this experimental setting:

- **Physics-Constrained Flow Matching (PCFM) (Utkarsh et al., 2025):** Uses an identical sampling setup to CAFM, isolating the impact of end-to-end training.
- **Projected Diffusion Models (PDM) (Christopher et al., 2024):** Constrained sampler that applies projections at intermediate steps rather than at the final state. We adapt the methodology to use a flow-matching backbone as opposed to score-based diffusion, keeping the sampler implementation otherwise consistent.
- **Functional Flow Matching (FFM) (Kerrigan et al., 2023):** Unconstrained generative flow matching baseline, adapted from standard flow matching for spatiotemporal function spaces.

Implementation. The wind velocity experiments use a fixed-grid version of the microweather field estimation problem. Unlike the continuous DeepONet representation (Lu et al., 2019) used in prior work, this implementation models wind samples directly as fixed-grid tensors (Warner et al., 2026). Each sample is represented on a 10×10 spatial grid with a temporal horizon of length 256. Wind values are normalized to the range $[-2, 47]$, and conditioning is provided through the sparse SODAR observation pattern.

The flow model is a data-space flow matching network with hidden size 128, four hidden layers, minimum noise scale $\sigma_{\min} = 10^{-2}$, and 12 ODE integration steps. The constraint enforces wind coherence over all point pairs. Coherence is computed across the entire grid. The constraint tolerance is 10^{-3}. The baseline flow model is trained with batch size 8, 25 epochs, Adam learning rate 10^{-3}, and gradient clipping at norm 1.0. The CAFM setting is trained with projection enabled, batch size 8, 25 epochs, Adam learning rate 10^{-4}, and gradient clipping at norm 1.0. Runs are evaluated over seeds 0, 1, and 2.

Training-time projection is performed in data space using an unrolled Augmented Lagrangian method rather than SQP. The training projector and evaluation-time projection use a full-grid augmented Lagrangian method, Adam learning rate 10^{-2}, `inner_iters=64`, `outer_iters=8`, `rho_init=1.0`, `rho_scale=2.0`, `rho_max=128.0`, and `beta=0.0`. Projection is applied at every sampling step.

B.3 Microstructure Inverse-Design

Evaluation Metrics.

- **Mean Squared Error (MSE):** Measures the average squared error between each sample and the ground-truth mean field. This metric is adopted rather than MMSE as the inverse-design targets shift the generated mean field out-of-distribution, making it not directly comparable to the training data.
- **Nearest Neighbor Mean Squared Error (NNMSE):** Measures the average squared error between each sample and the nearest neighbor in the training set. Benchmarks how close samples are to the ground truth data.
- **Constraint Violation of Porosity (CV (Porosity)):** Measures percentage of samples violating the exact porosity target.

Baselines. The following baselines are adopted for comparison in this experimental setting:

- **Physics-Constrained Flow Matching (PCFM) (Utkarsh et al., 2025):** Uses an identical sampling setup to CAFM, isolating the impact of end-to-end training.
- **Projected Diffusion Models (PDM) (Christopher et al., 2024):** Constrained sampler that applies projections at intermediate steps rather than at the final state. We adapt the methodology to use a flow-matching backbone as opposed to score-based diffusion, keeping the sampler implementation otherwise consistent.
- **Flow Matching (FM) (Lipman et al., 2022):** Unconstrained generative flow matching baseline.

Implementation. The microstructure experiments evaluate RGB Bentheimer sandstone generation under a target porosity constraint. Images are generated at resolution 256×256 using the dataset (Li et al., 2022). During training, images are randomly horizontally flipped and rescaled to $[-1, 1]$.

The generative backbone is a UNet with base channel count 128, channel multipliers $[1, 1, 2, 2, 4, 4]$, two residual blocks per resolution, attention at resolution 16, dropout 0, and EMA decay 0.999. The flow process uses 1000 training flow steps. The baseline flow matching model is trained with batch size 16, Adam learning rate 2×10^{-5}, gradient clipping at norm 1.0, snapshots every 5000 steps, and validation every 250 steps. The CAFM run is trained with batch size 8, maximum 35,000 steps, and Adam learning rate 2×10^{-5}. Sampling uses Euler integration with 250 steps and batch size 8. Evaluation uses 1000 generated samples. In `pcfm`, the predicted endpoint is projected at each Euler step. Both `pcfm` and `pdm` also apply a hard save-time projection with $\epsilon = 0$ to enforce the exact target porosity/count constraint after sampling.

The microstructure constraint is a porosity/count constraint. RGB samples are first converted to grayscale using weights $[0.299, 0.587, 0.114]$. Pixels are thresholded at 0.0, and the target fraction is set to $k = 0.6251$. The projection operator uses an unrolled heuristic top-k/count projector with a differentiable surrogate update. Projection uses step size 0.1, `n_iter=5`, and the fixed-point iteration backpropagation rule.

C Backward-Pass Implementation Details

Standard unrolling approaches often suffer from high memory cost and backward-pass instability. Instead, following Kotary et al. (2023), *folded optimization* is adopted, which extends unrolling by differentiating the solver's fixed-point conditions, providing gradients via a single linear solve rather than explicitly unrolling K iterations (Kotary et al., 2023). This enables compatibility with highly optimized blackbox solvers to compute the fixed-point, as it is only necessary to differentiate $\mathbf{x}^{(K)}(\hat{\mathbf{c}})$ rather than unrolling previous iterates.

Projecting with Sequential Quadratic Programming.

To accommodate constraint sets defined by general nonlinear relations, including nonconvex structure, optimization updates Φ is practically implemented using Sequential Quadratic Programming (SQP) correction steps. In contrast to proximal gradient updates, which are often more applicable for simple convex sets, SQP addresses nonlinearity by solving a sequence of local quadratic subproblems obtained from linearized constraints and a quadratic model of the Lagrangian

$$\mathcal{L}(z_1, \lambda, \nu) = \tfrac{1}{2}\|\hat{z}_1 - z_1\|^2 + \lambda^\top h(z_1) + \nu^\top g(z_1).$$

where $\nu \geq 0$. This follows the rationale provided by Kotary et al. (2023), who use SQP for nonconvex nonlinear constrained optimization layers, where feasibility cannot be enforced through a simple convex proximal map.

Efficient Backward-Pass. To propagate the gradients through this projection layer, the chain rule is then considered:

$$\frac{\partial \mathcal{L}}{\partial v_\theta} = \frac{\partial \mathcal{L}}{\partial \mathcal{P}_C(\hat{z})} \cdot \underbrace{\frac{\partial \mathcal{P}_C(\hat{z})}{\partial \hat{z}}}_{\text{Jacobian of projection}} \cdot \frac{\partial \hat{z}}{\partial v_\theta}$$

However, materializing the Jacobian $\frac{\partial \mathcal{P}_C(\hat{z})}{\partial \hat{z}}$ explicitly results in a high memory footprint, leading to challenges with scaling to higher dimensional settings and models. As opposed to explicitly constructing this matrix, consider that the full Jacobian is never required; rather, the vector-Jacobian product (VJP) can be used:

$$\left(\frac{\partial \mathcal{P}_C(\hat{z})}{\partial \hat{z}}\right)^\top u, \quad u = \frac{\partial \mathcal{L}}{\partial \mathcal{P}_C(\hat{z})},$$

where u is the upstream gradient. As shown by Kotary et al., when the projection layer is defined by the fixed-point of an iterative solver, where $z_1 = \mathbf{z}^\star(\hat{z}_1) = \Phi(\mathbf{z}^\star, \hat{z}_1)$, this differentiation with respect to $\hat{z}_1$ becomes:

$$\frac{\partial z_1}{\partial \hat{z}_1} = \frac{\partial \Phi}{\partial \mathbf{z}^\star}\frac{\partial z_1}{\partial \hat{z}_1} + \frac{\partial \Phi}{\partial \hat{z}_1}$$

The system can be rearranged as:

$$\left(I - \frac{\partial \Phi}{\partial z_1^{(k)}}\right) \frac{\partial z_1}{\partial \hat{z}_1} = \frac{\partial \Phi}{\partial \hat{z}_1}$$

Then, by algebra, the VJP can be computed implicitly by:

$$\left(I - \frac{\partial \Phi}{\partial z_1^{(k)}}\right)^\top w = u$$

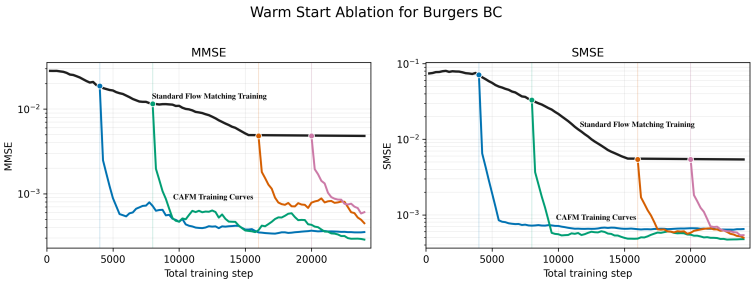
and obtain

$$\left(\frac{\partial \mathcal{P}_C(\hat{z})}{\partial \hat{z}}\right)^\top u = \left(\frac{\partial \Phi}{\partial \hat{z}_1}\right)^\top w,$$

where w is the solution to the linear system.

D Warm Starting Constraint-Aware Training

As mentioned in Section A, the CAFM objective can be warm started by initializing from pretrained flow matching model weights. While warm starting is not a fundamental requirement of the methodology, we view this as a practical adaptation of CAFM, reducing the training complexity introduced by unrolling the projection operator. This is particularly valuable, as including projections in the training forward pass typically results in a ~2.3× increase in runtime per training step on the PDE suite. The subsequent ablation tests how warm starting reduces the number of CAFM training steps needed to reach convergence.



	FFM	CAFM
Fwd.	111.6 ms	212.4 ms
Bwd.	12.8 ms	74.6 ms
Total	124.4 ms	287.0 ms

Figure 3: **Left:** Warm starting CAFM training from various points in standard flow matching training on Burgers BC. Additional warm starting steps yield faster convergence of the CAFM objective, with later transitions quickly converging to the same point as earlier transitions. **Right:** Table details average runtime for forward pass and backward pass operations, reporting the overall runtime impact of CAFM training.

Figure 3 provides empirical evidence that warm starting is useful, with runs adapted to the CAFM objective later in the standard flow matching training process converging more quickly to similar levels of MMSE and SMSE error. This provides strong practical evidence that training complexity can be mitigated through warm starting adaptations, and we defer further exploration of this behavior to subsequent studies.

E Rectified Flow Ablation

Section 5.2 motivates the instantaneous endpoint approximation as an efficient alternative to differentiating through the full forward ODE solve. In this section, we provide an ablation over the Burgers BC and Burgers IC tasks from the PDE benchmark, comparing three training approaches.

First, **1-RF** denotes the constraint-aware objective in Equation (10), which applies the projection directly to the instantaneous endpoint prediction

$$\hat{z}_1^{\text{inst}} = \mathbf{z}_t + (1-t) v_\theta(\mathbf{z}_t, t).$$

Second, **ODE Rollout** uses the original end-to-end objective, replacing this approximation with the integrated endpoint obtained by solving the learned dynamics from $t \to 1$ and backpropagating

Table 3: Ablation of the endpoint parameterization used for constraint-aware training. **1-RF** uses the instantaneous endpoint approximation, **ODE Rollout** differentiates through the integrated forward trajectory, and **2-RF** performs one additional ReFlow recoupling step before applying the instantaneous objective. Best mean and second-best are shown in **bold** and underlined, respectively.

Benchmark	Metric	1-RF	ODE Rollout	2-RF
Burgers BC	MMSE	$(1.32 \pm 1.34)\text{e}{-3}$	$(5.02 \pm 4.89)\text{e}{-3}$	$\mathbf{(6.12 \pm 3.55)e-4}$
	SMSE	$\mathbf{(3.00 \pm 2.13)e-4}$	$(1.19 \pm 0.86)\text{e}{-3}$	$(1.05 \pm 0.21)\text{e}{-3}$
	$\|\hat{z}_1^{\text{ODE}} - \hat{z}_1^{\text{inst}}\|$	$(2.75 \pm 1.61)\text{e}{-2}$	-	$(2.56 \pm 1.39)\text{e}{-2}$
Burgers IC	MMSE	$(4.05 \pm 0.66)\text{e}{-2}$	$\mathbf{(4.02 \pm 0.27)e-2}$	$(4.04 \pm 0.15)\text{e}{-2}$
	SMSE	$(5.52 \pm 0.54)\text{e}{-2}$	$(6.56 \pm 0.92)\text{e}{-2}$	$\mathbf{(2.99 \pm 0.24)e-2}$
	$\|\hat{z}_1^{\text{ODE}} - \hat{z}_1^{\text{inst}}\|$	$(3.55 \pm 0.84)\text{e}{-2}$	-	$(3.42 \pm 0.92)\text{e}{-2}$

through the resulting trajectory. Finally, **2-RF** applies one additional ReFlow step to update the endpoint coupling before training with the same instantaneous endpoint objective. For this experiment, the ReFlow coupling is constructed using a frozen PCFM teacher and a 100-step Euler solve. All reported results are means and sample standard deviations over three independently trained seeds.

The ablation results are reported in Table 3. Despite optimizing the exact integrated endpoint, the ODE rollout does not provide a consistent advantage over the instantaneous approximation; this behavior is expected, provided prior work's analysis of gradient instability in the backpropagation through the ODE solve (Liang et al., 2026; Wang et al., 2026). We additionally evaluate whether further straightening the learned transport through ReFlow improves the instantaneous approximation. The 2-RF variant provides substantial gains for selected metrics, obtaining the lowest Burgers BC MMSE and reducing Burgers IC SMSE from 5.52×10^{-2} to 2.99×10^{-2}. However, these improvements are not uniform, and 1-RF retains the best Burgers BC SMSE. This suggests that while further rectification can be useful in select cases, it is not a requirement for constraint-aware training.

F Unrolling Steps Ablation

Section 5.3 defines the differentiable projection operator as an iterative optimization layer. In particular, starting from the predicted clean endpoint $\mathbf{x}^{(0)}(\hat{\mathbf{c}}) = \hat{z}_1$, the projection is approximated by repeatedly applying an update map Φ,

$$\mathbf{x}^{(k+1)}(\hat{\mathbf{c}}) = \Phi(\mathbf{x}^{(k)}, \hat{\mathbf{c}}), \qquad k = 0, \dots, K-1,$$

so that the projected endpoint used by CAFM is given by

$$\mathcal{P}_C(\hat{z}_1) \approx \mathbf{x}^{(K)}(\hat{\mathbf{c}}).$$

The number of iterations K dictates the depth of the differentiable projection layer used during training. To study the sensitivity of CAFM to this approximation, we perform an ablation over the number of unrolling steps K in the fixed-grid wind setting.

Table 4: One-seed unrolling-steps ablation for constraint-aware fixed-grid wind sampling.

K	MMSE	Variance MSE	CV (Coherence)
1	5.515e-02	2.981e-01	1.65e-03
2	5.479e-02	3.109e-01	1.20e-04
4	5.472e-02	2.947e-01	**0.00e+00**
8	**5.288e-02**	**2.339e-01**	3.68e-04
16	5.435e-02	2.510e-01	4.24e-04
32	**5.286e-02**	2.413e-01	5.21e-04

For each value of K, we evaluate the downstream-selected checkpoint trained with that number of projection iterations. During inference, the augmented Lagrangian projection is run with a fixed

number of iterations, isolating the results to the training setup. The corresponding results are reported in Table 4.

The ablation suggests that only a few unrolled iterations are necessary for this setting. Increasing the number of projection-layer iterations from $K = 1$ to $K = 2$ sharply reduces coherence violation, and $K = 4$ reaches the training-data floor for this seed under the stronger inference projection. Beyond this point, constraint violation does not improve monotonically, suggesting that the feasibility benefit of additional unrolling steps has largely saturated. Larger values of K, however, preserve or slightly improve MMSE and substantially improve Variance MSE as $K \geq 8$.