LEARNING CONDITIONAL SOURCE DISTRIBUTION VIA FLOW REVERSAL FOR TEMPORAL FLOW MATCHING

Anonymous authors
Paper under double-blind review

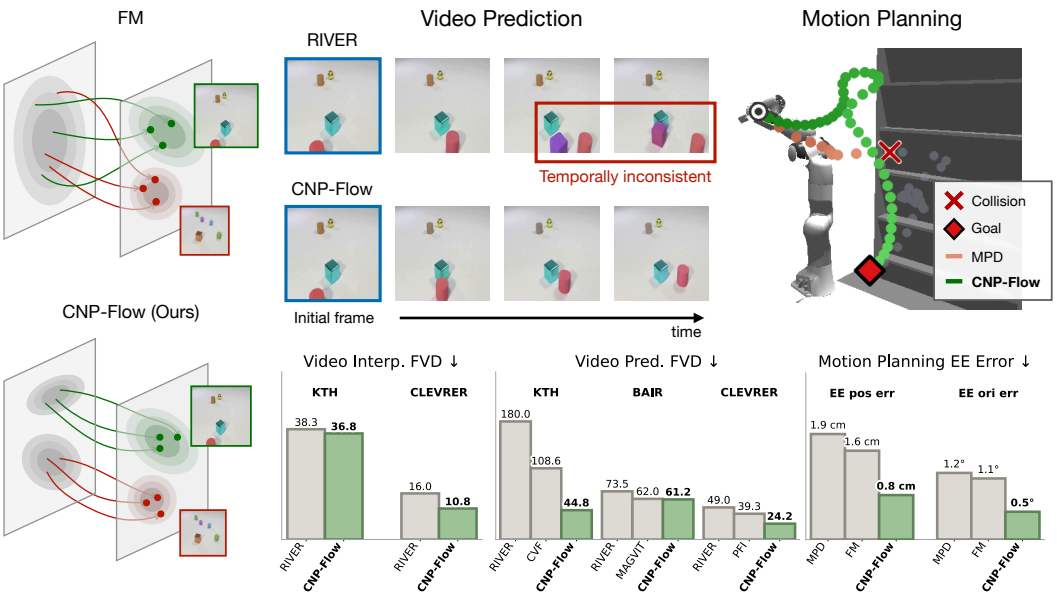


Figure 1: **CNP-Flow learns condition-dependent source distributions for temporal generation.** CNP-Flow uses a Conditional Noise Predictor (CNP) to predict a source distribution for each condition. This source-side conditioning reduces FVD by up to 75% over RIVER on video prediction, and yields more precise motion-planning trajectories with fewer collisions.

ABSTRACT

We introduce CNP-Flow, a flow matching framework for temporal generation that learns conditional source distributions through flow reversal. Whereas standard conditional flow matching (FM) incorporates conditioning through the vector field and draws source samples from a standard Gaussian, CNP-Flow uses a conditional noise predictor (CNP) to produce an isotropic Gaussian source for each temporal condition. The CNP is supervised by source samples obtained through flow reversal, which maps observed targets backward through a pretrained FM model. A three-stage pipeline pretrains the FM model, trains the CNP, and fine-tunes the FM model using the learned source distribution, while preserving the FM backbone architecture. Across video prediction, video interpolation, and 7-DoF Franka robot motion planning, CNP-Flow consistently improves generation quality. It also matches baseline performance with fewer function evaluations.

1 INTRODUCTION

Generative modeling is typically framed as drawing samples from a learned data distribution, where each generation is independent of the others. However, applications such as video generation (Ho et al., 2022; Ge et al., 2023) and robotic control (Janner et al., 2022; Carvalho et al., 2023) require a more challenging and constrained setup: they must generate a sequence of temporally correlated samples conditioned on past or future data. This defines the task of *temporal generation*, which imposes two requirements: the transitions between samples must follow those of the data distribution,

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.

and each sample must remain structurally consistent with the conditional inputs (Lai et al., 2018; Ge et al., 2023). Depending on the conditioning, temporal generation can be further divided into extrapolation and interpolation. Extrapolation predicts future samples given the past (Voleti et al., 2022; Davtyan et al., 2023), while interpolation generates intermediate trajectories given the start and end of a sequence (Janner et al., 2022; Carvalho et al., 2023; Reda et al., 2022).

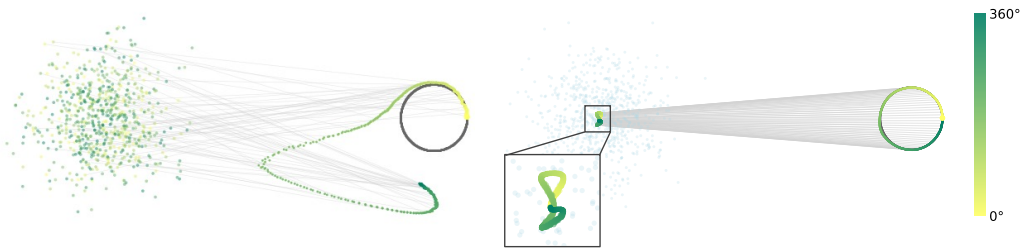
Flow Matching (FM) (Lipman et al., 2023) is an effective framework for generative modeling: a neural network regresses a deterministic vector field that transports samples from a source distribution to the target distribution. The source is typically chosen to be simple, such as a standard Gaussian. Thanks to this general formulation, FM models extend naturally to temporal generation by treating temporal conditions as additional inputs (Esser et al., 2024; Gkanatsios et al., 2025). While conceptually simple, their performance depends heavily on the quantity and quality of the conditions injected into the model (Peebles and Xie, 2023). Existing approaches, therefore, focus on improving the conditioning mechanism through various inductive biases: randomly masking conditional inputs as a form of regularization (Voleti et al., 2022; Davtyan et al., 2023) or biasing generated samples toward conditional data via gradient methods (Ho et al., 2022). However, the former sacrifices fidelity, while the latter cannot handle multiple conditions simultaneously.

These approaches all target the conditioning mechanism. A separate line of work, however, takes a different angle: modulating the source distribution itself. Recent studies show that FM models can be steered toward higher-quality outputs by selecting more suitable source samples (Pooladian et al., 2023; Tong et al., 2024; Chiang et al., 2026). A common approach in image editing (Meng et al., 2022; Song et al., 2021) replaces the standard Gaussian noise with a noised version of the original image, thereby preserving its spatial structure. In text-to-video generation (Ge et al., 2023) and video editing (Chang et al., 2024), drawing source samples in a directionally correlated manner across frames improves temporal coherence over drawing them independently per frame. Pushing this further, recent video prediction methods (Shrivastava and Shrivastava, 2024b;a) replace the standard Gaussian source with the previous frame. In robotic manipulation, action generation is largely determined by the source samples, so training a dedicated noise sampler enables adaptation of pretrained robot policies (Wagenmaker et al., 2025). Together, these results suggest that source samples themselves carry rich conditional information, which motivates us to ask: can a learned source distribution also benefit temporal generation?

We introduce **C**onditional **N**oise **P**redictor for **Flow** Matching (CNP-Flow), a flow matching framework for temporal generation. The core idea is to learn a noise predictor that takes the current condition as input and outputs a source distribution suited to downstream FM models under that condition. This formulation naturally encompasses both interpolation and extrapolation (see Fig. 1). To supervise the noise predictor, we construct pseudo-ground-truth source distributions by retrieving source samples for each condition through flow reversal of a pretrained FM model. A recent study (Chiang et al., 2026) shows that pretrained FM models map similar target data to nearby source samples through flow reversal. Motivated by this, we parameterize these distributions as a family of isotropic Gaussians, with the conditional noise predictor (CNP) outputting their mean and standard deviation. This sidesteps the exact search for optimal source distributions, which is intractable, while still providing effective surrogates that substantially improve temporal generation in our experiments.

CNP-Flow consists of a CNP and an FM model, trained via a three-stage pipeline that bootstraps each model from the other. First, we pretrain an FM model with a standard Gaussian source; flow reversal through the pretrained model then retrieves surrogate source samples for each condition. Next, we train the noise predictor by maximizing the likelihood of these source samples under each condition, enabling it to predict source distributions at test time. Finally, we fine-tune the FM model with the predicted distributions, refining its vector field. Fig. 2 illustrates this on a toy example: compared to a vanilla FM, CNP-Flow draws temporally correlated source samples and produces temporally coherent sequences.

CNP-Flow shows empirical gains along three independent axes. On video benchmarks, distribution-level alignment improves, with FVD on KTH dropping from 180.00 to 44.76 and comparable margins on CLEVRER and BAIR (see Fig. 1). Inference becomes far more efficient: with only 5 function evaluations, CNP-Flow surpasses what the standard Gaussian baseline achieves at 50. The same formulation transfers beyond video to a 7-DoF Franka motion planning task, where the fraction of valid trajectories rises from 84.9% to 88.5%, and end-effector position error drops from 1.6 cm to



(a) FM (Standard Gaussian) (b) CNP-Flow (Ours)

Figure 2: **2D toy example of temporal generation**. Colors indicate temporal phase along the target circular trajectory. (a) Vanilla FM draws source samples from a standard Gaussian. Samples associated with neighboring temporal conditions can be widely separated, and the generated sequence deviates from the target trajectory. (b) CNP-Flow learns a conditional source distribution with a smoother arrangement of samples by temporal phase (see the zoomed region), and the generated sequence follows the target trajectory more closely. Detailed experimental settings are provided in Appendix F.

0.8 cm (see Fig. 1). These results point to the same mechanism: conditioning the source improves source–target alignment and straightens the transport path, without modality-specific structure.

2 Related Work

Temporal conditions. Generating temporally coherent samples is a central challenge in video prediction and motion planning. Early approaches based on recurrent networks (Graves, 2012; Babaeizadeh et al., 2021) and autoregressive models (Vaswani et al., 2017; Le Moing et al., 2021; Ma et al., 2025) struggle with long-term temporal consistency or computational cost. Recently, diffusion (Ho et al., 2020; Dhariwal and Nichol, 2021; Song et al., 2021) and flow models (Lipman et al., 2023), driven by their success in high-fidelity image synthesis, have been extended to video generation, prediction (Ho et al., 2022; Voleti et al., 2022; Höppe et al., 2022; Yang et al., 2023; Davtyan et al., 2023) and motion planning (Janner et al., 2022; Carvalho et al., 2023; 2025), exploring the strategies to inject temporal conditions into the model. More specifically, Voleti et al. (2022) condition on all past and future frames but randomly mask them during training, enabling a single model to handle prediction, generation, and interpolation. Davtyan et al. (2023) instead condition on a reference frame and one past frame to reduce training cost. Janner et al. (2022) condition on the first frame and last frame to generate a feasible trajectory. Alternatively, Ho et al. (2022) bias the sampling trajectory toward conditioning frames via gradient-based reconstruction guidance, but the per-condition gradient pass scales poorly when multiple temporal conditions must be satisfied at once. Despite their differences, these methods all focus on how the condition enters the generation process, while assuming the source distribution is a fixed standard Gaussian.

Optimized noise prior. A separate line of work shapes the source distribution. In image generation, noise priors have been optimized through reward functions (Eyring et al., 2024), structured patterns (Samuel et al., 2024; Mao et al., 2024), pre-collected databases (Mao et al., 2023), or condition-dependent shifts (Chen et al., 2025). For video synthesis, Ge et al. (2023) introduce temporally correlated noise across frames, while Luo et al. (2023) and Zhang et al. (2024) model residual noise between consecutive frames. In video-to-video settings, Chang et al. (2024); Burgert et al. (2025); Liu and Vahdat (2025) warp the noise to align with the target motion. For video prediction, Shrivastava and Shrivastava (2024b) and Chen et al. (2024) bypass the standard Gaussian noise by modeling frame transitions as continuous interpolation. Despite their variety, these methods each impose a specific noise structure, such as temporal correlations, geometric warping, or frame-wise residuals, that either degrades over long horizons or sacrifices multimodality (Kim et al., 2026; Lee et al., 2025; Sadat et al., 2024). In robotic manipulation, Wagenmaker et al. (2025) train a dedicated noise sampler to adapt a pretrained diffusion policy to new tasks, though their approach operates in low-dimensional action spaces and its scalability to high-dimensional settings, such as video, remains unclear. Chiang et al. (2026) show that pretrained flow-matching models map similar targets to nearby source regions, but their focus is on image generation and optimal transport rather than temporal generation. To our knowledge, no prior work has learned a condition-dependent source distribution that generalizes across temporal generation tasks while preserving stochasticity.

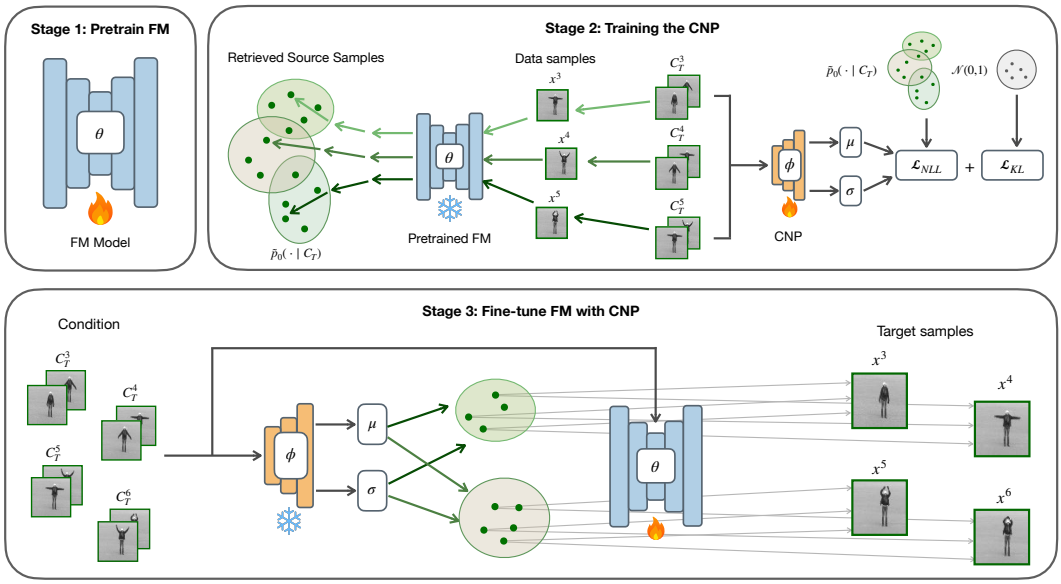


Figure 3: **Overview of the proposed three-stage training pipeline.** Stage 1: A pretrained FM is used to obtain flow-reversal-induced source samples, defining a source distribution aligned with the target data. Stage 2: The CNP is trained to model this distribution by predicting a conditional Gaussian prior via NLL and KL objectives. Stage 3: The FM model is fine-tuned using source samples drawn from the learned conditional prior, resulting in more coherent and target-aligned transport trajectories.

3 PRELIMINARY

Flow matching (FM) is a generative framework that learns to transport samples from a source distribution p_0 to a target distribution p_1. A neural network with parameters θ is trained to regress a time-varying vector field $\boldsymbol{v}_\theta$, which defines an ordinary differential equation (ODE): $\mathrm{d}\boldsymbol{x}_t = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)\,\mathrm{d}t$. Integrating this ODE from $t = 0$ to $t = 1$ transports a source sample $\boldsymbol{x}_0$ to a target sample $\boldsymbol{x}_1$. FM extends naturally to conditional generation by conditioning both the vector field $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C})$ and the target distribution $p_1(\cdot \mid \mathcal{C})$ on a conditioning input $\mathcal{C}$.

A standard training procedure pairs a source sample $\boldsymbol{x}_0$ with a target sample $\boldsymbol{x}_1$, samples a timestep $t \sim \mathcal{U}(0, 1)$, and constructs an intermediate sample by linear interpolation: $\boldsymbol{x}_t = (1 - t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$. The model is then optimized with the following objective (Lipman et al., 2023):

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{\mathcal{C},\boldsymbol{x}_0\sim p_0,\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}),t\sim\mathcal{U}(0,1)}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C}) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|_2^2\right]. \tag{1}$$

At inference time, generation proceeds by integrating the learned vector field from a source sample:

$$\hat{\boldsymbol{x}}_1 := \boldsymbol{x}_0 + \int_0^1 \boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C})\,\mathrm{d}t. \tag{2}$$

Eqn. 2 shows that the generated sample depends on both the condition $\mathcal{C}$ and the source sample $\boldsymbol{x}_0$. This observation motivates us to investigate whether more suitable source samples can improve generation quality under each condition.

4 METHOD

We propose CNP-Flow, a temporal generation framework that augments FM with a learned conditional source prior. Standard FM models draw source samples from a standard Gaussian and take the condition as an additional input. However, they often fail to generate high-quality data that faithfully follows the condition. CNP-Flow reinforces the conditioning mechanism by using a conditional noise predictor (CNP) that seeks a source distribution suitable for generating target data under each condition. Our design is general in two ways: (1) the noise predictor handles different types of conditioning, including both interpolation and extrapolation, allowing CNP-Flow to unify these tasks in a single framework; and (2) CNP-Flow only modulates the source distribution supplied to the FM model without changing its architecture, so it can be integrated into existing FM frameworks.

In the following, we describe the training and inference procedure of CNP-Flow in Sec. 4.1, and its application to downstream temporal generation tasks in Sec. 4.2.

4.1 Training and Inference

CNP-Flow is trained in three stages: we first pretrain the FM, then train the CNP from the pretrained FM, and finally fine-tune the FM under the learned CNP. An overview of the staged training pipeline is illustrated in Fig. 3. Generation itself is straightforward: CNP-Flow draws a source sample from the predicted source distribution and transforms it into a target sample via the learned vector field.

Stage 1: Pretraining the FM model. We pretrain the FM model in the standard manner for conditional generation, using a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as its source distribution. The model takes the temporal condition $\mathcal{C}_\mathcal{T}$ as an additional input and outputs a deterministic vector field $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C}_\mathcal{T})$. In temporal generation tasks, target samples under similar temporal conditions often share similar structures and dynamics. Since the learned ODE flow is continuous, flow reversal maps nearby targets to nearby regions in source space, yielding source samples that are clustered by the temporal condition rather than spread as a standard Gaussian. This suggests a feasible recipe: construct surrogate per-condition source distributions through flow reversal of the pretrained FM model, and train a model to predict them.

Stage 2: Training the conditional noise predictor. We train the CNP using source samples obtained through flow reversal of the pretrained FM model. For each temporal condition $\mathcal{C}_\mathcal{T}$, the CNP parameterizes an isotropic Gaussian source distribution:

$$p_\phi(\boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T}) := \mathcal{N}\left(\boldsymbol{x}_0; \boldsymbol{\mu}_\phi(\mathcal{C}_\mathcal{T}), \sigma_\phi^2(\mathcal{C}_\mathcal{T})\mathbf{I}\right), \tag{3}$$

where ϕ denotes the CNP parameters, $\boldsymbol{\mu}_\phi(\mathcal{C}_\mathcal{T})$ denotes the predicted mean, and $\sigma_\phi(\mathcal{C}_\mathcal{T})$ denotes the standard deviation. Given a target sample $\boldsymbol{x}_1 \sim p_1(\cdot \mid \mathcal{C}_\mathcal{T})$, we retrieve its corresponding source sample $\tilde{\boldsymbol{x}}_0$ through *flow reversal*:

$$\tilde{\boldsymbol{x}}_0 = \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T}) := \boldsymbol{x}_1 - \int_0^1 \boldsymbol{v}_\theta\left(\boldsymbol{x}_\tau, \tau \mid \mathcal{C}_\mathcal{T}\right) \mathrm{d}\tau, \tag{4}$$

This mapping induces the conditional source distribution $\tilde{p}_0(\cdot \mid \mathcal{C}_\mathcal{T})$. The CNP model is trained to minimize two losses: (1) a negative log-likelihood (NLL) loss: $\mathcal{L}_{\mathrm{NLL}}(\phi) = -\mathbb{E}_{\mathcal{C}_\mathcal{T}, \tilde{\boldsymbol{x}}_0 \sim \tilde{p}_0(\cdot \mid \mathcal{C}_\mathcal{T})}\left[\log p_\phi(\tilde{\boldsymbol{x}}_0 \mid \mathcal{C}_\mathcal{T})\right]$ that trains the CNP to approximate the flow-reversal-induced source distribution; and (2) a KL regularization loss: $\mathcal{L}_{\mathrm{KL}}(\phi) = \mathcal{D}_{\mathrm{KL}}(p_\phi(\cdot \mid \mathcal{C}_\mathcal{T}) \| \mathcal{N}(\mathbf{0}, \mathbf{I}))$ that regularizes the learned conditional source distribution toward the standard Gaussian prior. The total objective is:

$$\mathcal{L}_{\mathrm{CNP}}(\phi) = \mathcal{L}_{\mathrm{NLL}}(\phi) + \lambda \mathcal{L}_{\mathrm{KL}}(\phi), \tag{5}$$

where $\lambda > 0$ controls the strength of regularization. Our choice of loss objectives is justified by the following propositions. In Prop. 4.1, we show that, with the NLL loss, the optimal output mean and output variance of the CNP model coincide with the mean and the scaled second central moment of the source distribution induced by flow reversal.

Proposition 4.1 ((Informal) Population minimizer of NLL objective)**.** *Let the CNP model the isotropic Gaussians $p_\phi(\boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T}) = \mathcal{N}(\boldsymbol{x}_0; \boldsymbol{\mu}_\phi(\mathcal{C}_\mathcal{T}), \sigma_\phi^2(\mathcal{C}_\mathcal{T})\mathbf{I})$. Then the population minimizer of $\mathcal{L}_{\mathrm{NLL}}$ over the isotropic Gaussian family is $p_\phi^*(\boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T}) = \mathcal{N}(\boldsymbol{x}_0; \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T}), \tilde{\sigma}_0^2(\mathcal{C}_\mathcal{T})\mathbf{I})$, whose parameters satisfy*

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T}) &= \mathbb{E}_{\boldsymbol{x}_1 \sim p_1(\cdot \mid \mathcal{C}_\mathcal{T})}[\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})], \text{ and} \\ \tilde{\sigma}_0^2(\mathcal{C}_\mathcal{T}) &= \frac{1}{d}\mathbb{E}_{\boldsymbol{x}_1 \sim p_1(\cdot \mid \mathcal{C}_\mathcal{T})} \|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T}) - \tilde{\boldsymbol{\mu}}_0\|_2^2. \end{aligned} \tag{6}$$

The propositions covering more general cases, along with their proofs, are provided in Appendix B. In particular, Prop. B.1 establishes the results considering both the NLL and the KL divergence loss, while Cor. B.2 characterizes the limiting behavior of the distribution in extreme cases. We also summarize the entire training procedure of the CNP model in Algo. 1 in Appendix A.

Stage 3: Fine-tuning the FM model with predicted conditional source distributions. Lastly, we fine-tune the FM model with the per-condition source distribution predicted by the CNP model. This

allows us to refine its underlying vector field and further improve generation performance. Following the same training procedure as in the first stage, we replace the standard Gaussian with the predicted distributions for each condition.

$$\mathcal{L}_{\mathrm{FT-FM}}(\theta) = \mathbb{E}_{\mathcal{C}_\mathcal{T}, \boldsymbol{x}_0 \sim p_\phi(\cdot|\mathcal{C}_\mathcal{T}), \boldsymbol{x}_1 \sim p_1(\cdot|\mathcal{C}_\mathcal{T}), t \sim \mathcal{U}(0,1)} \left[\| \boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C}_\mathcal{T}) - (\boldsymbol{x}_1 - \boldsymbol{x}_0) \|_2^2 \right]. \quad (7)$$

We summarize the fine-tuning procedure in Algo. 2 in Appendix A.

Inference with CNP-Flow. During inference, we first draw source samples with the CNP model $\boldsymbol{x}_0 \sim p_\phi(\cdot \mid \mathcal{C}_\mathcal{T})$, and transform them into target data with the FM model (Eqn. 2).

Theorem 4.2 ((Informal) Conditional source locality)**.** *Suppose the predicted source mean $\boldsymbol{\mu}^*_\phi(\cdot)$ is L_μ-Lipschitz continuous in the temporal condition. For two conditions $\mathcal{C}_\mathcal{T}$ and $\mathcal{C}'_\mathcal{T}$, independently drawn CNP source samples satisfy, with high probability,*

$$\|\boldsymbol{x}_0 - \boldsymbol{x}'_0\|_2 \leq L_\mu \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2 + \mathcal{O}\left(\sqrt{d\big(\sigma^{*2}_\phi(\mathcal{C}_\mathcal{T}) + \sigma^{*2}_\phi(\mathcal{C}'_\mathcal{T})\big)}\right). \quad (8)$$

This theorem characterizes locality in the learned source space. Under a Lipschitz condition on the predicted mean, nearby temporal conditions yield nearby source centers, while the predicted variances control stochastic dispersion around these centers. The formal statement and proof are provided in Appendix B.

4.2 Application to Temporal Generation Tasks

We consider a series of temporal generation tasks that progressively generate sequential observations given temporal conditions. This formulation covers video prediction, video interpolation, and motion planning, which differ primarily in how the temporal condition is constructed.

Video Prediction. Given initial frames $\boldsymbol{x}^0$ and $\boldsymbol{x}^1$, the goal is to autoregressively predict future frames $\boldsymbol{x}^i$ for $i = 2, \ldots, N$. At step i, the preceding frame $\boldsymbol{x}^{i-1}$ serves as the reference frame, while a context frame $\boldsymbol{x}^c$ is sampled from earlier observations with $c \sim \mathcal{U}(\{0, 1, \ldots, i-2\})$, yielding the temporal condition $\mathcal{C}^i_\mathcal{T} = (\boldsymbol{x}^{i-1}, \boldsymbol{x}^c, i-c)$ that captures both recent dynamics and longer-range temporal structure.

Video Interpolation. Given boundary frames $\boldsymbol{x}^0$ and $\boldsymbol{x}^N$, the goal is to progressively predict intermediate frames $\boldsymbol{x}^i$ for $i = 1, \ldots, N-1$. At step i, the previously generated frame $\boldsymbol{x}^{i-1}$ serves as the reference, while $\boldsymbol{x}^N$ provides a fixed future context, giving $\mathcal{C}^i_\mathcal{T} = (\boldsymbol{x}^{i-1}, \boldsymbol{x}^N, N-i)$, where $N-i$ denotes the remaining temporal distance to the endpoint.

Motion Planning. Given an initial state $\boldsymbol{x}^0$ and a target end-effector pose $\boldsymbol{x}^N$, the objective is to generate a feasible trajectory that reaches the target pose from the initial state. The model generates the entire trajectory jointly, conditioned on $\mathcal{C}_\mathcal{T} = (\boldsymbol{x}^0, \boldsymbol{x}^N)$.

Detailed sampling procedures and algorithms are provided in Appendix A.

5 Experiments

5.1 Experimental Setup

We evaluate CNP-Flow across two modalities: *video generation*, comprising prediction and interpolation, where output quality emphasizes visual coherence; and *motion planning*, where the learned, condition-dependent source is expected to generalize from pixels to robot trajectories.

Video generation. We evaluate on BAIR Robot Pushing (Ebert et al., 2017) (64×64), KTH actions (Schüldt et al., 2004) (64×64), and CLEVRER (Yi et al., 2020) (128×128), spanning stochastic manipulation, structured human movement, and deterministic object dynamics. Following RIVER (Davtyan et al., 2023), we use the SRVP splits (Franceschi et al., 2020) and report PSNR, SSIM (Wang et al., 2004), and FVD (Unterthiner et al., 2019) under the SRVP/RIVER protocol. We define prediction as $p \to k$, where p observed frames are used to autoregressively generate k future frames. For interpolation, $p + f \to k$ denotes using p initial and f terminal observed frames to generate k intermediate frames; for brevity we write $2 \to 14$ for the two-endpoint case $1{+}1 \to 14$. Our primary baseline is RIVER, the fixed-source FM backbone we replace, providing

Table 1: **Results on video prediction.** CNP-Flow improves over the fixed-source RIVER baseline and remains competitive with recent source-learning methods. CNP-Flow results are averaged over three seeds.

Dataset	$p \to k$	Method	PSNR ↑	SSIM ↑	FVD ↓
BAIR64	$1 \to 15$	RIVER (Davtyan et al., 2023)	–	–	73.50
		MAGVIT (Yu et al., 2023)	19.30	0.787	62.00
		iVideoGPT (Wu et al., 2024)	**20.40**	0.823	75.00
		CAR-Flow (Chen et al., 2025)	20.08	0.836	71.37
		PYoCo (Ge et al., 2023)	20.13	<u>0.837</u>	<u>61.88</u>
		CSFM (Kim et al., 2026)	<u>20.16</u>	0.836	64.20
		CNP-Flow (Ours)	20.15	**0.839**	**61.17**
KTH	$10 \to 30$	RIVER (Davtyan et al., 2023)	30.40	0.860	180.00
		CVF (Shrivastava and Shrivastava, 2024a)	30.60	0.891	108.60
		CAR-Flow (Chen et al., 2025)	29.25	0.897	126.64
		PYoCo (Ge et al., 2023)	31.14	0.903	82.32
		CSFM (Kim et al., 2026)	<u>31.64</u>	<u>0.915</u>	<u>69.00</u>
		CNP-Flow (Ours)	**34.25**	**0.943**	**44.76**
KTH	$10 \to 40$	RIVER (Davtyan et al., 2023)	29.00	0.820	170.50
		CVF (Shrivastava and Shrivastava, 2024a)	<u>29.70</u>	<u>0.852</u>	<u>100.80</u>
		CNP-Flow (Ours)	**32.69**	**0.930**	**54.88**
CLEVRER	$2 \to 14$	RIVER (Davtyan et al., 2023)	–	–	48.96
		PFI (Chen et al., 2024)	–	–	<u>39.31</u>
		CNP-Flow (Ours)	**42.23**	**0.994**	**24.18**

Table 2: **Results on video interpolation.** CNP-Flow improves all metrics over RIVER on both datasets.

Dataset	$p+f \to k$	Method	PSNR ↑	SSIM ↑	FVD ↓
KTH	$2 \to 14$	RIVER (Davtyan et al., 2023)	35.65	0.9627	38.28
		CNP-Flow (Ours)	**36.07**	**0.9634**	**36.81**
CLEVRER	$2 \to 14$	RIVER (Davtyan et al., 2023)	43.79	0.9949	16.01
		CNP-Flow (Ours)	**45.68**	**0.9954**	**10.77**

a direct comparison between the standard Gaussian source and our learned, condition-dependent source. For other video-generation baselines, including MAGVIT (Yu et al., 2023), iVideoGPT (Wu et al., 2024), CVF (Shrivastava and Shrivastava, 2024a), and PFI (Chen et al., 2024), we report results from the original papers. We further compare with recent methods that modify or learn the source distribution, including CAR-Flow (Chen et al., 2025), PYoCo (Ge et al., 2023), and CSFM (Kim et al., 2026). Since directly comparable results are not available under our BAIR and KTH settings, we reimplement these methods following their original formulations and evaluate them under matched backbone, training-budget, and evaluation settings. Complete implementation and evaluation details are provided in Appendix C.

Motion planning. For cross-modality transfer, we evaluate on the 7-DoF Franka Panda benchmark from MPD (Carvalho et al., 2025): each sample is a joint-space trajectory conditioned on start and goal configurations. We compare CNP-Flow against MPD (a diffusion-based planner) and a fixed-source FM, all trained on identical trajectory data. Following MPD, we report success, valid fraction, end-effector position/orientation error, and collision intensity. FM-based methods use a 15-step Euler ODE sampler followed by MPD post-hoc cost refinement; details are provided in Appendix C.

5.2 Main results

Video generation. CNP-Flow achieves the best FVD and SSIM across all prediction settings (Tab. 1), consistently improving over the fixed-source RIVER backbone and the strongest competing baselines. The gain is particularly large on KTH $10 \to 30$, where FVD decreases from 180.00 with RIVER to 44.76 with CNP-Flow, compared with 126.64, 82.32, and 69.00 for CAR-Flow, PYoCo, and CSFM, respectively, while the gap is much smaller on BAIR. These methods either use predefined source structures or learn the source jointly with the flow using regularization or alignment. In contrast, CNP-Flow learns its conditional source from condition-specific source samples obtained through flow reversal, providing explicit supervision rather than relying on predefined structure or indirect joint optimization.

Table 3: **Results on motion planning.** All methods use a matched 15-step sampling budget, and results are averaged over three seeds.

Method	Success ↑	Valid (%) ↑	EE pos (cm) ↓	EE ori (°) ↓	Collision ↓
MPD (Carvalho et al., 2025)	**97%**	86.6	1.9	1.2	0.0030
Fixed-source FM	96%	84.9	1.6	1.1	0.0038
CNP-Flow (Ours)	**97%**	**88.5**	**0.8**	**0.5**	**0.0023**

The same trend holds for $2 \rightarrow 14$ interpolation (Tab. 2), with consistent gains on all metrics over RIVER. Interpolation gains are smaller in absolute terms, as both endpoints constrain the rollout and reduce the burden on the source distribution.

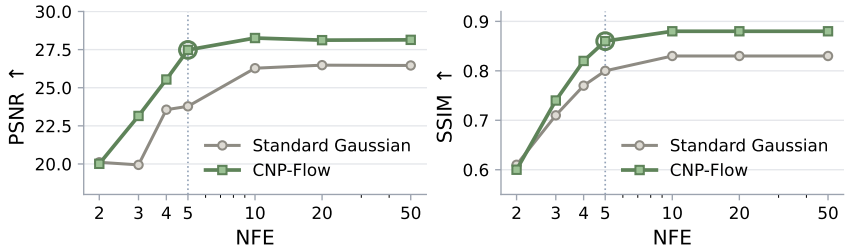


Figure 4: **Inference efficiency on KTH prediction.** Curves show PSNR/SSIM averaged across generated samples per video. CNP-Flow reaches the standard Gaussian PSNR plateau (≈ 26.5) at NFE $= 5$ and converges higher (≈ 28.2) by NFE $= 10$; SSIM shows a similar trend. The crossover point is highlighted on each panel.

Motion planning. The learned source is modality-agnostic: it parameterizes an isotropic Gaussian over latents conditioned on the temporal condition, regardless of what those latents represent. On the 7-DoF Franka Panda benchmark (Tab. 3), replacing diffusion (MPD) with a standard Gaussian source FM of similar capacity changes no metric outside noise. This rules out the generator family as the active variable. Adding the learned source to the same FM raises Valid from 84.9% to 88.5%, halves both end-effector errors, and cuts collision intensity by about 40%. Success rate moves only $96\% \rightarrow 97\%$ because this metric ("at least one valid trajectory in the batch") is already saturated. Valid counts the fraction of sampled trajectories that are feasible, which is precisely what a better source distribution should improve. The mechanism matches the video case: the learned source places the integration start aligned with the manifold of valid outputs. The modality, dimension, and failure mode (a robot collision rather than low SSIM) are different.

Inference efficiency. Because $\boldsymbol{\mu}_\phi(\mathcal{C}^i)$ is already aligned with the data manifold, the ODE no longer needs to first traverse the gap from a standard Gaussian before refining—each step is local refinement, so coarse solvers suffice. Fig. 4 substantiates this on the KTH prediction. Standard Gaussian source baseline plateaus at PSNR ≈ 26.5 from NFE $= 10$ onward; CNP-Flow attains and surpasses that plateau at NFE $= 5$ (PSNR 27.48, SSIM 0.86) and continues rising to PSNR ≈ 28.2 at NFE $= 10$. CNP-Flow at NFE $= 5$ already exceeds the baseline at NFE $= 50$, an order-of-magnitude reduction in solver budget; the final quality is superior, so the gain is both efficiency and endpoint improvement.

5.3 Analysis

Qualitative comparison. Fig. 5 shows the effect of the learned source. On CLEVRER interpolation, the fixed-source FM baseline incurs appearance drift and ghosting, while CNP-Flow maintains object identity and tracks the ground truth. On Franka Panda, MPD and fixed-source FM produce numerous colliding trajectories, whereas CNP-Flow produces a denser set of valid, collision-free plans. The visual failure modes reflect the metric results: a condition-dependent source improves temporal consistency in video and trajectory validity in motion planning.

Ablation studies. We ablate two design choices of CNP-Flow in Tab. 4: the parameterization of the learned source for video prediction, and whether the FM backbone should be fine-tuned after introducing the learned source. On KTH $10 \rightarrow 30$, replacing the standard Gaussian source with the deterministic learned mean $\boldsymbol{\mu}_\phi(\mathcal{C}^i)$ improves FVD from 180.00 to 83.82, showing that adapting the source to the temporal condition already provides a large gain over an unconditional source. However, the deterministic mean restricts each condition to a single source point. Adding the learned variance to form the full source $\boldsymbol{\mu}_\phi(\mathcal{C}^i) + \boldsymbol{\sigma}_\phi(\mathcal{C}^i)\boldsymbol{\epsilon}$ improves FVD to 44.76, PSNR from 31.46 to 34.25, and

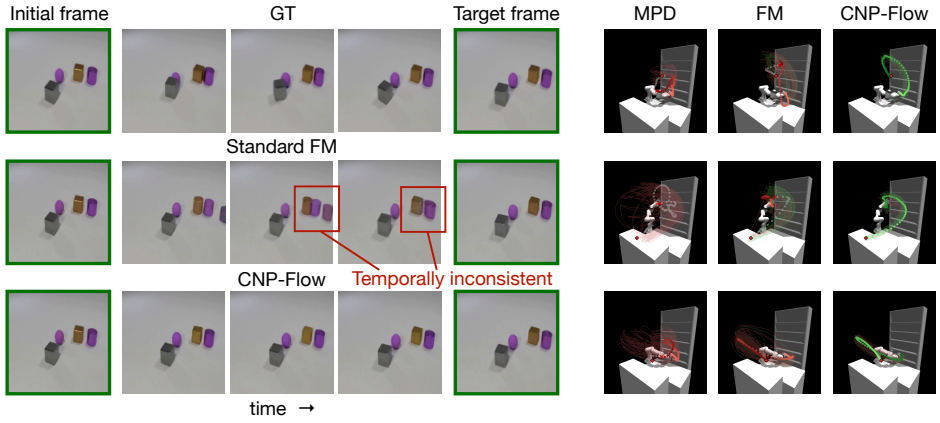


Figure 5: **Qualitative comparison.** *Left*: CLEVRER interpolation. Fixed-source FM drifts at the highlighted frames, while CNP-Flow matches the ground truth. *Right*: Franka Panda motion planning. Each panel overlays 50 sampled trajectories (red: colliding, green: collision-free). CNP-Flow produces more collision-free plans than MPD and fixed-source FM.

Table 4: **Ablation studies.** *Left*: source parameterization on KTH $10 \rightarrow 30$ video prediction, isolating the contribution of the learned mean and learned variance. *Right*: source learning and backbone fine-tuning (FT) on Franka Panda motion planning. In both tasks, metrics improve as the source better matches the condition.

Video prediction (KTH)			
Source $\boldsymbol{x}_0$	PSNR ↑	SSIM ↑	FVD ↓
$\mathcal{N}(\mathbf{0}, \mathbf{I})$	30.40	0.860	180.00
$\boldsymbol{\mu}_\phi(C^i)$	31.46	0.908	83.82
$\mathcal{N}(\boldsymbol{\mu}_\phi(C^i), \boldsymbol{\sigma}^2_\phi(C^i)\mathbf{I})$	**34.25**	**0.943**	**44.76**

Motion planning (Franka Panda)				
Variant	Valid (%) ↑	EE pos (cm) ↓	EE ori (°) ↓	Coll. ↓
Standard Gaussian	84.9	1.6	1.1	0.0038
CNP-Flow w/o FT	87.5	1.0	0.7	0.0024
CNP-Flow w/ FT	**88.5**	**0.8**	**0.5**	**0.0023**

SSIM from 0.908 to 0.943. This shows that stochasticity remains important for video prediction, as multiple plausible futures can exist under the same temporal conditions. Additional controls in Appendix E show that, with the FM frozen, a CNP without flow reversal supervision performs similarly to the standard Gaussian source, whereas flow reversal supervision substantially improves FVD, isolating its contribution to source learning.

For motion planning, the right side of Tab. 4 isolates two effects. Replacing the standard Gaussian source with a learned source (rows $1 \rightarrow 2$) yields the larger effect: Valid jumps from 84.9% to 87.5%, end-effector position error drops from 1.6 cm to 1.0 cm, and collision intensity drops by about 37%. All of this occurs without touching the FM backbone. Fine-tuning the backbone on the new source (rows $2 \rightarrow 3$) closes the remaining gap: Valid $87.5\% \rightarrow 88.5\%$, orientation $0.7° \rightarrow 0.5°$, collision $0.0024 \rightarrow 0.0023$. This suggests that re-aligning the velocity field with the learned, condition-dependent source distribution adds refinement.

Limitations and future work. While CNP-Flow improves generation quality and sampling efficiency across video prediction, video interpolation, and motion planning, one key limitation remains: the CNP uses a fixed form of temporal condition per task, limiting its use of richer contexts such as longer-horizon histories. A natural extension is to pair more expressive conditional source families with a flexible context encoder that absorbs longer-range, task-dependent structure. More broadly, treating the source distribution as a learnable conditioning channel is orthogonal to vector-field conditioning and applicable to any conditional FM task.

6 Conclusion

We presented CNP-Flow, a framework for learning conditional source distributions through flow reversal for temporal generation. By using a pretrained flow model to provide source supervision, CNP-Flow learns a source distribution adapted to each temporal condition while preserving the FM backbone architecture. Experiments across video prediction, video interpolation, and 7-DoF Franka motion planning demonstrate improved generation quality, alongside gains in sampling efficiency.

AI USE DISCLOSURE STATEMENT

Large language models were used to assist with language editing and improve the clarity of the manuscript. Claude Code was also used to support software development, including implementation and debugging. All AI-generated text and code were reviewed, revised as needed, and validated by the authors, who take full responsibility for the final manuscript and code.

REPRODUCIBILITY STATEMENT

We provide the details necessary to reproduce our results in the main paper and appendix. The complete training and inference procedures of CNP-Flow are provided in Appendix A, with additional mathematical details in Appendix B. Appendix C specifies the experimental setup, model architectures, training hyperparameters, flow reversal configurations, computational resources, and evaluation protocols for both video generation and motion planning. We also report results over multiple independent random seeds where applicable and provide additional analyses and generated results in Appendices E and G.

REFERENCES

Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. Fitvid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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)*, 2023.

Joao Carvalho, An T Le, Piotr Kicki, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and adapting robot motion planning with diffusion models. *IEEE Transactions on Robotics*, 2025.

Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C Azevedo. How i warped your noise: a temporally-correlated noise prior for diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

Chen Chen, Pengsheng Guo, Liangchen Song, Jiasen Lu, Rui Qian, Xinze Wang, Tsu-Jui Fu, Wei Liu, Yinfei Yang, and Alex Schwing. Car-flow: Condition-aware reparameterization aligns source and target for better flow matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael Samuel Albergo, Nicholas Matthew Boffi, and Eric Vanden-Eijnden. Probabilistic forecasting with stochastic interpolants and föllmer processes. In *International Conference on Machine Learning (ICML)*, 2024.

Chiensheng Chiang, Kuan-Hsun Tu, Jia-Wei Liao, Cheng-Fu Chou, and Tsung-Wei Ke. Cot-fm: Cluster-wise optimal transport flow matching. *arXiv preprint arXiv:2603.13395*, 2026.

Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient video prediction via sparsely conditioned flow matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *Conference on Robot Learning (CoRL)*, 2017.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *International Conference on Machine Learning (ICML)*, 2020.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Nikolaos Gkanatsios, Jiahe Xu, Matthew Bronars, Arsalan Mousavian, Tsung-Wei Ke, and Katerina Fragkiadaki. 3d flowmatch actor: Unified 3d policy for single- and dual-arm manipulation. In *arXiv preprint arXiv:2508.11002*, 2025.

Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 2012.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *Transactions on Machine Learning Research (TMLR)*, 2022.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning (ICML)*, 2022.

Junwan Kim, Jiho Park, Seonghu Jeon, and Seungryong Kim. Better source, better flow: Learning condition-dependent source distribution for flow matching. *arXiv preprint arXiv:2602.05951*, 2026.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Guillaume Le Moing, Jean Ponce, and Cordelia Schmid. Ccvs: Context-aware controllable video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Junho Lee, Kwanseok Kim, and Joonseok Lee. Is there a better source distribution than gaussian? exploring source distributions for image flow matching. *Transactions on Machine Learning Research (TMLR)*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Chao Liu and Arash Vahdat. Equivdm: Equivariant video diffusion models with temporally consistent noise. *arXiv preprint arXiv:2504.09789*, 2025.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research (TMLR)*, 2025.

Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Semantic-driven initial image construction for guided image synthesis in diffusion model. *arXiv preprint arXiv:2312.08872*, 2023.

Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. The lottery ticket hypothesis in denoising: Towards semantic-driven initialization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning (ICML)*, 2023.

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. In *International Conference on Learning Representations (ICLR)*, 2024.

Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2024.

Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, 2004.

Gaurav Shrivastava and Abhinav Shrivastava. Efficient continuous video flow model for video prediction. *arXiv preprint arXiv:2412.05633*, 2024a.

Gaurav Shrivastava and Abhinav Shrivastava. Video prediction by modeling videos as continuous multi-dimensional processes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research (TMLR)*, 2024.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2025.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Haotian Xue, Yipu Chen, Liqian Ma, Zelin Zhao, Lama Moukheiber, Yuchen Zhu, and Yongxin Chen. Acwm-phys: Investigating generalized physical interaction in action-conditioned video world models. *arXiv preprint arXiv:2605.08567*, 2026.

Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 2023.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Zhongwei Zhang, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Ting Yao, Yang Cao, and Tao Mei. Trip: Temporal residual learning with image noise prior for image-to-video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Appendix

A Algorithms of CNP-Flow

The training procedures of CNP-Flow are summarized in Algo. 1 and Algo. 2. The inference procedures for video prediction are provided in Algo. 3, while the inference procedures for video interpolation are provided in Algo. 4.

Algorithm 1 Training the CNP

Input: Temporal dataset $\mathcal{D}_{\mathcal{T}}$, pretrained FM θ, KL weight λ, learning rate γ

repeat
  Sample $(\boldsymbol{x}_1^i, \mathcal{C}_{\mathcal{T}}^i) \sim \mathcal{D}_{\mathcal{T}}$
  Flow reversal: $\tilde{\boldsymbol{x}}_0^i = \widetilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1^i \mid \mathcal{C}_{\mathcal{T}}^i)$
  $\mathcal{L}_{\text{NLL}}(\phi) = -\mathbb{E}[\log p_\phi(\tilde{\boldsymbol{x}}_0^i \mid \mathcal{C}_{\mathcal{T}}^i)]$
  $\mathcal{L}_{\text{KL}}(\phi) = \mathcal{D}_{\text{KL}}\left(p_\phi(\cdot \mid \mathcal{C}_{\mathcal{T}}^i) \| \mathcal{N}(\mathbf{0}, \mathbf{I})\right)$
  $\mathcal{L}_{\text{CNP}} = \mathcal{L}_{\text{NLL}} + \lambda \mathcal{L}_{\text{KL}}$
  $\phi \leftarrow \phi - \gamma \nabla_\phi \mathcal{L}_{\text{CNP}}$
until convergence

Return: Optimized CNP ϕ

Algorithm 2 Fine-Tuning FM

Input: Temporal dataset $\mathcal{D}_{\mathcal{T}}$, CNP ϕ, FM θ, learning rate γ

repeat
  Sample $(\boldsymbol{x}_1^i, \mathcal{C}_{\mathcal{T}}^i) \sim \mathcal{D}_{\mathcal{T}}$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\boldsymbol{x}_0^i = \boldsymbol{\mu}_\phi(\mathcal{C}_{\mathcal{T}}^i) + \sigma_\phi(\mathcal{C}_{\mathcal{T}}^i)\,\boldsymbol{\epsilon}$
  $t \sim \mathcal{U}(0,1)$; $\boldsymbol{x}_t^i = (1-t)\boldsymbol{x}_0^i + t\boldsymbol{x}_1^i$
  $\boldsymbol{v}_t^i = \boldsymbol{x}_1^i - \boldsymbol{x}_0^i$
  $\mathcal{L}_{\text{FT-FM}}(\theta) = \mathbb{E}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t^i, t \mid \mathcal{C}_{\mathcal{T}}^i) - \boldsymbol{v}_t^i\|_2^2\right]$
  $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_{\text{FT-FM}}$
until convergence

Return: Optimized FM θ

Algorithm 3 Video Prediction with CNP-Flow

Input: Initial $\boldsymbol{x}^0, \boldsymbol{x}^1$, CNP ϕ, FM θ, timesteps $0 = t_0 < \cdots < t_K = 1$

$\boldsymbol{x}_1^0 \leftarrow \boldsymbol{x}^0$, $\boldsymbol{x}_1^1 \leftarrow \boldsymbol{x}^1$
for $i = 2$ **to** N **do**
  VP condition: $\mathcal{C}_{\mathcal{T}}^i = (\boldsymbol{x}_1^{i-1}, \boldsymbol{x}_1^c, i-c)$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\boldsymbol{x}_0^i = \boldsymbol{\mu}_\phi(\mathcal{C}_{\mathcal{T}}^i) + \sigma_\phi(\mathcal{C}_{\mathcal{T}}^i)\,\boldsymbol{\epsilon}$
  for $k = 0$ **to** $K-1$ **do**
    $\Delta t_k = t_{k+1} - t_k$
    $\boldsymbol{x}_{t_{k+1}}^i = \boldsymbol{x}_{t_k}^i + \boldsymbol{v}_\theta(\boldsymbol{x}_{t_k}^i, t_k \mid \mathcal{C}_{\mathcal{T}}^i)\Delta t_k$
  end for
  $\boldsymbol{x}^i \leftarrow \boldsymbol{x}_1^i$
end for

Return: Temporal predictions $\{\boldsymbol{x}^i\}_{i=0}^N$

Algorithm 4 Video Interpolation with CNP-Flow

Input: Initial $\boldsymbol{x}^0, \boldsymbol{x}^N$, CNP ϕ, FM θ, timesteps $0 = t_0 < \cdots < t_K = 1$

$\boldsymbol{x}_1^0 \leftarrow \boldsymbol{x}^0$, $\boldsymbol{x}_1^N \leftarrow \boldsymbol{x}^N$
for $i = 1$ **to** $N-1$ **do**
  VI condition: $\mathcal{C}_{\mathcal{T}}^i = (\boldsymbol{x}_1^{i-1}, \boldsymbol{x}_1^N, N-i)$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\boldsymbol{x}_0^i = \boldsymbol{\mu}_\phi(\mathcal{C}_{\mathcal{T}}^i) + \sigma_\phi(\mathcal{C}_{\mathcal{T}}^i)\,\boldsymbol{\epsilon}$
  for $k = 0$ **to** $K-1$ **do**
    $\Delta t_k = t_{k+1} - t_k$
    $\boldsymbol{x}_{t_{k+1}}^i = \boldsymbol{x}_{t_k}^i + \boldsymbol{v}_\theta(\boldsymbol{x}_{t_k}^i, t_k \mid \mathcal{C}_{\mathcal{T}}^i)\Delta t_k$
  end for
  $\boldsymbol{x}^i \leftarrow \boldsymbol{x}_1^i$
end for

Return: Temporal predictions $\{\boldsymbol{x}^i\}_{i=0}^N$

B Mathematical Details

We consider the general form of the CNP objective and derive the equations satisfied by the corresponding optimal CNP parameters.

Proposition B.1 (Population minimizer of CNP objective)**.** *Consider the CNP objective $\mathcal{L}_{\text{CNP}}(\phi) = \alpha\mathcal{L}_{\text{NLL}}(\phi) + \beta\mathcal{L}_{\text{KL}}(\phi)$, with $\alpha > 0, \beta \geq 0$. Let the CNP family be the isotropic Gaussians $p_\phi(\boldsymbol{x}_0|\mathcal{C}_{\mathcal{T}}) = \mathcal{N}(\boldsymbol{x}_0; \boldsymbol{\mu}_\phi(\mathcal{C}_{\mathcal{T}}), \sigma_\phi^2(\mathcal{C}_{\mathcal{T}})\mathbf{I})$. For each condition $\mathcal{C}_{\mathcal{T}}$, assume the flow-reversal-induced source distribution has finite second moments, and denote its mean and variance by*

$$\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_{\mathcal{T}}) = \mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}_{\mathcal{T}})}[\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})], \quad \tilde{\sigma}_0^2(\mathcal{C}_{\mathcal{T}}) = \frac{1}{d}\mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}_{\mathcal{T}})}\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}}) - \tilde{\boldsymbol{\mu}}_0\|_2^2.$$

Then for each $\mathcal{C}_{\mathcal{T}}$, the population minimizer of $\mathcal{L}_{\text{CNP}}$ over the isotropic Gaussian family is

$$p_\phi^*(\boldsymbol{x}_0 \mid \mathcal{C}_{\mathcal{T}}) = \mathcal{N}(\boldsymbol{x}_0; \boldsymbol{\mu}_\phi^*(\mathcal{C}_{\mathcal{T}}), {\sigma_\phi^*}^2(\mathcal{C}_{\mathcal{T}})\mathbf{I}), \tag{9}$$

whose parameters $(\boldsymbol{\mu}_\phi^*, {\sigma_\phi^*}^2)$ satisfy

$$\begin{aligned} &\boldsymbol{\mu}_\phi^* = \frac{\alpha}{\alpha + \beta{\sigma_\phi^*}^2}\tilde{\boldsymbol{\mu}}_0, \text{ and} \\ &\beta({\sigma_\phi^*}^2)^2 + (\alpha-\beta){\sigma_\phi^*}^2 - \alpha\tilde{\sigma}_0^2 - \frac{\alpha\beta^2({\sigma_\phi^*}^2)^2}{d(\alpha+\beta{\sigma_\phi^*}^2)^2}\|\tilde{\boldsymbol{\mu}}_0\|_2^2 = 0. \end{aligned} \tag{10}$$

Proof. Given a condition $\mathcal{C}_\mathcal{T}$, let the flow-reversal-induced source distribution be $p_0(\cdot \mid \mathcal{C}_\mathcal{T}) = (\tilde{\Phi}_{1\to 0}(\cdot \mid \mathcal{C}_\mathcal{T}))_{\#} p_{\mathcal{D}_\mathcal{T}}$. The negative log-likelihood (NLL) of $p_\phi(\boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T})$ is

$$\mathcal{L}_{\mathrm{NLL}}(\phi) = -\mathbb{E}_{\boldsymbol{x}_0\sim p_0} \log \mathcal{N}(\boldsymbol{x}_0; \boldsymbol{\mu}_\phi, \sigma_\phi^2 \mathbf{I}) = \frac{d}{2}\log(2\pi\sigma_\phi^2) + \frac{1}{2\sigma_\phi^2}\mathbb{E}_{\boldsymbol{x}_0\sim p_0}\|\boldsymbol{x}_0 - \boldsymbol{\mu}_\phi\|_2^2.$$

By the second-moment decomposition $\mathbb{E}_{\boldsymbol{x}_0\sim p_0}\|\boldsymbol{x}_0 - \boldsymbol{\mu}_\phi\|_2^2 = d\tilde{\sigma}_0^2 + \|\boldsymbol{\mu}_\phi - \tilde{\boldsymbol{\mu}}_0\|_2^2$, we obtain

$$\mathcal{L}_{\mathrm{NLL}}(\phi) = \frac{d}{2}\log(2\pi\sigma_\phi^2) + \frac{1}{2\sigma_\phi^2}\left(d\tilde{\sigma}_0^2 + \|\boldsymbol{\mu}_\phi - \tilde{\boldsymbol{\mu}}_0\|_2^2\right).$$

Moreover, the KL term against the standard Gaussian has the closed form

$$\mathcal{L}_{\mathrm{KL}}(\phi) = \frac{1}{2}\left(\|\boldsymbol{\mu}_\phi\|_2^2 + d\sigma_\phi^2 - d - d\log\sigma_\phi^2\right).$$

Therefore, the CNP objective becomes

$$\mathcal{L}_{\mathrm{CNP}}(\phi) = \frac{\alpha}{2\sigma_\phi^2}\|\boldsymbol{\mu}_\phi - \tilde{\boldsymbol{\mu}}_0\|_2^2 + \frac{\beta}{2}\|\boldsymbol{\mu}_\phi\|_2^2 + \frac{\alpha d}{2}\log\sigma_\phi^2 + \frac{\alpha d\tilde{\sigma}_0^2}{2\sigma_\phi^2} + \frac{\beta d}{2}\sigma_\phi^2 - \frac{\beta d}{2}\log\sigma_\phi^2 + \frac{\alpha d}{2}\log 2\pi - \frac{\beta d}{2}.$$

At a stationary point $(\boldsymbol{\mu}_\phi^*, {\sigma_\phi^*}^2)$, the first-order optimality conditions satisfy

$$\nabla_\phi \mathcal{L}_{\mathrm{CNP}} = \begin{bmatrix} \nabla_{\boldsymbol{\mu}_\phi}\mathcal{L}_{\mathrm{CNP}}^\top(\boldsymbol{\mu}_\phi^*, {\sigma_\phi^*}^2) & \dfrac{\partial \mathcal{L}_{\mathrm{CNP}}}{\partial \sigma_\phi^2}(\boldsymbol{\mu}_\phi^*, {\sigma_\phi^*}^2) \end{bmatrix}^\top = \begin{bmatrix} \mathbf{0}_d^\top & 0 \end{bmatrix}^\top.$$

Differentiating $\mathcal{L}_{\mathrm{CNP}}$ with respect to $\boldsymbol{\mu}_\phi$ yields

$$\nabla_{\boldsymbol{\mu}_\phi}\mathcal{L}_{\mathrm{CNP}}(\boldsymbol{\mu}_\phi^*, {\sigma_\phi^*}^2) = \frac{\alpha}{\sigma_\phi^2}(\boldsymbol{\mu}_\phi^* - \tilde{\boldsymbol{\mu}}_0) + \beta\boldsymbol{\mu}_\phi^*.$$

Setting this gradient to zero gives

$$\boldsymbol{\mu}_\phi^* = \frac{\alpha}{\alpha + \beta{\sigma_\phi^*}^2}\tilde{\boldsymbol{\mu}}_0.$$

Substituting the optimal mean $\boldsymbol{\mu}_\phi^*$ into $\frac{\partial \mathcal{L}_{\mathrm{CNP}}}{\partial \sigma_\phi^2}(\boldsymbol{\mu}_\phi^*, {\sigma_\phi^*}^2) = 0$, we obtain

$$\beta({\sigma_\phi^*}^2)^2 + (\alpha - \beta){\sigma_\phi^*}^2 - \alpha\tilde{\sigma}_0^2 - \frac{\alpha\beta^2({\sigma_\phi^*}^2)^2}{d(\alpha + \beta{\sigma_\phi^*}^2)^2}\|\tilde{\boldsymbol{\mu}}_0\|_2^2 = 0.$$

The proof is complete. □

Next, we consider the limiting behavior of the optimal parameters when the two loss terms dominate under extreme coefficient settings.

Corollary B.2. *Under the setting of Prop. B.1, let $p_\phi^*(\boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T}) = \mathcal{N}(\boldsymbol{x}_0; \boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T}), {\sigma_\phi^*}^2(\mathcal{C}_\mathcal{T})\mathbf{I})$ be the population minimizer of $\mathcal{L}_{\mathrm{CNP}}$ over the isotropic Gaussian family. Then*

1. *If $\beta/\alpha \to 0^+$, then the population minimizer reduces to the moment-matching optimum,*

$$\boldsymbol{\mu}_\phi^* \to \tilde{\boldsymbol{\mu}}_0, \text{ and } {\sigma_\phi^*}^2 \to \tilde{\sigma}_0^2.$$

2. *If $\alpha/\beta \to 0^+$, then the population minimizer reduces to the standard Gaussian distribution,*

$$\boldsymbol{\mu}_\phi^* \to \mathbf{0}_d, \text{ and } {\sigma_\phi^*}^2 \to 1.$$

Proof. For the first limiting regime, let $\eta = \beta/\alpha$. Then

$$\boldsymbol{\mu}_\phi^* = \frac{1}{1+\eta{\sigma_\phi^*}^2}\tilde{\boldsymbol{\mu}}_0,$$

and the variance equation becomes

$$\eta({\sigma_\phi^*}^2)^2 + (1-\eta){\sigma_\phi^*}^2 - \tilde{\sigma}_0^2 - \frac{\eta^2({\sigma_\phi^*}^2)^2}{d(1+\eta{\sigma_\phi^*}^2)^2}\|\tilde{\boldsymbol{\mu}}_0\|_2^2 = 0.$$

Letting $\eta \to 0^+$, the nonlinear correction term vanishes, and the equation reduces to ${\sigma_\phi^*}^2 - \tilde{\sigma}_0^2 = 0$. Hence ${\sigma_\phi^*}^2 \to \tilde{\sigma}_0^2$ and consequently $\boldsymbol{\mu}_\phi^* \to \tilde{\boldsymbol{\mu}}_0$.

For the second limiting regime, let $\tau = \alpha/\beta$. Then

$$\boldsymbol{\mu}_\phi^* = \frac{\tau}{\tau+{\sigma_\phi^*}^2}\tilde{\boldsymbol{\mu}}_0,$$

and the variance equation becomes

$$({\sigma_\phi^*}^2)^2 + (\tau-1){\sigma_\phi^*}^2 - \tau\tilde{\sigma}_0^2 - \frac{\tau({\sigma_\phi^*}^2)^2}{d(\tau+{\sigma_\phi^*}^2)^2}\|\tilde{\boldsymbol{\mu}}_0\|_2^2 = 0.$$

Letting $\tau \to 0^+$, the equation reduces to $({\sigma_\phi^*}^2)^2 - {\sigma_\phi^*}^2 = 0$. Since ${\sigma_\phi^*}^2 > 0$, thus ${\sigma_\phi^*}^2 \to 1$ and $\boldsymbol{\mu}_\phi^* \to \mathbf{0}_d$. The proof is complete. □

Prop. B.1 shows that, after introducing the KL regularization term, the optimal mean and standard deviation become coupled and jointly influence each other. In particular, increasing the KL regularization strength pulls the mean toward zero while simultaneously constraining the variance toward the standard Gaussian prior. Cor. B.2 further analyzes the limiting regimes of this objective. When the KL coefficient dominates the NLL term, the learned distribution converges toward $\mathcal{N}(\mathbf{0}, \mathbf{I})$; conversely, when the KL regularization becomes negligible, the solution reduces to the setting characterized in Prop. 4.1 in the main paper.

Lemma B.3 (Lipschitz continuity of the conditional flow reversal map)**.** *Suppose that $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C}_\mathcal{T})$ is Lipschitz continuous in $(\boldsymbol{x}_t, \mathcal{C}_\mathcal{T})$ on $\mathcal{X}_t \times \mathcal{X}_{\mathcal{C}_\mathcal{T}}$, uniformly over $t \in [0,1]$. Then $\tilde{\Phi}_{\theta,1\to0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})$ is Lipschitz continuous on $\mathcal{X}_1 \times \mathcal{X}_{\mathcal{C}_\mathcal{T}}$.*

Proof. For any $\boldsymbol{x}_1, \boldsymbol{x}_1' \in \mathcal{X}_1$ and $\mathcal{C}_\mathcal{T}, \mathcal{C}_\mathcal{T}' \in \mathcal{X}_{\mathcal{C}_\mathcal{T}}$, since $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C}_\mathcal{T})$ is Lipschitz continuous on $\mathcal{X}_t \times \mathcal{X}_{\mathcal{C}_\mathcal{T}}$, there exist $L_{\boldsymbol{x}}, L_\mathcal{C} > 0$, independent of t, such that for all $t \in [0,1]$,

$$\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \mathcal{C}_\mathcal{T}) - \boldsymbol{v}_\theta(\boldsymbol{x}_t', t \mid \mathcal{C}_\mathcal{T}')\|_2 \le L_{\boldsymbol{x}}\|\boldsymbol{x}_t - \boldsymbol{x}_t'\|_2 + L_\mathcal{C}\|\mathcal{C}_\mathcal{T} - \mathcal{C}_\mathcal{T}'\|_2.$$

Define the reverse-time trajectories $\boldsymbol{z}_\tau := \boldsymbol{x}_{1-\tau}$ and $\boldsymbol{z}_\tau' := \boldsymbol{x}_{1-\tau}'$ for $\tau \in [0,1]$. By definition,

$$\boldsymbol{z}_1 = \boldsymbol{x}_0 = \tilde{\Phi}_{\theta,1\to0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T}) \quad \text{and} \quad \boldsymbol{z}_1' = \boldsymbol{x}_0' = \tilde{\Phi}_{\theta,1\to0}(\boldsymbol{x}_1' \mid \mathcal{C}_\mathcal{T}'),$$

and the reverse ODEs are given by

$$\frac{\mathrm{d}\boldsymbol{z}_\tau}{d\tau} = -\boldsymbol{v}_\theta(\boldsymbol{z}_\tau, 1-\tau \mid \mathcal{C}_\mathcal{T}) \quad \text{and} \quad \frac{\mathrm{d}\boldsymbol{z}_\tau'}{d\tau} = -\boldsymbol{v}_\theta(\boldsymbol{z}_\tau', 1-\tau \mid \mathcal{C}_\mathcal{T}').$$

It follows that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}\|\boldsymbol{z}_\tau - \boldsymbol{z}_\tau'\|_2 &\le \left\|\frac{\mathrm{d}\boldsymbol{z}_\tau}{\mathrm{d}\tau} - \frac{\mathrm{d}\boldsymbol{z}_\tau'}{\mathrm{d}\tau}\right\|_2 \\ &= \|-\boldsymbol{v}_\theta(\boldsymbol{z}_\tau, 1-\tau \mid \mathcal{C}_\mathcal{T}) - [-\boldsymbol{v}_\theta(\boldsymbol{z}_\tau', 1-\tau \mid \mathcal{C}_\mathcal{T}')]\|_2 \\ &\le L_{\boldsymbol{x}}\|\boldsymbol{z}_\tau - \boldsymbol{z}_\tau'\|_2 + L_\mathcal{C}\|\mathcal{C}_\mathcal{T} - \mathcal{C}_\mathcal{T}'\|_2.\end{aligned}$$

By Grönwall's inequality, we have

$$\begin{aligned}\|\boldsymbol{z}_1 - \boldsymbol{z}_1'\|_2 &\le \mathrm{e}^{L_{\boldsymbol{x}}}\|\boldsymbol{z}_0 - \boldsymbol{z}_0'\|_2 + \int_0^1 \mathrm{e}^{L_{\boldsymbol{x}}(1-\tau)} L_{\mathcal{C}}\|\mathcal{C}_{\mathcal{T}} - \mathcal{C}_{\mathcal{T}}'\|_2 \mathrm{d}\tau \\ &= \mathrm{e}^{L_{\boldsymbol{x}}}\|\boldsymbol{z}_0 - \boldsymbol{z}_0'\|_2 + L_{\mathcal{C}}\|\mathcal{C}_{\mathcal{T}} - \mathcal{C}_{\mathcal{T}}'\|_2 \int_0^1 \mathrm{e}^{L_{\boldsymbol{x}}(1-\tau)} \mathrm{d}\tau \\ &= \mathrm{e}^{L_{\boldsymbol{x}}}\|\boldsymbol{z}_0 - \boldsymbol{z}_0'\|_2 + L_{\mathcal{C}}\|\mathcal{C}_{\mathcal{T}} - \mathcal{C}_{\mathcal{T}}'\|_2 \left(\frac{e^{L_{\boldsymbol{x}}} - 1}{L_{\boldsymbol{x}}}\right) \\ &= \mathrm{e}^{L_{\boldsymbol{x}}}\|\boldsymbol{z}_0 - \boldsymbol{z}_0'\|_2 + \frac{L_{\mathcal{C}}}{L_{\boldsymbol{x}}}(e^{L_{\boldsymbol{x}}} - 1)\|\mathcal{C}_{\mathcal{T}} - \mathcal{C}_{\mathcal{T}}'\|_2.\end{aligned}$$

Since $\boldsymbol{z}_0 = \boldsymbol{x}_1$, $\boldsymbol{z}_0' = \boldsymbol{x}_1'$, $\boldsymbol{z}_1 = \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})$ and $\boldsymbol{z}_1' = \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1' \mid \mathcal{C}_{\mathcal{T}}')$, we obtain

$$\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}}) - \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1' \mid \mathcal{C}_{\mathcal{T}}')\right\|_2 \le L_\Phi\|\boldsymbol{x}_1 - \boldsymbol{x}_1'\|_2 + L_\Phi\|\mathcal{C}_{\mathcal{T}} - \mathcal{C}_{\mathcal{T}}'\|_2,$$

where $L_\Phi = \max\{\mathrm{e}^{L_{\boldsymbol{x}}}, \frac{L_{\mathcal{C}}}{L_{\boldsymbol{x}}}(\mathrm{e}^{L_{\boldsymbol{x}}} - 1)\}$. □

Proposition B.4 (Lipschitz continuity of the NLL objective population minimizer)**.** *Suppose that $\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})$ is Lipschitz continuous in $(\boldsymbol{x}_1, \mathcal{C}_{\mathcal{T}})$ on $\mathcal{X}_1 \times \mathcal{X}_{\mathcal{C}_{\mathcal{T}}}$, $\mathcal{C}_{\mathcal{T}} \mapsto p_1(\cdot \mid \mathcal{C}_{\mathcal{T}})$ is Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_{\mathcal{T}}}$ with respect to the W_2-distance, and there exists $M > 0$ such that $\mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot\mid\mathcal{C}_{\mathcal{T}})}\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})\|_2^2 \le M$ for all $\mathcal{C}_{\mathcal{T}} \in \mathcal{X}_{\mathcal{C}_{\mathcal{T}}}$. Then $\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_{\mathcal{T}})$ and $\tilde{\sigma}_0^2(\mathcal{C}_{\mathcal{T}})$ are Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_{\mathcal{T}}}$.*

Proof. **Step 1: Lipschitz continuity of the optimal mean.** For any $\mathcal{C}_{\mathcal{T}}', \mathcal{C}_{\mathcal{T}} \in \mathcal{X}_{\mathcal{C}_{\mathcal{T}}}$, since $\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})$ is Lipschitz continuous on $\mathcal{X}_1 \times \mathcal{X}_{\mathcal{C}_{\mathcal{T}}}$, there exists $L_\Phi > 0$ such that

$$\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1' \mid \mathcal{C}_{\mathcal{T}}') - \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})\right\|_2 \le L_\Phi\left(\|\boldsymbol{x}_1' - \boldsymbol{x}_1\|_2 + \|\mathcal{C}_{\mathcal{T}}' - \mathcal{C}_{\mathcal{T}}\|_2\right) \text{ for all } \boldsymbol{x}_1', \boldsymbol{x}_1 \in \mathcal{X}_1.$$

Moreover, since $\mathcal{C}_{\mathcal{T}} \mapsto p_1(\cdot \mid \mathcal{C}_{\mathcal{T}})$ is Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_{\mathcal{T}}}$ with respect to the W_2-distance, there exists $L_{p_1} > 0$ such that

$$W_2(p_1(\cdot \mid \mathcal{C}_{\mathcal{T}}'), p_1(\cdot \mid \mathcal{C}_{\mathcal{T}})) \le L_{p_1}\|\mathcal{C}_{\mathcal{T}}' - \mathcal{C}_{\mathcal{T}}\|_2.$$

Furthermore, there exists $\gamma_1 \in \Pi(p_1(\cdot \mid \mathcal{C}_{\mathcal{T}}'), p_1(\cdot \mid \mathcal{C}_{\mathcal{T}}))$ such that

$$\mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma_1}\|\boldsymbol{x}_1' - \boldsymbol{x}_1\|_2^2 \le \inf_{\gamma^*\in\Pi(p_1(\cdot\mid\mathcal{C}_{\mathcal{T}}'),p_1(\cdot\mid\mathcal{C}_{\mathcal{T}}))} \mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma^*}\|\boldsymbol{x}_1' - \boldsymbol{x}_1\|_2^2 + \|\mathcal{C}_{\mathcal{T}}' - \mathcal{C}_{\mathcal{T}}\|_2^2.$$

Then we obtain

$$\begin{aligned}\mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma_1}\|\boldsymbol{x}_1' - \boldsymbol{x}_1\|_2 &\le \left(\mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma_1}\|\boldsymbol{x}_1' - \boldsymbol{x}_1\|_2^2\right)^{\frac{1}{2}} \text{ (by Jensen's inequality)} \\ &\le W_2(p_1(\cdot \mid \mathcal{C}_{\mathcal{T}}'), p_1(\cdot \mid \mathcal{C}_{\mathcal{T}})) + \|\mathcal{C}_{\mathcal{T}}' - \mathcal{C}_{\mathcal{T}}\|_2 \le (L_{p_1} + 1)\|\mathcal{C}_{\mathcal{T}}' - \mathcal{C}_{\mathcal{T}}\|_2.\end{aligned}$$

Therefore, we have

$$\begin{aligned}\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_{\mathcal{T}}') - \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_{\mathcal{T}})\|_2 &= \left\|\mathbb{E}_{\boldsymbol{x}_1'\sim p_1(\cdot\mid\mathcal{C}_{\mathcal{T}}')}[\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1' \mid \mathcal{C}_{\mathcal{T}}')] - \mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot\mid\mathcal{C}_{\mathcal{T}})}[\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})]\right\|_2 \\ &= \left\|\mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma_1}[\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1' \mid \mathcal{C}_{\mathcal{T}}')] - \mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma_1}[\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})]\right\|_2 \\ &\le \mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma_1}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1' \mid \mathcal{C}_{\mathcal{T}}') - \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}})\right\|_2 \text{ (by the triangle inequality)} \\ &\le L_\Phi\left[\mathbb{E}_{(\boldsymbol{x}_1',\boldsymbol{x}_1)\sim\gamma_1}\|\boldsymbol{x}_1' - \boldsymbol{x}_1\|_2 + \|\mathcal{C}_{\mathcal{T}}' - \mathcal{C}_{\mathcal{T}}\|_2\right] \\ &\le L_{\tilde{\boldsymbol{\mu}}_0}\|\mathcal{C}_{\mathcal{T}}' - \mathcal{C}_{\mathcal{T}}\|_2.\end{aligned}$$

where $L_{\tilde{\boldsymbol{\mu}}_0} := L_\Phi(L_{p_1} + 2)$.

Step 2: Lipschitz continuity of the optimal variance. Recall that

$$\tilde{\sigma}_0^2(\mathcal{C}_{\mathcal{T}}) = \frac{1}{d}\mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot\mid\mathcal{C}_{\mathcal{T}})}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_{\mathcal{T}}) - \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_{\mathcal{T}})\right\|_2^2.$$

Equivalently,

$$\tilde{\sigma}_0^2(\mathcal{C}_\mathcal{T}) = \frac{1}{d}\left[\mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}_\mathcal{T})}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\right\|_2^2 - \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2^2\right].$$

For any $\mathcal{C}'_\mathcal{T}, \mathcal{C}_\mathcal{T} \in \mathcal{X}_{\mathcal{C}_\mathcal{T}}$, choose a coupling $\gamma_2 \in \Pi(p_1(\cdot \mid \mathcal{C}'_\mathcal{T}), p_1(\cdot \mid \mathcal{C}_\mathcal{T}))$ such that

$$\mathbb{E}_{(\boldsymbol{x}'_1,\boldsymbol{x}_1)\sim\gamma_2}\|\boldsymbol{x}'_1 - \boldsymbol{x}_1\|_2^2 \le W_2^2\left(p_1(\cdot \mid \mathcal{C}'_\mathcal{T}), p_1(\cdot \mid \mathcal{C}_\mathcal{T})\right) + \|\mathcal{C}'_\mathcal{T} - \mathcal{C}_\mathcal{T}\|_2^2 \le (L_{p_1}^2 + 1)\|\mathcal{C}'_\mathcal{T} - \mathcal{C}_\mathcal{T}\|_2^2.$$

Then we obtain

$$\begin{aligned}
&\left|\mathbb{E}_{\boldsymbol{x}'_1\sim p_1(\cdot|\mathcal{C}'_\mathcal{T})}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}'_1 \mid \mathcal{C}'_\mathcal{T})\right\|_2^2 - \mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}_\mathcal{T})}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\right\|_2^2\right| \\
=&\left|\mathbb{E}_{(\boldsymbol{x}'_1,\boldsymbol{x}_1)\sim\gamma_2}\left[\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}'_1 \mid \mathcal{C}'_\mathcal{T})\right\|_2^2 - \left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\right\|_2^2\right]\right| \\
\le&\mathbb{E}_{(\boldsymbol{x}'_1,\boldsymbol{x}_1)\sim\gamma_2}\Big[\left(\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}'_1 \mid \mathcal{C}'_\mathcal{T})\right\|_2 + \left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\right\|_2\right)\cdot\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}'_1 \mid \mathcal{C}'_\mathcal{T}) - \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\right\|_2\Big] \\
\le&2\sqrt{M}\cdot\left(\mathbb{E}_{(\boldsymbol{x}'_1,\boldsymbol{x}_1)\sim\gamma_2}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}'_1 \mid \mathcal{C}'_\mathcal{T}) - \tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\right\|_2^2\right)^{1/2} \text{ (by the Cauchy–Schwarz inequality)} \\
\le&2L_\Phi\sqrt{2M}\left(\mathbb{E}_{(\boldsymbol{x}'_1,\boldsymbol{x}_1)\sim\gamma_2}\|\boldsymbol{x}'_1 - \boldsymbol{x}_1\|_2^2 + \|\mathcal{C}'_\mathcal{T} - \mathcal{C}_\mathcal{T}\|_2^2\right)^{\frac{1}{2}} \text{ (since } \tilde{\Phi}_{\theta,1\to 0} \text{ is Lipschitz continuous).} \\
\le&2L_\Phi\sqrt{2M(L_{p_1}^2+2)}\|\mathcal{C}'_\mathcal{T} - \mathcal{C}_\mathcal{T}\|_2.
\end{aligned}$$

Next, for any $\mathcal{C}_\mathcal{T} \in \mathcal{X}_{\mathcal{C}_\mathcal{T}}$,

$$\begin{aligned}
\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2 &= \|\mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}_\mathcal{T})}[\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})]\|_2 \\
&\le \left(\mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}_\mathcal{T})}\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\|_2^2\right)^{\frac{1}{2}} \text{ (by Jensen's inequality)} \\
&\le \sqrt{M}.
\end{aligned}$$

Hence

$$\begin{aligned}
\left|\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T})\|_2^2 - \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2^2\right| &\le (\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T})\|_2 + \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2)\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T}) - \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2 \\
&\le 2\sqrt{M}\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T}) - \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2 \le 2L_\Phi(L_{p_1}+2)\sqrt{M}\|\mathcal{C}'_\mathcal{T} - \mathcal{C}_\mathcal{T}\|_2.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\left|\tilde{\sigma}_0^2(\mathcal{C}'_\mathcal{T}) - \tilde{\sigma}_0^2(\mathcal{C}_\mathcal{T})\right| \le& \frac{1}{d}\left|\mathbb{E}_{\boldsymbol{x}'_1\sim p_1(\cdot|\mathcal{C}'_\mathcal{T})}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}'_1 \mid \mathcal{C}'_\mathcal{T})\right\|_2^2 - \mathbb{E}_{\boldsymbol{x}_1\sim p_1(\cdot|\mathcal{C}_\mathcal{T})}\left\|\tilde{\Phi}_{\theta,1\to 0}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T})\right\|_2^2\right| \\
&+ \frac{1}{d}\left|\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T})\|_2^2 - \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2^2\right| \le L_{\tilde{\sigma}_0^2}\|\mathcal{C}'_\mathcal{T} - \mathcal{C}_\mathcal{T}\|_2.
\end{aligned}$$

where $L_{\tilde{\sigma}_0^2} := \frac{2L_\Phi}{d}\left[\sqrt{2M(L_{p_1}^2+2)} + (L_{p_1}+2)\sqrt{M}\right]$. The proof is complete. □

Proposition B.5 (Lipschitz continuity of the CNP objective population minimizer)**.** *Consider the CNP objective $\mathcal{L}_{\mathrm{CNP}}(\phi) = \alpha\mathcal{L}_{\mathrm{NLL}}(\phi) + \beta\mathcal{L}_{\mathrm{KL}}(\phi), \alpha > 0, \beta \ge 0$. Let $s(\mathcal{C}_\mathcal{T}) = \sigma_\phi^{*2}(\mathcal{C}_\mathcal{T})$ denote the unique positive solution of $F(s, \mathcal{C}_\mathcal{T}) = 0$, where*

$$F(s, \mathcal{C}_\mathcal{T}) = \beta s^2 + (\alpha - \beta)s - \alpha\tilde{\sigma}_0^2(\mathcal{C}_\mathcal{T}) - \frac{\alpha\beta^2 s^2}{d(\alpha + \beta s)^2}\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|^2.$$

Assume that for all $\mathcal{C}_\mathcal{T} \in \mathcal{X}_{\mathcal{C}_\mathcal{T}}$,

(1) $\|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2 \le M_{\tilde{\boldsymbol{\mu}}_0}$,

(2) $\tilde{\boldsymbol{\mu}}_0(\cdot)$ is $L_{\tilde{\boldsymbol{\mu}}_0}$-Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_\mathcal{T}}$,

(3) $\tilde{\sigma}_0^2(\cdot)$ is $L_{\tilde{\sigma}_0^2}$-Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_\mathcal{T}}$,

(4) $0 < s_{\min} \le s(\mathcal{C}_\mathcal{T}) \le s_{\max} < \infty$*, and*

(5) there exists $\kappa > 0$ *such that* $|\partial_s F(s, \mathcal{C}_\mathcal{T})| \ge \kappa$ *for all* $s \in [s_{\min}, s_{\max}]$.

Then $\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T})$ *and* $\sigma_\phi^{*2}(\mathcal{C}_\mathcal{T})$ *are Lipschitz continuous on* $\mathcal{X}_{\mathcal{C}_\mathcal{T}}$.

Proof. For any two conditions $\mathcal{C}_\mathcal{T}, \mathcal{C}'_\mathcal{T} \in \mathcal{X}_{\mathcal{C}_\mathcal{T}}$, we have

$$F(s(\mathcal{C}_\mathcal{T}), \mathcal{C}_\mathcal{T}) = F(s(\mathcal{C}'_\mathcal{T}), \mathcal{C}'_\mathcal{T}) = 0.$$

Therefore,

$$F(s(\mathcal{C}_\mathcal{T}), \mathcal{C}_\mathcal{T}) - F(s(\mathcal{C}'_\mathcal{T}), \mathcal{C}_\mathcal{T}) = F(s(\mathcal{C}'_\mathcal{T}), \mathcal{C}'_\mathcal{T}) - F(s(\mathcal{C}'_\mathcal{T}), \mathcal{C}_\mathcal{T}).$$

By the Mean Value Theorem, there exists $\bar{s}$ between $s(\mathcal{C}_\mathcal{T})$ and $s(\mathcal{C}'_\mathcal{T})$ such that

$$F(s(\mathcal{C}_\mathcal{T}), \mathcal{C}_\mathcal{T}) - F(s(\mathcal{C}'_\mathcal{T}), \mathcal{C}_\mathcal{T}) = \partial_s F(\bar{s}, \mathcal{C}_\mathcal{T})(s(\mathcal{C}_\mathcal{T}) - s(\mathcal{C}'_\mathcal{T})).$$

Since $s(\mathcal{C}_\mathcal{T})$ and $s(\mathcal{C}'_\mathcal{T})$ both lie in $[s_{\min}, s_{\max}]$, we have $\bar{s} \in [s_{\min}, s_{\max}]$. Thus,

$$\kappa |s(\mathcal{C}_\mathcal{T}) - s(\mathcal{C}'_\mathcal{T})| \le |F(s(\mathcal{C}'_\mathcal{T}), \mathcal{C}'_\mathcal{T}) - F(s(\mathcal{C}'_\mathcal{T}), \mathcal{C}_\mathcal{T})|.$$

For any $s \in [s_{\min}, s_{\max}]$, we have

$$\begin{aligned}
|F(s, \mathcal{C}_\mathcal{T}) - F(s, \mathcal{C}'_\mathcal{T})| &\le \alpha |\tilde{\sigma}_0^2(\mathcal{C}_\mathcal{T}) - \tilde{\sigma}_0^2(\mathcal{C}'_\mathcal{T})| + \frac{\alpha\beta^2 s^2}{d(\alpha + \beta s)^2} \left| \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T})\|_2^2 - \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T})\|_2^2 \right| \\
&\le \alpha L_{\tilde{\sigma}_0^2} \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2 + \frac{\beta^2 s_{\max}^2}{d\alpha} (2 M_{\tilde{\boldsymbol{\mu}}_0} L_{\tilde{\boldsymbol{\mu}}_0}) \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2 \\
&= \left(\alpha L_{\tilde{\sigma}_0^2} + \frac{2\beta^2 s_{\max}^2 M_{\tilde{\boldsymbol{\mu}}_0} L_{\tilde{\boldsymbol{\mu}}_0}}{d\alpha} \right) \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2.
\end{aligned}$$

Hence $|s(\mathcal{C}_\mathcal{T}) - s(\mathcal{C}'_\mathcal{T})| \le L_s \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2$, where $L_s = \frac{1}{\kappa} \left(\alpha L_{\tilde{\sigma}_0^2} + \frac{2\beta^2 s_{\max}^2 M_{\tilde{\boldsymbol{\mu}}_0} L_{\tilde{\boldsymbol{\mu}}_0}}{d\alpha} \right)$. Therefore, $\sigma_\phi^{*2}(\mathcal{C}_\mathcal{T})$ is Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_\mathcal{T}}$. Next, by Prop. B.1, the optimal mean satisfies

$$\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T}) = \frac{\alpha}{\alpha + \beta s(\mathcal{C}_\mathcal{T})} \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T}).$$

Therefore,

$$\begin{aligned}
\|\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T}) - \boldsymbol{\mu}_\phi^*(\mathcal{C}'_\mathcal{T})\|_2 &= \left\| \frac{\alpha}{\alpha + \beta s(\mathcal{C}_\mathcal{T})} \tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T}) - \frac{\alpha}{\alpha + \beta s(\mathcal{C}'_\mathcal{T})} \tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T}) \right\|_2 \\
&\le \frac{\alpha}{\alpha + \beta s(\mathcal{C}_\mathcal{T})} \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}_\mathcal{T}) - \tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T})\|_2 + \alpha \left| \frac{1}{\alpha + \beta s(\mathcal{C}_\mathcal{T})} - \frac{1}{\alpha + \beta s(\mathcal{C}'_\mathcal{T})} \right| \|\tilde{\boldsymbol{\mu}}_0(\mathcal{C}'_\mathcal{T})\|_2.
\end{aligned}$$

Moreover,

$$\begin{aligned}
\left| \frac{1}{\alpha + \beta s(\mathcal{C}_\mathcal{T})} - \frac{1}{\alpha + \beta s(\mathcal{C}'_\mathcal{T})} \right| &= \frac{\beta |s(\mathcal{C}_\mathcal{T}) - s(\mathcal{C}'_\mathcal{T})|}{(\alpha + \beta s(\mathcal{C}_\mathcal{T}))(\alpha + \beta s(\mathcal{C}'_\mathcal{T}))} \\
&\le \frac{\beta}{\alpha^2} |s(\mathcal{C}_\mathcal{T}) - s(\mathcal{C}'_\mathcal{T})| \le \frac{\beta L_s}{\alpha^2} \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2.
\end{aligned}$$

Hence $\|\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T}) - \boldsymbol{\mu}_\phi^*(\mathcal{C}'_\mathcal{T})\|_2 \le L_{\boldsymbol{\mu}_\phi^*} \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2$, where $L_{\boldsymbol{\mu}_\phi^*} = L_{\tilde{\boldsymbol{\mu}}_0} + \frac{\beta L_s M_{\tilde{\boldsymbol{\mu}}_0}}{\alpha}$. Therefore, $\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T})$ is Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_\mathcal{T}}$. The proof is complete. □

Lemma B.6 (High-probability two-sided bound for Gaussian sample distance)**.** *Let* $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_d), \boldsymbol{x}' \sim \mathcal{N}(\boldsymbol{\mu}', \sigma'^2 \mathbf{I}_d)$ *be independent random vectors in* $\mathbb{R}^d$ *with* $\sigma > 0$ *and* $\sigma' > 0$*. Then, for any* $\delta \in (0, 1)$*, with probability at least* $1 - \delta$*,*

$$\|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_2 - \sqrt{\sigma^2 + \sigma'^2} c_{d,\delta} \le \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \le \|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_2 + \sqrt{\sigma^2 + \sigma'^2} c_{d,\delta},$$

where $c_{d,\delta} = \sqrt{d + 2\sqrt{d \log \frac{1}{\delta}} + 2 \log \frac{1}{\delta}}$.

Proof. Since $\boldsymbol{x}$ and $\boldsymbol{x}'$ are independent Gaussian random vectors,

$$\boldsymbol{x} - \boldsymbol{x}' \sim \mathcal{N}(\boldsymbol{\mu} - \boldsymbol{\mu}', (\sigma^2 + \sigma'^2)\mathbf{I}_d).$$

Therefore, we obtain

$$\left\| \frac{(\boldsymbol{x} - \boldsymbol{x}') - (\boldsymbol{\mu} - \boldsymbol{\mu}')}{\sqrt{\sigma^2 + \sigma'^2}} \right\|_2^2 \sim \chi^2(d).$$

By Laurent–Massart bounds of chi-square distribution with degrees of freedom d,

$$\mathbb{P}\left(\left\| \frac{(\boldsymbol{x} - \boldsymbol{x}') - (\boldsymbol{\mu} - \boldsymbol{\mu}')}{\sqrt{\sigma^2 + \sigma'^2}} \right\|_2^2 \leq d + 2\sqrt{d\tau} + 2\tau \right) \geq 1 - \mathrm{e}^{-\tau} \text{ for all } \tau > 0.$$

Setting $\tau = \log \frac{1}{\delta} > 0$ and $c_{d,\delta} = \sqrt{d + 2\sqrt{d \log \frac{1}{\delta}} + 2 \log \frac{1}{\delta}}$, then we have

$$\mathbb{P}\left(\|(\boldsymbol{x} - \boldsymbol{x}') - (\boldsymbol{\mu} - \boldsymbol{\mu}')\|_2 \leq c_{d,\delta} \sqrt{\sigma^2 + \sigma'^2} \right) \geq 1 - \delta.$$

Hence, with probability at least $1 - \delta$, by the reverse triangle inequality,

$$|\|\boldsymbol{x} - \boldsymbol{x}'\|_2 - \|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_2| \leq \|(\boldsymbol{x} - \boldsymbol{x}') - (\boldsymbol{\mu} - \boldsymbol{\mu}')\|_2 \leq \sqrt{\sigma^2 + \sigma'^2} c_{d,\delta}.$$

Therefore, we have

$$\|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_2 - \sqrt{\sigma^2 + \sigma'^2} c_{d,\delta} \leq \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \leq \|\boldsymbol{\mu} - \boldsymbol{\mu}'\|_2 + \sqrt{\sigma^2 + \sigma'^2} c_{d,\delta}.$$

□

Lemma B.6 establishes that the distance between two independent Gaussian samples can be bounded in terms of the distance between their means and the scale of their standard deviations. Rmk. B.7 separately examines the implications of this bound for standard Gaussian samples and conditional noise prediction (CNP).

Remark B.7. Given the condition $\mathcal{C}_\mathcal{T}, \mathcal{C}'_\mathcal{T} \in \mathcal{X}_{\mathcal{C}_\mathcal{T}}$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$,

1. Standard Gaussian: If $\boldsymbol{x}_0^{\mathrm{SG}}, \boldsymbol{x}_0^{\mathrm{SG}'} \overset{\mathrm{iid}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, then
$$\|\boldsymbol{x}_0^{\mathrm{SG}} - \boldsymbol{x}_0^{\mathrm{SG}'}\|_2 = \mathcal{O}(\sqrt{2} c_{d,\delta}).$$
2. CNP: If $\boldsymbol{x}_0^{\mathrm{CNP}} \sim \mathcal{N}(\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T}), \sigma_\phi^{*2}(\mathcal{C}_\mathcal{T})\mathbf{I}_d)$ and $\boldsymbol{x}_0^{\mathrm{CNP}'} \sim \mathcal{N}(\boldsymbol{\mu}_\phi^*(\mathcal{C}'_\mathcal{T}), \sigma_\phi^{*2}(\mathcal{C}'_\mathcal{T})\mathbf{I}_d)$ are drawn independently, then
$$\|\boldsymbol{x}_0^{\mathrm{CNP}} - \boldsymbol{x}_0^{\mathrm{CNP}'}\|_2 = \|\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T}) - \boldsymbol{\mu}_\phi^*(\mathcal{C}'_\mathcal{T})\|_2 \pm \mathcal{O}\left(\sqrt{\sigma_\phi^{*2}(\mathcal{C}_\mathcal{T}) + \sigma_\phi^{*2}(\mathcal{C}'_\mathcal{T})} c_{d,\delta} \right).$$

Theorem B.8. *Suppose $\boldsymbol{\mu}_\phi^*(\cdot)$ is $L_{\boldsymbol{\mu}_\phi^*}$-Lipschitz continuous on $\mathcal{X}_{\mathcal{C}_\mathcal{T}}$. Given $\mathcal{C}_\mathcal{T}, \mathcal{C}'_\mathcal{T} \in \mathcal{X}_{\mathcal{C}_\mathcal{T}}$, if $\boldsymbol{x}_0^{\mathrm{CNP}} \sim \mathcal{N}(\boldsymbol{\mu}_\phi^*(\mathcal{C}_\mathcal{T}), \sigma_\phi^{*2}(\mathcal{C}_\mathcal{T})\mathbf{I}_d)$ and $\boldsymbol{x}_0^{\mathrm{CNP}'} \sim \mathcal{N}(\boldsymbol{\mu}_\phi^*(\mathcal{C}'_\mathcal{T})), \sigma_\phi^{*2}(\mathcal{C}'_\mathcal{T})\mathbf{I}_d)$, then with probability at least $1 - \delta$,*

$$\|\boldsymbol{x}_0^{\mathrm{CNP}} - \boldsymbol{x}_0^{\mathrm{CNP}'}\|_2 \leq L_{\boldsymbol{\mu}_\phi^*} \|\mathcal{C}_\mathcal{T} - \mathcal{C}'_\mathcal{T}\|_2 + \sqrt{\sigma_\phi^{*2}(\mathcal{C}_\mathcal{T}) + \sigma_\phi^{*2}(\mathcal{C}'_\mathcal{T})} c_{d,\delta}.$$

Rmk. B.7 reveals that the discrepancy between two CNP samples can be decomposed into two components: a deterministic shift induced by the difference between their conditions, and a stochastic deviation induced by the predicted uncertainty. In particular, Theorem B.8 shows that when the conditional mean predictor is Lipschitz continuous, nearby conditions lead to nearby predicted means. Therefore, the sample-level discrepancy remains controlled with high probability as long as the conditions are close and the predicted variances are small.

Table 5: **Inference overhead of CNP-Flow.** Latency is measured over 1,000 generated frames under matched hardware, batch-size, and solver settings.

Dataset	Fixed-source FM	CNP-Flow	Latency increase	Additional memory
KTH / BAIR64	207.1 ms/frame	214.2 ms/frame	3.43%	∼15 MB
CLEVRER	211.9 ms/frame	219.1 ms/frame	3.39%	

Proposition B.9. *Given the condition $\mathcal{C}_\mathcal{T}$, let $\boldsymbol{x}_0^{\mathrm{SG}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{x}_0^{\mathrm{CNP}} \sim p_\phi^*(\cdot \mid \mathcal{C}_\mathcal{T})$, then*

$$\mathrm{trace}(\mathrm{Cov}(\boldsymbol{x}_1 - \boldsymbol{x}_0^{\mathrm{CNP}} \mid \mathcal{C}_\mathcal{T})) - \mathrm{trace}(\mathrm{Cov}(\boldsymbol{x}_1 - \boldsymbol{x}_0^{\mathrm{SG}} \mid \mathcal{C}_\mathcal{T})) = d(\sigma_\phi^2(\mathcal{C}_\mathcal{T}) - 1).$$

Proof. Since $\boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T}$ and $\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T}$ are independent, we have

$$\mathrm{Cov}(\boldsymbol{x}_1 - \boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T}) = \mathrm{Cov}(\boldsymbol{x}_1 \mid \mathcal{C}_\mathcal{T}) + \mathrm{Cov}(\boldsymbol{x}_0 \mid \mathcal{C}_\mathcal{T}).$$

Moreover,

$$\mathrm{Cov}(\boldsymbol{x}_0^{\mathrm{CNP}} \mid \mathcal{C}_\mathcal{T}) = \sigma_\phi^{*2}(\mathcal{C}_\mathcal{T})\mathbf{I}_d, \quad \mathrm{Cov}(\boldsymbol{x}_0^{\mathrm{SG}} \mid \mathcal{C}_\mathcal{T}) = \mathbf{I}_d$$

Therefore, the proof is conducted. □

Prop. B.9 shows that replacing the standard Gaussian source with the CNP source directly changes the variance of the FM target velocity. In particular, when the learned CNP variance satisfies $\sigma_\phi^2(\mathcal{C}_\mathcal{T}) < 1$, which typically holds when the temporal condition strongly constrains the target sample, the covariance trace of the target velocity is reduced compared to standard Gaussian initialization. This indicates that CNP provides a more concentrated source distribution, yielding lower-variance velocity targets.

C Experimental Details

C.1 Video Generation

Compute environment. Video prediction and interpolation experiments are conducted on the AMD AI & HPC Clusters Taiwan, equipped with AMD Instinct MI300X GPUs with 192GB HBM3 memory and AMD EPYC 9684X 96-Core processors. GPUs within each node are interconnected through AMD Infinity Fabric (XGMI). Flow reversal and final evaluation are run on a single 8-GPU MI300X node, while CNP training and FM fine-tuning use a single MI300X GPU. All AMD experiments use ROCm 7.2.0. Flow matching pretraining and fine-tuning use BF16 mixed precision.

Training time. The wall-clock runtime varies with dataset size and video resolution. For KTH, BAIR64, and CLEVRER, the additional offline cost of the CNP-Flow pipeline is approximately 5.5, 9.1, and 17 hours, respectively, relative to vanilla fixed-source FM. On CLEVRER, our largest-resolution video benchmark, source retrieval via flow reversal takes approximately 3 hours on a single 8-GPU MI300X node, while CNP training and FM fine-tuning take approximately 14 hours on a single MI300X GPU. Including final evaluation, each video dataset/task completes within 24 hours on the specified hardware.

Computational overhead. To quantify the additional cost of CNP-Flow at generation time, we measure inference latency, peak memory, and model FLOPs under same hardware, batch-size, and ODE-solver settings. As summarized in Tab. 5, CNP-Flow increases inference latency by only about 3.4%. The CNP is evaluated once per generated frame and requires 0.297 GFLOPs; at NFE=10, this corresponds to approximately 0.34% additional model FLOPs, with about 15 MB additional peak memory. The 98.08M-parameter FM backbone is augmented with a 47.69M-parameter CNP (50.84M on CLEVRER). Source samples are retrieved offline via flow reversal and cached, and flow reversal is never invoked during generation.

Backbone and latent space. All video experiments are performed in the VQ-GAN latent space released by RIVER (Davtyan et al., 2023). The latent resolution is 8×8 for BAIR64 and KTH, and 16×16 for CLEVRER at 128×128. For video prediction, we reuse the released RIVER prediction

Table 6: Video prediction fixed-source FM and flow reversal configuration. Source samples obtained through flow reversal are cached only for CNP supervision and are not used at inference time.

Component	KTH	CLEVRER	BAIR64
Flow reversal steps	100	20	50
FM fine-tune length	200k steps	20k steps	50k steps
FM fine-tune batch	64	128	64
FM fine-tune LR	5×10^{-5}	5×10^{-5}	5×10^{-5}

Table 7: Video interpolation fixed-source FM and flow reversal configuration. Since no compatible interpolation checkpoint is available, the fixed-source CFM backbone is trained from scratch.

Component	KTH	CLEVRER
FM pretrain steps	300k	300k
FM pretrain batch	32	16
Flow reversal steps	100	40
FM fine-tune steps	200k	200k
FM fine-tune batch	32	16
FM fine-tune LR	1×10^{-4}	1×10^{-4}

checkpoints for KTH and CLEVRER, while retraining the BAIR64 fixed-source FM backbone due to checkpoint incompatibility with our evaluation setup. For video interpolation, since no compatible pretrained interpolation checkpoints are provided, we train the fixed-source FM backbone from scratch. Detailed FM fine-tuning and flow reversal configurations are given in Tab. 6 and 7.

Source samples obtained through flow reversal are cached only for CNP supervision and are not used at inference time. FM fine-tuning uses Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay, and gradient clipping with maximum norm 1.0.

CNP architecture and training. The video CNP is a convolutional encoder over latent feature grids. It predicts the mean and log-variance of an isotropic Gaussian source distribution. For prediction, the CNP conditions on the available context latents and temporal-distance embedding. For interpolation, the CNP conditions on the local reference latent, the endpoint latent, and the temporal-distance embedding. The shared architecture is summarized in Tab. 8. All video CNPs are trained with Adam using a learning rate of 1×10^{-4} and a KL weight of $\lambda_{\mathrm{KL}} = 1\times10^{-4}$. Task-specific training lengths, batch sizes, and KL warm-up schedules are listed in Tab. 9.

Source-learning baselines. For CAR-Flow (Chen et al., 2025), we use its source-only variant, which modifies the source while leaving the target unchanged. For CSFM (Kim et al., 2026), we adapt its source predictor to the same temporal condition used by the RIVER backbone and retain its original regularization and source–target alignment objectives. All source-learning baselines use the same FM backbone, training data, pretrained FM initialization, fine-tuning budget, sampling NFE, and evaluation protocol as CNP-Flow. Method-specific hyperparameters follow the original recommendations or are selected on validation data without using test-set performance.

Evaluation. For both video prediction and video interpolation, we evaluate 256 videos and generate 100 samples per video. PSNR and SSIM are reported as best-of-100 scores, while FVD is computed over the full generated sample pool using the same I3D feature extractor as RIVER (Davtyan et al., 2023). Unless otherwise stated, all main-text results in Tab. 1 and 2 use a 10-step Euler solver (NFE=10); Fig. 4 sweeps NFE separately to characterize inference efficiency.

Variability across runs. To evaluate run-to-run variability, we run the complete pipeline with three independent random seeds for each setting and report mean $\pm$ standard deviation. Tab. 10 provides the complete three-seed results for both video prediction and interpolation across all evaluated datasets.

Table 8: Video CNP architecture. The same architecture is used for video prediction and video interpolation.

Component	Setting
Backbone	Convolutional encoder
Input	Conditional latents + temporal-distance embedding
Output	$\mu, \log \sigma^2$
Base channels	64
Channel multipliers	$(1, 2, 4, 8)$
Residual blocks per resolution	2
Attention resolution	Middle latent resolution
Dropout	0.1
Latent grid	8×8 or 16×16

Table 9: Video CNP training configuration. All video CNPs are trained with Adam using learning rate 1×10^{-4} and KL weight $\lambda_{\mathrm{KL}} = 1\times10^{-4}$.

Task	Dataset	Length	Batch	KL warm-up
Prediction	KTH	50 epochs	64	10 epochs
Prediction	CLEVRER	15 epochs	1024	10 epochs
Prediction	BAIR64	15 epochs	64	5 epochs
Interpolation	KTH	200k steps	32	5 epochs
Interpolation	CLEVRER	200k steps	16	2 epochs

C.2 Motion Planning

Compute environment. Motion planning experiments are conducted on a workstation equipped with NVIDIA RTX 6000 Ada generation GPUs with 48GB memory. Each training, flow reversal, and evaluation run uses a single GPU. We build on the public MPD implementation (Carvalho et al., 2025) with PyTorch and CUDA. Training uses EMA weights and BF16 automatic mixed precision, while flow reversal and evaluation are performed in FP32.

Training time. The complete CNP-Flow pipeline completes within 12 hours on this workstation: fixed-source FM pretraining (~ 4 hours), source retrieval and caching via flow reversal (~ 1 hour, run once), CNP training (~ 2 hours), and FM fine-tuning (~ 4 hours), with the remaining time spent on final evaluation. The cost of source retrieval and caching is amortized across all subsequent CNP epochs, and the overall pipeline takes roughly $3\times$ the training time of a vanilla fixed-source FM.

Task and trajectory representation. We evaluate on the Panda Warehouse benchmark from MPD (Carvalho et al., 2025), which considers 7-DoF Franka Panda joint-space planning with end-effector pose goals. We use the same demonstration dataset, robot model, collision environment, and evaluation protocol as MPD. Trajectories are parameterized as cubic B-splines with 22 desired control points along the trajectory, together with 2 phantom control points used to enforce boundary conditions, giving 24 learnable control points in total. The spline is evaluated at 128 time samples over a 10-second horizon.

Backbone architecture. The diffusion baseline and the fixed-source flow matching backbone share the same temporal U-Net and conditioning architecture. The trajectory backbone predicts 24 learnable B-spline control points. The U-Net uses a base channel size of 32 with channel multipliers $(1, 2, 4, 8)$. The start state and goal pose are encoded by separate 2-layer MLPs, and both are projected into a 128-dimensional context embedding.

Motion planning CNP. The motion planning CNP is a residual MLP over vectorized B-spline trajectory control points. It predicts the mean and log-variance of an isotropic Gaussian source distribution over the learnable control-point noise. The architecture is summarized in Tab. 11. We train the CNP for 2000 epochs with batch size 512, learning rate 1×10^{-4}, and KL weight $\lambda_{\mathrm{KL}} = 1\times10^{-6}$. The KL weight is linearly warmed up for the first 50 epochs. Optimization uses

Table 10: **Variability across runs for CNP-Flow.** Results are reported as mean ± standard deviation over three independent runs for video prediction and interpolation.

Dataset	PSNR ↑	SSIM ↑	FVD ↓
Video prediction			
BAIR64 $1 \rightarrow 15$	20.15 ± 0.13	0.8390 ± 0.0021	61.17 ± 1.58
KTH $10 \rightarrow 30$	34.25 ± 0.26	0.9432 ± 0.0025	44.76 ± 1.53
KTH $10 \rightarrow 40$	32.69 ± 0.22	0.9303 ± 0.0040	54.88 ± 3.85
CLEVRER $2 \rightarrow 14$	42.23 ± 0.19	0.9935 ± 0.0002	24.18 ± 2.46
Video interpolation			
KTH	36.07 ± 0.54	0.9634 ± 0.0021	36.81 ± 2.65
CLEVRER	45.68 ± 0.01	0.9954 ± 0.0001	10.77 ± 0.02

Table 11: **Motion planning CNP architecture.** The CNP predicts an isotropic Gaussian source distribution over the learnable B-spline control-point noise.

Component	Setting
Backbone	Residual MLP
Input	Start state + EE goal pose + context embedding
Raw condition dimension	26
Context embedding dimension	128
Output	$\mu, \log \sigma^2$
Hidden dimension	1024
Residual blocks	6
Activation	GELU
Output initialization	Zero-initialized head
EMA decay	0.995

Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay, gradient clipping with maximum norm 1.0, and FP32 precision.

Training configuration. For the diffusion baseline, we use the released MPD checkpoint. The fixed-source flow matching model is pretrained for 6.0×10^5 steps with batch size 512, learning rate 3×10^{-4}, EMA weights, and BF16 automatic mixed precision. Source samples obtained through flow reversal are then cached once for CNP supervision using 200 Euler steps with batch size 1024 in FP32; these cached targets are not used during inference. After CNP training, we freeze the CNP and fine-tune the flow matching backbone for 200k steps using CNP-sampled source noise instead of the standard Gaussian source, with batch size 512, learning rate 1.5×10^{-4}, EMA weights, and BF16 automatic mixed precision.

Evaluation. For evaluation, we follow the MPD inference protocol. All methods are evaluated on the same 100 test start-goal pairs, and results are averaged over three independent seeds. For each pair, we generate $N = 50$ candidate trajectories and select the shortest valid trajectory. The MPD baseline uses its cost-guided DDIM sampler, which interleaves denoising with cost-gradient updates: we follow the paper and use 15 DDIM steps, activate guidance during the final 20% of the denoising process. Specifically, the last 3 denoising steps each perform $n_{\text{guide}} = 4$ cost-gradient updates, resulting in 12 gradient updates in total.

For CNP-Flow, we use a 15-step Euler ODE sampler followed by 12 post-hoc cost-gradient refinement steps, using the same cost terms and weights as MPD. Thus, both methods use 15 generative steps and 12 cost-gradient updates.

Specifically, after ODE sampling, we run 12 refinement steps with learning rate 0.01, maximum perturbation 0.15, and value clipping at 1.0, using the same cost terms and weights as MPD. This keeps the cost-refinement stage matched to the released MPD configuration, while replacing the diffusion prior with our learned-source flow matching sampler. Flow reversal is used only to construct CNP supervision and is not invoked during sampling or evaluation. We report the same evaluation metrics as MPD: *Success*, *Valid fraction*, end-effector position/orientation error, and *collision intensity*.

Table 12: **Motion-planning performance without post-hoc refinement.** Results are reported as mean ± standard deviation over three independent runs.

Method	Valid (%) ↑	EE pos. (cm) ↓	EE ori. (°) ↓	Collision ↓
Diffusion prior	64.3 ± 1.5	5.3 ± 0.1	7.5 ± 0.5	0.0402 ± 0.0013
Fixed-source FM	61.1 ± 3.8	3.8 ± 0.2	5.6 ± 0.5	0.0445 ± 0.0034
CNP-Flow	64.3 ± 4.6	$\mathbf{2.1 \pm 0.2}$	$\mathbf{2.9 \pm 0.3}$	$\mathbf{0.0366 \pm 0.0045}$

We use *Valid fraction* as the primary metric, as *Success* saturates with $N = 50$ candidates per task. Following MPD, *collision intensity* is computed as the fraction of evaluated trajectory waypoints that are in collision.

Performance without post-hoc refinement. To verify that the gains of CNP-Flow are not limited to settings with post-hoc refinement, we additionally evaluate trajectories without the refinement stage. As shown in Tab. 12, CNP-Flow reduces the end-effector position and orientation errors from 3.8 cm and $5.6°$ with fixed-source FM to 2.1 cm and $2.9°$, respectively. This indicates that the improvement is already present before the post-hoc optimization stage.

D Large-Scale Physical World-Model Benchmark

To evaluate whether the benefit of CNP-Flow extends beyond the low-resolution video benchmarks and RIVER backbone used in the main experiments, we further evaluate on ACWM-Phys (Xue et al., 2026), a recent action-conditioned physical world-model benchmark. We use ACWM-DiT-S, a 205M-parameter flow-matching DiT with a frozen Wan2.1 causal video VAE, operating on 240×240 videos.

ACWM-Phys covers four distinct physical regimes: rigid-body dynamics, deformable objects, particle-based dynamics, and kinematic systems. We select one task from each regime and evaluate both the in-distribution (IND) and out-of-distribution (OOD) splits. Specifically, we consider *push_cube*, *push_rope*, *pour_water*, and *robot_arm*. We compare the standard Gaussian source against CNP-Flow under the same flow-matching backbone and evaluation protocol. Results are averaged over three independent seeds and reported using MSE, masked MSE (M-MSE), PSNR, and SSIM.

As shown in Tab. 13, CNP-Flow improves all four reported metrics across every physical regime on both IND and OOD splits. The improvement varies with the underlying dynamics: the gains are modest on *push_rope*, while substantially larger improvements are observed on *pour_water* and *robot_arm*. In particular, on the OOD *robot_arm* split, CNP-Flow improves PSNR from 22.12 to 24.39 and SSIM from 0.9021 to 0.9511. These results show that the benefit of condition-dependent source modeling is not specific to the RIVER backbone or 64×64 video generation, and transfers to a substantially larger flow-matching DiT under distribution shift.

E Additional Analysis

Source-learning controls. To verify that the gains of CNP-Flow arise from meaningful source learning rather than additional model parameters, we freeze the Stage-1 FM and compare the same CNP architecture with and without flow reversal supervision. As shown in Tab. 14, without this supervision, CNP performs similarly to the standard Gaussian source, whereas flow reversal supervision substantially reduces FVD. This isolates the contribution of the supervision signal from the additional CNP parameters.

Prediction diversity. To verify that conditioning the source distribution does not reduce intra-condition diversity, we measure the average pairwise LPIPS among predictions generated from the same condition on BAIR64. As shown in Tab. 15, CNP-Flow achieves comparable diversity to the standard Gaussian source, indicating that stochastic prediction diversity is preserved.

Iterative source and flow refinement. To investigate whether further alternating source learning and FM refinement continues to improve performance, we repeat Stages 2–3 for multiple rounds. As shown in Tab. 16, a second round improves FVD on both BAIR64 and KTH, whereas an additional

Table 13: **Results on the ACWM-Phys physical world-model benchmark at 240×240 resolution.** We evaluate one task from each physical regime on both in-distribution (IND) and out-of-distribution (OOD) splits. All results are averaged over three independent seeds.

Regime	Task	Split	Method	MSE ↓	M-MSE ↓	PSNR ↑	SSIM ↑
Rigid	push_cube	IND	Vanilla FM	0.002886	0.017354	26.29	0.9546
			CNP-Flow (Ours)	**0.002192**	**0.013373**	**27.80**	**0.9647**
		OOD	Vanilla FM	0.002977	0.016039	26.78	0.9539
			CNP-Flow (Ours)	**0.002338**	**0.012326**	**28.13**	**0.9629**
Deformable	push_rope	IND	Vanilla FM	0.000213	0.002708	36.75	0.9878
			CNP-Flow (Ours)	**0.000207**	**0.002608**	**36.88**	**0.9889**
		OOD	Vanilla FM	0.000331	0.004163	35.34	0.9848
			CNP-Flow (Ours)	**0.000326**	**0.004054**	**35.46**	**0.9859**
Particle	pour_water	IND	Vanilla FM	0.002630	0.012214	26.03	0.9107
			CNP-Flow (Ours)	**0.001742**	**0.007419**	**27.90**	**0.9292**
		OOD	Vanilla FM	0.003425	0.013788	25.02	0.8748
			CNP-Flow (Ours)	**0.002514**	**0.009194**	**26.47**	**0.8930**
Kinematics	robot_arm	IND	Vanilla FM	0.001441	0.015606	28.70	0.9689
			CNP-Flow (Ours)	**0.000286**	**0.003000**	**35.70**	**0.9892**
		OOD	Vanilla FM	0.006552	0.050357	22.12	0.9021
			CNP-Flow (Ours)	**0.003854**	**0.036716**	**24.39**	**0.9511**

Table 14: **Effect of flow reversal supervision on source learning.** The Stage-1 FM is frozen for all variants.

Source, frozen FM	**PSNR ↑**	**SSIM ↑**	**FVD ↓**
Standard Gaussian	30.29	0.8591	179.97
CNP w/o flow reversal supervision	30.46	0.8612	179.54
CNP w/ flow reversal supervision	**30.85**	**0.9008**	**100.91**

round provides no consistent further gain. We therefore use the single-round pipeline in the main experiments for simplicity and efficiency.

Robustness of source retrieval via flow reversal. To verify that flow reversal supervision is insensitive to the numerical solver used for retrieval, we construct CNP supervision using Dopri, Heun-100, and Euler-100 on BAIR64 $1 \to 15$. As shown in Tab. 17, all three solvers yield similar downstream FVD, indicating that the observed improvement is robust to the choice of numerical solver for flow reversal.

Transport geometry. To directly examine whether the learned source provides better source–target alignment, we evaluate 256 BAIR64 condition–target pairs using Euler-100. As shown in Tab. 18, CNP-Flow reduces source–target distance, path length, cumulative turning, and endpoint deviation by 9.19%, 8.94%, 8.51%, and 9.69%, respectively. These results indicate that the learned source is better aligned with the conditional target region, leading to shorter and less curved transport trajectories.

Alternative Stage-3 couplings. We further compare alternative couplings for Stage-3 fine-tuning. As shown in Tab. 19, direct pairing via flow reversal achieves the shortest transport geometry, but requires the target sample and therefore has no inference-time counterpart. Conditional OT also shortens the path relative to CNP-Flow, yet increases turning and endpoint deviation and substantially degrades generation quality. These results indicate that minimizing per-sample transport distance alone is insufficient; the training coupling should remain compatible with the condition-dependent source available at inference.

Table 15: **Prediction diversity on BAIR64.** We report the average pairwise LPIPS among predictions generated from the same condition.

Source	Pairwise LPIPS ↑
Standard Gaussian	0.0666
CNP-Flow	0.0686

Table 16: **Iterative Stage-2/Stage-3 refinement.** FVD is reported after each round of source learning and FM refinement.

Dataset	Round 1	Round 2	Round 3
BAIR64	61.17	**60.48**	67.07
KTH	44.76	**35.04**	36.37

Table 17: **Robustness to the numerical solver used for flow reversal on BAIR64 prediction.** Values are mean ± standard deviation over three seeds.

Flow reversal solver	FVD ↓
Dopri	61.73 ± 1.47
Heun-100	$\mathbf{61.16 \pm 0.99}$
Euler-100	61.17 ± 1.56

Table 18: **Transport geometry on BAIR64.** Both methods are evaluated on the same 256 condition–target pairs using Euler-100.

Method	$\|\boldsymbol{x}_0 - \boldsymbol{x}_1\|_2$ ↓	Path length ↓	Turning (rad) ↓	Endpoint dev. ↓
Standard Gaussian	19.1715	18.6780	0.8292	2.7362
CNP-Flow (Ours)	**17.4091**	**17.0088**	**0.7586**	**2.4710**

Table 19: **Alternative Stage-3 couplings on BAIR64 prediction.** *Left*: transport geometry on 256 condition–target pairs using Euler-100. *Right*: downstream generation quality. Direct pairing via flow reversal requires the target and is therefore reported only as a training-time geometry reference.

Coupling	Distance ↓	Path ↓	Turning ↓	Endpoint dev. ↓
Flow reversal pair	**15.7342**	**15.7185**	**0.2261**	**0.4913**
Conditional OT	17.3715	16.2187	1.0610	3.9173
CNP-Flow (Ours)	17.4091	17.0088	0.7586	2.4710

Method	PSNR ↑	SSIM ↑	FVD ↓
Conditional OT	18.68	0.7828	101.43
Standard Gaussian	20.01	0.8121	70.16
CNP-Flow (Ours)	**20.15**	**0.8390**	**61.17**

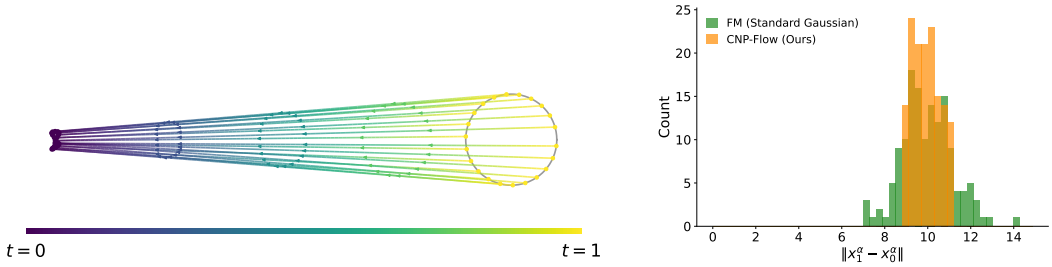
F Temporal Phase 2D Toy Example

Dataset. Our 2D toy experiment is designed to faithfully mirror the full three-stage pipeline of conditional generative modeling for temporal sequential data rather than serving as a generic trajectory forecasting benchmark. Each 2D trajectory point is treated as a temporally ordered observation, while temporal conditioning is constructed from the two preceding observations to capture both short-term and longer-range sequential dependencies. The dataset consists of parameterized circular trajectories generated as

$$\boldsymbol{x}_1^\alpha = [10 + \cos\alpha,\ 10 + \sin\alpha]^\top, \tag{11}$$

where α ranges from $\pi/180$ to 2π with $\Delta\alpha = \pi/360$. For each target point $\boldsymbol{x}_1^\alpha$, the condition is defined as $\mathcal{C}_\mathcal{T}^\alpha = [\boldsymbol{x}_1^{\alpha-2\Delta\alpha}, \boldsymbol{x}_1^{\alpha-\Delta\alpha}]$.

Implementation Details. All networks are implemented as multi-layer perceptrons (MLPs) with SiLU activation functions. FM model parameterizes the velocity field $\boldsymbol{v}_\theta$, taking as input $(\boldsymbol{x}_t^\alpha, t, \mathcal{C}_\mathcal{T}^\alpha)$ and outputting $\boldsymbol{v}_\theta$, using a 4-layer architecture with hidden dimension 128. The CNP model takes the condition $\mathcal{C}_\mathcal{T}^\alpha$ as input and predicts the parameters of a conditional isotropic Gaussian distribution



(a) Flow reversal trajectories.

(b) Transport distance.

Figure 6: Toy example visualization of the flow reversal mapping and transport distance analysis.

$(\boldsymbol{\mu}_\phi(\mathcal{C}_\mathcal{T}^\alpha), \sigma_\phi(\mathcal{C}_\mathcal{T}^\alpha))$, defining $p_\phi(\boldsymbol{x}_0^\alpha \mid \mathcal{C}_\mathcal{T}^\alpha) = \mathcal{N}(\boldsymbol{x}_0^\alpha; \boldsymbol{\mu}_\phi(\mathcal{C}_\mathcal{T}^\alpha), \sigma_\phi^2(\mathcal{C}_\mathcal{T}^\alpha)\mathbf{I})$, where σ_ϕ is constrained via a sigmoid activation to lie in $(0, 1)$. The CNP consists of 10 layers with a hidden dimension of 128. Our FM shares the same architecture as the baseline but replaces the standard Gaussian source with samples drawn from the learned conditional prior and is initialized from the pretrained baseline FM.

Analysis. This behavior can be understood from three observations: (1) As illustrated in Fig. 2, vanilla FM transports samples from a standard Gaussian prior to the target distribution. Under neighboring conditions, the corresponding source samples can nevertheless be widely separated, which may lead to intersecting ODE trajectories across timesteps and accumulated deviation from the target distribution in long-horizon generation. (2) Fig. 6(a) shows that flow reversal maps nearby target samples back to nearby source samples, indicating that the pretrained flow preserves local temporal correspondence in the source space. This motivates learning a conditional source prior that preserves proximity among source samples under similar conditions, resulting in more coherent ODE trajectories that better capture the temporal structure of the data. (3) As shown in Fig. 6(b), CNP-Flow produces a more concentrated source distribution with reduced variance compared to FM, which contributes to improved stability and accuracy in temporal generation.

G Additional Generated Results

We provide additional qualitative examples for video prediction and interpolation.



Figure 7: **Additional qualitative results on BAIR video prediction.**

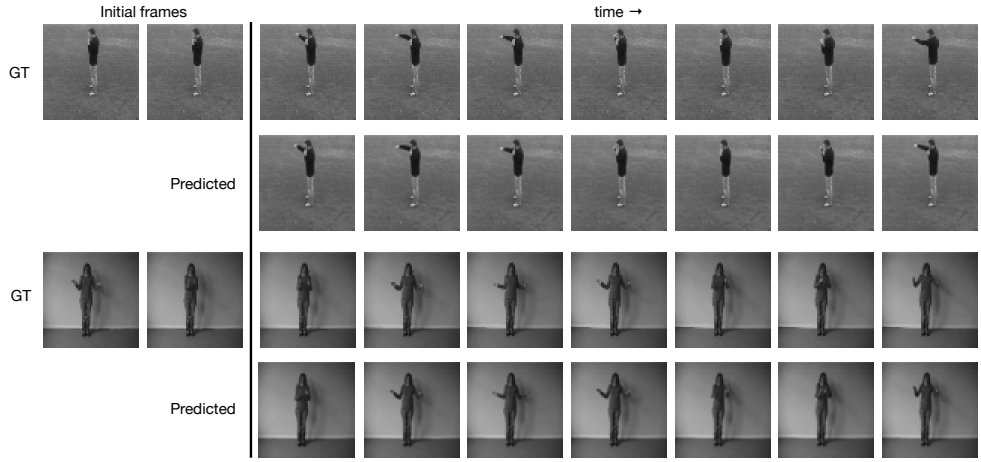


Figure 8: **Additional qualitative results on KTH video prediction.**

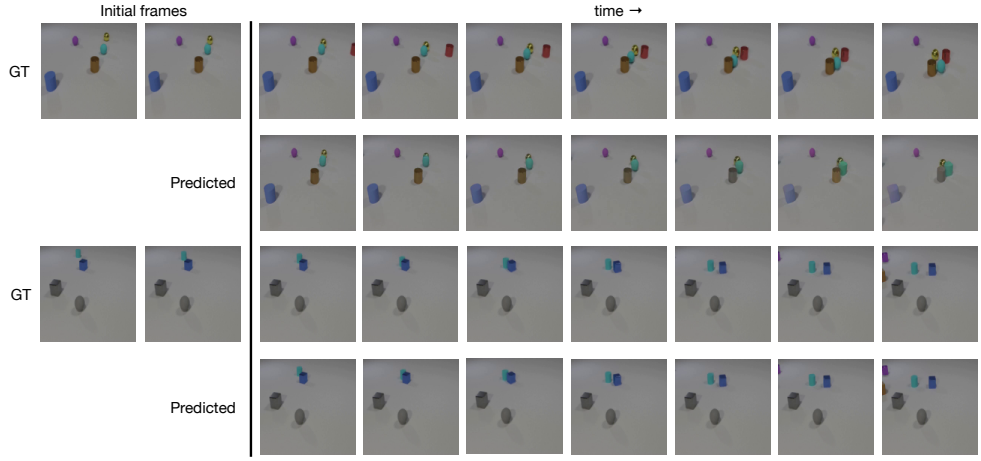


Figure 9: **Additional qualitative results on CLEVRER video prediction.**

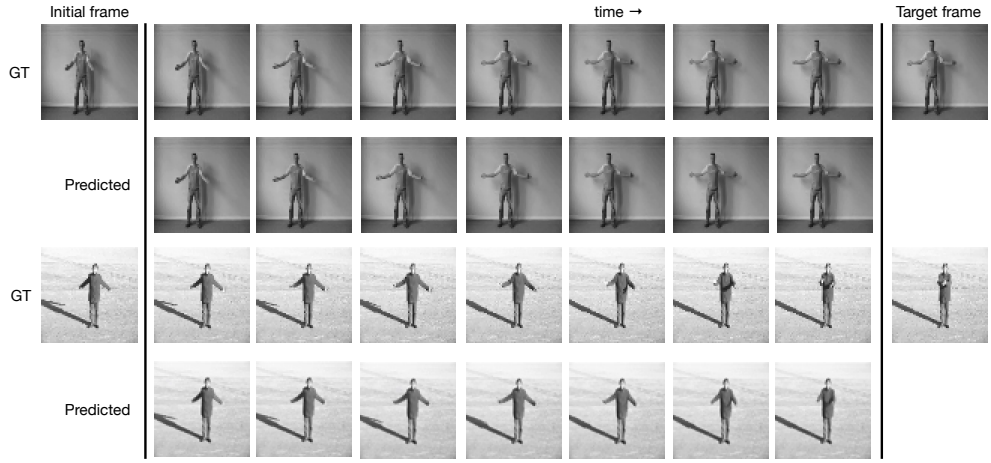


Figure 10: **Additional qualitative results on KTH video interpolation.**

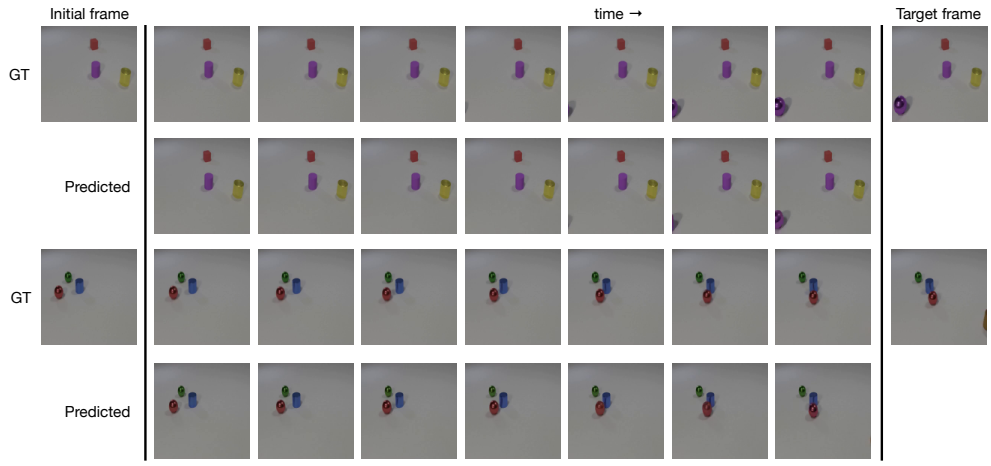


Figure 11: **Additional qualitative results on CLEVRER video interpolation.**