CONDITIONAL PARTICLE MATCHING: STATE-CONDITIONED PARTICLE FIELDS FOR FEW-STEP GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Drifting models train a generator by moving its endpoint pushforward toward the data distribution, while recent Drift Flow Matching extends this principle to time-indexed marginal transports. A marginal transport objective, however, does not uniquely identify how a generated clean output should depend on a particular noisy state: different reverse kernels can induce the same clean marginal. We introduce **Conditional Particle Matching** (CPM), a population conditional reverse-KL formulation over a stochastic reverse kernel $\rho_\theta(x_0 \mid x_t, t, c)$ that targets the corruption posterior $p(x_0 \mid x_t, t, c)$. This stochastic family can in principle represent multiple clean explanations of the same fixed corrupted state, and its intended state dependence can be tested by fixed-state sampling and observation-shuffle controls that pooled FID cannot see. At the population level, its conditional reverse-KL field decomposes into a data-prior term, an analytic likelihood term from the known corruption process, and a state-conditioned model self-field. We realize this structure as a practical feature-space particle semi-gradient built on the released Drifting system. CPM predicts in clean-endpoint coordinates: frozen perceptual features act only on clean-coordinate predictions and references, while the practical update introduces explicit noisy-state dependence through the analytic corruption likelihood and an SNR-dependent normalized distance in native latent space. In a controlled three-seed ImageNet-256 objective isolation at 30K updates, independent 50K one-NFE evaluation at CFG 2.0 gives FID 8.866 ± 0.128 for CPM-full, 8.960 ± 0.272 after shuffling the state affinity, and 9.332 ± 0.254 for pooled marginal particle matching. The small CPM–shuffled separation relative to seed variation, together with a failed trained A/B affinity control, prevents us from attributing the endpoint gain to recovery of the conditional posterior. The current evidence instead shows a consistent endpoint gap, across these three seeds, between the composite CPM recipe and its broader pooled-marginal control.

1 INTRODUCTION

Drifting and Drift Flow Matching (DFM) train generators through particle-based distribution matching (Deng et al., 2026; Ma et al., 2026b). Matching the pooled output distribution, however, does not determine which clean outputs are appropriate for a particular noisy input. Even a model that ignores the observation can reproduce the clean distribution. This motivates a conditional target that constrains the relationship between each noisy input and its possible clean explanations.

We introduce **Conditional Particle Matching** (CPM), built on the released Drifting system. Its stochastic generator predicts a clean sample from a noisy state x_t, noise level t, auxiliary randomness z, and optional class c. A conditional reverse-KL objective targets the corruption posterior $p(x_0 \mid x_t, t, c)$; varying z permits multiple explanations when the input is ambiguous. Each call predicts in clean coordinates, with optional re-noising between calls for multistep sampling (Figure 1). Learned perceptual comparisons therefore remain in clean coordinates. Our practical update combines feature-space particle interactions with an analytic corruption likelihood and native-state affinities. This surrogate does not guarantee posterior recovery.

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

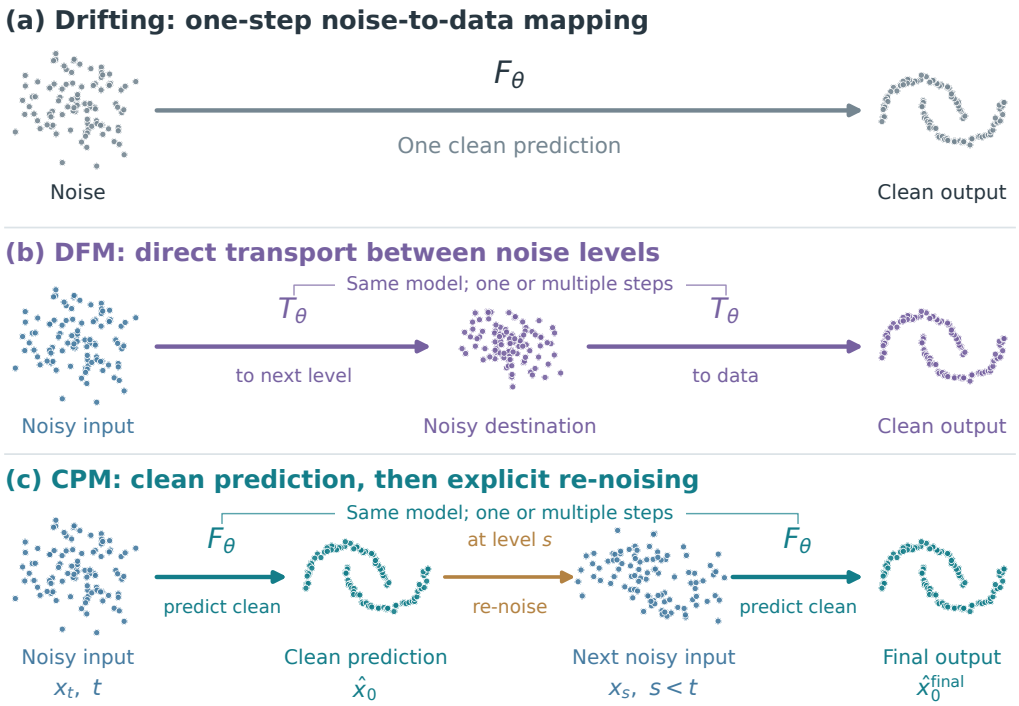

Figure 1: **Sampling interfaces of Drifting, DFM, and CPM.** (a) Drifting maps noise to a clean output in one call. (b) Drift Flow Matching (DFM) composes learned two-time transports T_θ to requested destination states, including intermediate noisy states (Ma et al., 2026b). (c) CPM predicts clean coordinates at every call. To continue, it explicitly re-noises that prediction as $x_s = \alpha_s \hat{x}_0 + \sigma_s \epsilon$, $s < t$, and feeds the result back as the next input. The last clean prediction is returned. Both DFM and CPM support one or multiple steps with the same model; two calls are shown. DFM's time arguments, class conditioning, and CPM's auxiliary z are omitted for clarity. Point clouds are schematic, not trained outputs or evidence of quality gains.

In a three-seed ImageNet-256 comparison at 30K updates and common CFG 2.0, CPM-full improves one-step FID over pooled marginal matching on every paired seed. Shuffling state affinities retains most of this gain, however, and the trained A/B affinity control fails to establish the conditional mechanism. These results support an endpoint improvement for the composite recipe; conditional recovery remains unconfirmed.

Our contributions are:

- **Conditional formulation.** We formulate a conditional reverse-KL objective and show why matching pooled marginals alone does not identify the desired reverse kernel.
- **Practical algorithm.** We derive the population particle field and build a clean-coordinate surrogate on Drifting, combining analytic likelihood and native-state affinity terms with one- or multistep sampling.
- **Controlled evaluation.** We provide conditional diagnostics and a three-seed ImageNet study that distinguishes endpoint quality from evidence of conditional recovery.

2 Related Work and Preliminaries

Drifting. Drifting learns a one-step noise-to-data map by regressing toward stop-gradient targets defined by an attraction–repulsion field (Deng et al., 2026). CPM inherits its feature-space interactions, normalization, data banks, and guidance. The released Drift–Laplace field is mean-shift-like rather

than an exact KDE score (Cao et al., 2026; Lai et al., 2026; Esteban-Casadevall et al., 2026); Appendix F derives the stopped-target regression gradient.

Drift Flow Matching. DFM trains two-time maps to match destination marginals within each time pair (Ma et al., 2026b):

$$T^{\mathrm{DFM}}_{s,r}(x_s) = x_s + (r-s)u_\theta(x_s, s, r), \qquad q^\theta_{s,r} = (T^{\mathrm{DFM}}_{s,r})_{\#}p_s. \tag{1}$$

Its time runs from noise at 0 to data at 1, opposite to ours, with $s < r$. DFM directly conditions its network on x_s and supports multistep sampling; CPM instead targets the clean law conditional on each noisy state. DFM's ImageNet field uses latent-MAE features at sampled destination levels, whereas CPM evaluates learned features on clean-coordinate predictions and references. This difference in feature domains does not establish MAE failure on noisy states. Table 1 reports DFM's published MNIST results and our local FFHQ Algorithm 2 reconstruction. The latter shares our FFHQ representation and evaluation protocol, but exact parity with the authors' implementation remains unverified.

Scores, distribution matching, and few-step generation. Score matching and its denoising formulation relate density gradients to Gaussian corruption (Hyvärinen, 2005; Vincent, 2011); diffusion and flow-based generators learn score or velocity fields (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Liu et al., 2023). Teacher-based distillation (Luo et al., 2023; Yin et al., 2024b;a; Salimans & Ho, 2022) and consistency models (Song et al., 2023) enable few-step sampling. CPM combines clean-data particle fields with an analytic likelihood, while retaining pretrained representations. Kernel moment matching and SVGD provide particle-based precedents (Li et al., 2015; Liu & Wang, 2016).

Corruption posteriors and bridges. CPM's posterior is induced by artificial corruption; inverse-problem posterior sampling instead conditions on external measurements (Chung et al., 2023). Bridge methods learn dynamics under endpoint constraints (Shi et al., 2023; De Bortoli et al., 2021). CPM uses clean prediction followed by re-noising (Section 3.2); its population construction uses conditional-density and Gaussian mean-shift identities (Hyndman et al., 1996; Cheng, 1995).

2.1 Forward corruption and notation

For an optional class c, let $x_0 \sim p_{\mathrm{data}}(\cdot \mid c)$ denote clean data and $\hat{x}_0$ a generated clean prediction. Our ImageNet setup uses variance-preserving Gaussian corruption

$$x_t = \alpha_t x_0 + \sigma_t \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, I), \qquad \alpha_t^2 + \sigma_t^2 = 1, \tag{2}$$

with kernel $q_t(x_t \mid x_0)$ and noisy marginal

$$q_t(x_t \mid c) = \int q_t(x_t \mid x_0) p_{\mathrm{data}}(x_0 \mid c)\, dx_0. \tag{3}$$

The ImageNet implementation uses $\sigma_t = t$ and $\alpha_t = \sqrt{1-t^2}$: $t = 0$ is the noiseless limit and $t = 1$ is pure noise. The clean posterior is

$$\pi_{t,c}(x_0 \mid x_t) := p_{\mathrm{data}}(x_0 \mid x_t, t, c) = \frac{p_{\mathrm{data}}(x_0 \mid c) q_t(x_t \mid x_0)}{q_t(x_t \mid c)}. \tag{4}$$

The generator

$$\hat{x}_0 = F_\theta(x_t, t, z, c), \qquad z \sim p_Z, \tag{5}$$

induces $\rho_\theta(\cdot \mid x_t, t, c)$. We use u as a dummy clean-space coordinate. The base law p_Z is parameter-free; the image implementation combines Gaussian channels and categorical noise embeddings. Varying z permits multiple clean explanations at fixed x_t. Actual latent use is evaluated through fixed-state spread and a $z = 0$ control, rather than assumed from this representational capability.

3 Conditional Particle Matching

3.1 From Drifting to a conditional target

Drifting trains a noise-to-data generator through particle interactions (Deng et al., 2026). CPM extends this rule to clean explanations of each noisy input: a realistic dog output, for example, may still contradict the fur color visible in the observation.

Varying z defines a distribution of clean predictions, the *reverse kernel* ρ_θ. CPM matches it to the corruption posterior $\pi_{t,c}$, with both laws conditioned on the same (x_t, t, c):

$$\mathcal{L}_{\mathrm{CPM}}(\theta) = \mathbb{E}_{x_t,t,c}[D_{\mathrm{KL}}(\rho_\theta \| \pi_{t,c})]\,. \tag{6}$$

This target constrains which clean outputs belong to each input, beyond matching their overall distribution. The formal comparison and expanded objective are in Appendices A and B.1. Training across noise levels supports one-step and multistep generation with the same model.

3.2 Conditional particle training

The conditional-KL objective induces the following population descent direction for a generated clean sample (see Appendix B.2 for regularity conditions and the derivation):

$$f_{\mathrm{CPM}} = \underbrace{\boldsymbol{s}_{\mathrm{data}}(\hat{x}_0 \mid c)}_{\text{(A) data attraction}} + \underbrace{\frac{\alpha_t}{\sigma_t^2}(x_t - \alpha_t \hat{x}_0)}_{\text{(B) analytic anchor}} - \underbrace{\boldsymbol{s}_\rho(\hat{x}_0 \mid x_t, t, c)}_{\text{(C) conditional model self-field}}\,. \tag{7}$$

Here $\boldsymbol{s}_{\mathrm{data}} = \nabla \log p_{\mathrm{data}}$ and $\boldsymbol{s}_\rho = \nabla \log \rho_\theta$. The terms favor clean-data realism, observation compatibility, and conditional diversity. The likelihood term corresponds to $\mathcal{L}_{\mathrm{like}} = \|x_t - \alpha_t \hat{x}_0\|^2/(2\sigma_t^2)$. The likelihood gradient vanishes at pure noise. For $\alpha_t > 0$, this term alone favors $\hat{x}_0 = x_t/\alpha_t$, so the particle field remains essential.

We retain Drifting's feature engineering for KDE-based particle matching. Features compare clean-coordinate predictions and references. CPM adds the analytic likelihood and weights generated neighbours by their noisy-input similarity within common class/time groups: $A^x_{t,ij} = \exp[-g(t)d^x_{ij}/h_x]$, where d^x_{ij} is a normalized native-latent distance, $h_x > 0$ is its bandwidth, and $g(t) = \alpha_t^2$ weakens selectivity near pure noise. Appendix B.4 gives the conditional-density interpretation, and Appendix B.5 derives the native-state geometry.

The same generator supports one-step generation from pure noise at $t = 1$ or multistep sampling: re-noise each clean prediction as $x_s = \alpha_s \hat{x}_0 + \sigma_s \epsilon$, $s < t$, and predict again. Use fresh, independent z and ϵ at each step and return the final clean prediction. With exact posterior draws and $X_t \sim q_t(\cdot \mid c)$, this preserves the desired level marginals (Proposition 3; Appendices B.6 and D). More steps need not improve an approximate model.

4 Experiments

We evaluate generation on two-dimensional moons and checkerboard, MNIST, FFHQ, and ImageNet-256. Two further ImageNet analyses examine input–output compatibility beyond marginal quality and the effect of conditional selectivity on generation quality.

4.1 Clean-Space Particle Matching Beyond One-Step Generation

Compared with one-step Drifting, CPM supports multistep generation, providing further opportunities to correct generated samples. Compared with DFM, CPM computes its output-space particle interactions in clean coordinates, preserving the target geometry instead of also matching noisy intermediate destinations.

Figure 2 illustrates these benefits on moons and checkerboard, comparing one-step Drifting with DFM and CPM at 1, 2, 4, and 8 network evaluations. The moon arcs and checkerboard cells therefore remain explicit matching targets at every step. Two-step CPM improves on its one-step outputs and reduces sliced Wasserstein-2 distance relative to one-step Drifting by 21% on moons and 15% on checkerboard. Compared with our local DFM baseline, CPM visibly produces clearer checkerboard cells and gaps, and better preserves the moon arcs at four and eight calls. Drifting retains lower boundary leakage, but its one-step generator is not trained to refine an existing sample.

4.2 MNIST and FFHQ few-step latent generation

Table 1 presents our **CPM** results on MNIST and FFHQ. Baseline MNIST cells quote Ma et al. (2026b, Table 1) only where method and NFE match; every populated FFHQ cell is measured locally.

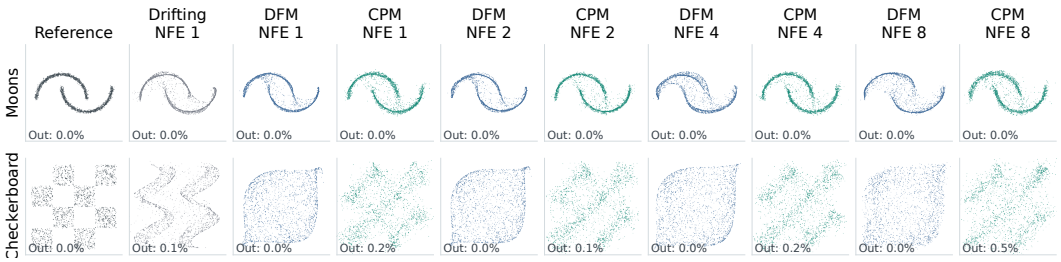

Figure 2: **Toy generation on moons (top) and checkerboard (bottom).** Reference data, one-step Drifting, and DFM/CPM at 1, 2, 4, and 8 network evaluations (NFE). Each panel shows 2,000 points, with common axes within each dataset; "Out" denotes samples outside the axes.

We use a frozen 16-dimensional MNIST autoencoder and a public 512-dimensional FFHQ ALAE, with a conditional MLP for each dataset. Each dataset's one-, two-, and four-call results reuse the same final EMA checkpoint.

Validation-selected nonuniform schedules reduce MNIST EMD from 36.434 at one call to 35.939 at two and 36.020 at four. FFHQ EMD similarly decreases from 171.198 to 150.879 and 151.050; within the CPM rows, two calls give the lowest EMD on both datasets. FFHQ pooled FID is 14.178, 14.953, and 14.571, respectively, so the multistep EMD gain does not also improve FID over one call.

Figure 3 visualizes the MNIST comparison. The top row shows decoded samples at fixed sample indices; the bottom row projects real and generated latents through a single UMAP fitted on training data. Reusing the projection and axis limits makes the NFE panels directly comparable. CPM already produces recognizable digits and distinct class clusters at one call. Additional calls preserve this organization while modestly improving latent distribution matching, as measured by EMD. The original-space silhouette scores are 0.215, 0.211, and 0.207 at 1, 2, and 4 NFE, respectively: class separation does not increase with the inference budget. Thus the figure illustrates the small distributional changes associated with the measured EMD gains, rather than progressive cluster separation or uniformly sharper individual samples.

The table also includes native local FFHQ reconstructions of DFM, MeanFlow, Flow Matching, and an endpoint-only Drift Model under the same representation and metric protocol. The table shows 50-call Flow Matching, 1-, 2-, and 5-call MeanFlow and DFM, and the one-call Drift Model, using final EMA checkpoints. The baseline protocol contains 14 evaluations, including the endpoint-only run; their full records are retained in the companion results. Among the rows shown in Table 1, the lowest FFHQ EMD is 150.879, achieved by CPM at 2 calls; the lowest FFHQ pooled FID is 3.558, achieved by Mean Flow at 5 calls. Each baseline uses one fixed recipe; CPM had two training candidates and validation-selected schedules, so this is a comparison under bounded training and tuning budgets.

Flow Matching receives explicit velocity supervision from each sampled data–noise pair, while MeanFlow builds on this signal with a learned derivative correction (Lipman et al., 2023; Geng et al., 2025a). Drift Model, DFM, and CPM instead estimate distribution-matching directions through finite particle interactions (Deng et al., 2026; Ma et al., 2026b). Constructing these directions using kernel weights or transport couplings from small particle groups in a high-dimensional latent space introduces an additional estimation problem. Errors in these directions can weaken the correction of covariance and diversity, even when coarse distributional statistics improve. This provides a plausible explanation for the stronger FFHQ FID of velocity-supervised methods under our fixed update budget. We interpret this as a possible optimization effect: FID measures decoded image-feature statistics and does not intrinsically favor a training objective. The present comparison does not isolate this mechanism causally.

Within the evaluated particle-based methods—Drift Model, DFM, and CPM—CPM achieves both the lowest measured FFHQ EMD and the lowest measured FFHQ FID. Its best EMD is 150.879 at two calls, compared with 157.174 for two-call DFM and 156.615 for one-call Drift Model. Its best FID is 14.178 at one call, compared with 14.375 for five-call DFM and 50.206 for one-call Drift

Table 1: **Class-conditional generation on MNIST and FFHQ.** Baseline MNIST values are quoted from Table 1 of Ma et al. (2026b); all FFHQ values are measured using our local benchmark. The local Drift Model uses an endpoint-only variant. CPM is evaluated locally on both datasets. Local EMD is class-averaged latent squared W_2; local FFHQ FID pools 10,000 decoded real and 10,000 decoded generated images. MNIST representation assets differ; local FFHQ tuning budgets also differ. Bold marks the lowest displayed EMD and FID. CPM accuracies are rounded to whole percentages; — denotes an unavailable result.

		MNIST		FFHQ	
Method	NFE	EMD↓	Accuracy↑	EMD↓	FID↓
Flow Matching (Lipman et al., 2023)	50	37.2	100.0%	193.004	4.438
Mean Flow (Geng et al., 2025a)	1	68.6	96.7%	214.875	5.698
Mean Flow (Geng et al., 2025a)	2	48.1	99.1%	185.112	3.920
Mean Flow (Geng et al., 2025a)	5	37.5	100.0%	210.083	**3.558**
Drift Model (Deng et al., 2026)	1	37.3	100.0%	156.615	50.206
DFM (Ma et al., 2026b)	1	37.3	100.0%	163.838	50.665
DFM (Ma et al., 2026b)	2	37.2	100.0%	157.174	27.562
DFM (Ma et al., 2026b)	5	37.2	100.0%	173.929	14.375
CPM (ours)	1	36.434	100%	171.198	14.178
CPM (ours)	2	**35.939**	100%	**150.879**	14.953
CPM (ours)	4	36.020	100%	151.050	14.571

Model. These two CPM minima occur at different sampling budgets; the claim concerns the best measured value of each metric across the tested configurations.

For MNIST, our reimplementation agrees with the results reported by Ma et al. (2026b). We therefore retain the published MNIST baseline entries from their Table 1 alongside our local CPM measurements. These baseline entries remain numerical references, since proximity of the scores does not verify that the representation and evaluation assets match. For FFHQ, our local EMD and FID values differ substantially from the published results, so we report locally evaluated results for all methods under a common representation and metric protocol. The local FFHQ Drift Model is trained separately at the fixed noise-to-data endpoint with the same particle field as our DFM reconstruction; it is not a verified reproduction of the original published Drift Model variant. All reported FFHQ results share our local metric protocol: FID pools 10,000 decoded real and 10,000 decoded generated images.

4.3 IMAGENET GENERATION

We evaluate class-conditional generation on ImageNet-256 (Deng et al., 2009; Russakovsky et al., 2015) using our 463M-parameter DiT-L/2 CPM model. The separate controlled studies in Sections 4.4 and 4.5 use DiT-B/2 generators.

Generation quality. Table 2 reports FID 1.30 and IS 311.2 for the L/2 model with four network function evaluations, alongside the published ImageNet reference values.

4.4 MATCHING EACH OUTPUT TO ITS NOISY INPUT

This controlled study uses Drifting's DiT-B/2 generator, fixed VAE, and frozen MAE features (Deng et al., 2026). Training uses one eight-GPU node.

Contribution of the conditional objective. We train CPM-full and two controls for 30K updates under the same architecture and training budget, using paired seeds 42–44. SHUFFLED AFFINITY permutes state-to-neighbour associations while retaining CPM's losses. MARGINAL-PM replaces both the conditional particle objective and likelihood anchor with pooled clean-marginal matching. Both controls retain CPM's architecture and sampling interface. Evaluation uses one NFE and the same 50K-sample manifest. At common CFG 2.0, mean FID is 8.866 ± 0.128 for CPM-full, 8.960 ± 0.272 for shuffled affinity, and 9.332 ± 0.254 for Marginal-PM. Shuffling retains most of the endpoint gain, and its ordering relative to CPM reverses on one seed. This comparison therefore

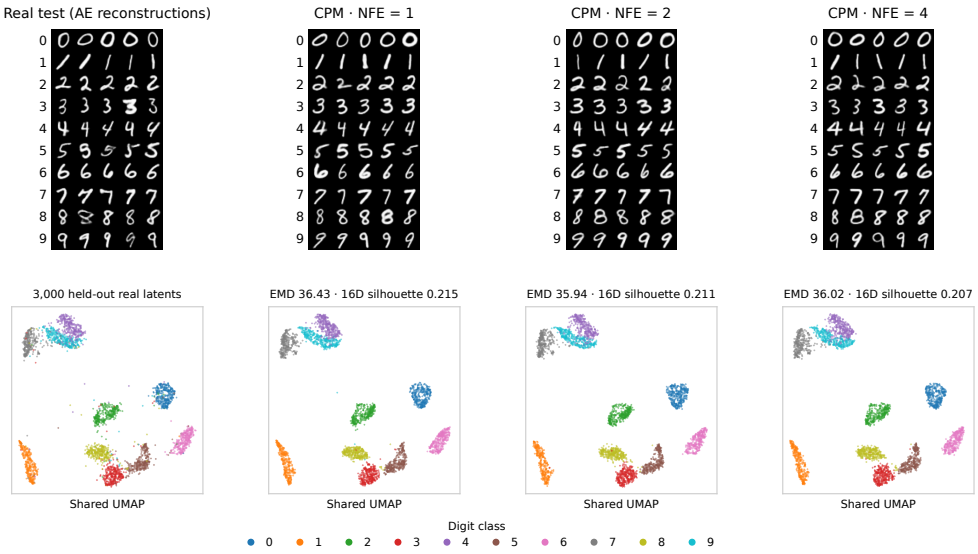

Figure 3: **MNIST generation with additional inference steps.** Real autoencoder reconstructions and CPM outputs at 1, 2, and 4 NFE, following the visualization format of Ma et al. (2026b, Figure 3). Top: five fixed, seeded examples per digit. Bottom: 300 latents per class under one shared UMAP (McInnes et al., 2018), fitted without labels on training data, with common axis limits. All CPM panels use the same checkpoint and evaluation runs as Table 1. EMD improves modestly over one call; class separation remains similar. Zooming in, the 2- and 4-NFE UMAP panels show fewer isolated points away from the main clusters than the 1-NFE panel. Silhouette scores are measured in the original 16D space.

supports the composite training recipe's endpoint improvement over Marginal-PM without isolating correct state affinities as its cause. Matched Drifting and multistep ImageNet comparisons remain planned.

Input–output pairing. Marginal quality can hide incorrect input–output associations. Using saved ImageNet outputs from three CPM models evaluated on the same 10,000 validation images, we reassign outputs among noisy inputs within each class. The output collection is unchanged, so FID is identical by construction.

Figure 4 measures squared distance to each input's clean source image in a fixed MAE feature projection. At log-SNR 2 ($\sigma = 0.345$), reassignment increases error from 13.445 to 13.701 (1.90%; 95% interval for the increase: $[0.128, 0.370]$). CPM's intervals at the other two noise levels include zero. Thus pairing reveals a compatibility difference invisible to marginal FID. This tests the whole generator, and the controls also show the effect; it does not establish a CPM-specific gain or posterior recovery.

4.5 Balancing Conditional Selectivity and Generation Quality

We use DiT-B/2 CPM models for this controlled bandwidth study. The conditioning bandwidth h_x controls how selectively particles interact across noisy inputs: smaller values restrict interactions to more similar states. In an ImageNet-256 sensitivity study at 45K updates and CFG 2.0, reducing h_x from 1 to 0.2, 0.1, and 0.05 increases logged FID from 9.14 to 13.69, 20.69, and 33.79, respectively. Figure 5 shows the common training checkpoints and all six saved guidance settings.

Stronger selectivity is therefore not automatically beneficial: a broader conditional neighbourhood better preserves generation quality in this regime. Narrow affinities downweight more reference particles, potentially making finite-particle estimates less reliable. This is a bandwidth sensitivity result; it does not establish an advantage over uniform or shuffled weights.

Table 2: **ImageNet-256 generation.** Reference rows reproduce Table 2 of Ma et al. (2026b). The final row reports our L/2 CPM model with four network function evaluations.

Method	Parameters	NFE	FID↓	IS↑
DiT-XL/2 (Peebles & Xie, 2023)	675M	250×2	2.27	278.2
SiT-XL/2 (Ma et al., 2024)	675M	250×2	2.06	270.3
SiT-XL/2+REPA (Yu et al., 2024)	675M	250×2	1.42	305.7
LightningDiT-XL/2 (Yao et al., 2025)	675M	250×2	1.35	295.3
RAE+DiT^{DH}-XL/2 (Zheng et al., 2025)	839M	50×2	1.13	262.6
iCT-XL/2 (Song & Dhariwal, 2024)	675M	1	34.24	–
Shortcut-XL/2 (Frans et al., 2025)	675M	1	10.60	–
MeanFlow-XL/2 (Geng et al., 2025a)	676M	1	3.43	–
AdvFlow-XL/2 (Lin et al., 2025)	673M	1	2.38	284.2
iMeanFlow-XL/2 (Geng et al., 2025b)	610M	1	1.72	282.0
S-VFM-XL/2 (Ma et al., 2026a)	677M	1	3.31	–
Drifting Model, L/2 (Deng et al., 2026)	463M	1	1.54	258.9
DFM, L/2 (Ma et al., 2026b)	463M	1	1.52	259.4
iCT-XL/2 (Song & Dhariwal, 2024)	675M	2	20.30	–
iMM-XL/2 (Zhou et al., 2025)	675M	1×2	7.77	–
MeanFlow-XL/2 (Geng et al., 2025a)	676M	2	2.93	–
AdvFlow-XL/2 (Lin et al., 2025)	673M	2	2.11	288.7
iMeanFlow-XL/2 (Geng et al., 2025b)	610M	2	1.54	285.1
S-VFM-XL/2 (Ma et al., 2026a)	677M	2	2.86	–
DFM, L/2 (Ma et al., 2026b)	463M	2	1.45	268.3
DFM, L/2 (Ma et al., 2026b)	463M	5	1.34	279.8
DFM, L/2 (Ma et al., 2026b)	463M	10	1.31	287.4
CPM, L/2 (ours)	463M	4	1.30	311.2

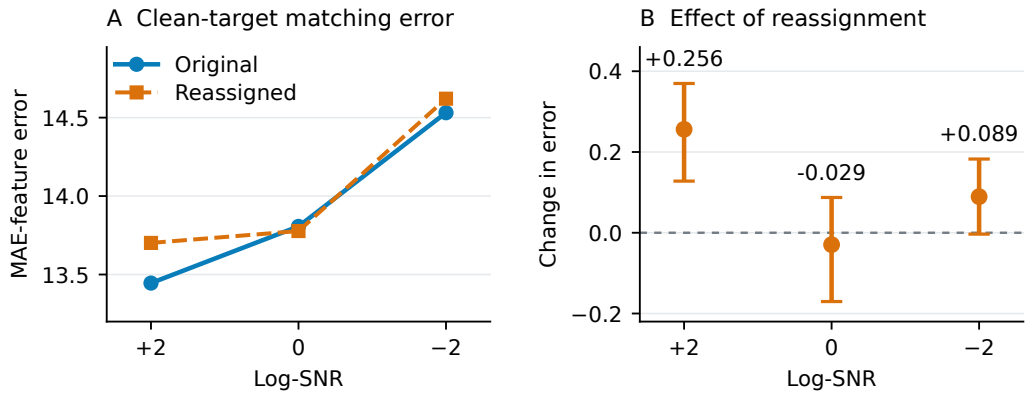

Figure 4: **ImageNet output quality does not determine pairing quality.** (a) Mean clean-feature error before and after reassigning the same outputs within class. (b) Paired error increase with pointwise 95% run/class bootstrap intervals; positive values favor the original pairing. All three evaluated noise levels are shown. FID is unchanged by construction.

5 Limitations

Our sampling guarantee assumes exact posterior sampling, while the accuracy of the posterior learned through finite-particle training remains an empirical question. Performance depends on frozen representations, particle count, and bandwidths, while replicas and repeated sampling add compute. Forward corruption readily supplies clean–noisy pairs, but constructing a representative set of clean targets for the same partially noisy input is difficult when many same-class explanations remain plausible. The current implementation learns this conditional diversity through particle interactions and the known corruption likelihood, without constructing explicit posterior target sets.

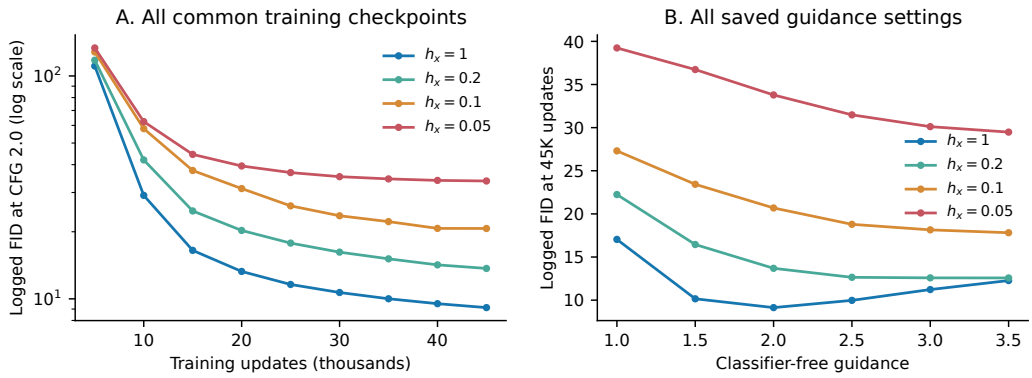

Figure 5: **Conditional selectivity affects ImageNet generation quality.** Left: logged FID at CFG 2.0 over all common checkpoints, with a logarithmic vertical axis. Right: all six saved CFG values at 45K updates. The broader $h_x = 1$ neighbourhood achieves lower FID throughout both comparisons. The training configurations agree except for bandwidth and planned stopping point. These are historical logged evaluations; the exact sample count and evaluator version are not bound by an immutable evaluation record.

These interactions introduce an estimation problem beyond Flow Matching's explicit per-pair targets: narrow conditional neighborhoods may contain too few effective particles, while broad ones may mix distinct conditional distributions. Reaching comparable FID may therefore require larger particle sets or more optimization, increasing the training budget. We hypothesize that efficiently constructing representative posterior target sets could provide more direct supervision for conditional diversity, improving sample quality and accelerating convergence.

More broadly, scaling particle-based training, including CPM, to text-to-image and video generation poses additional challenges. Diverse text conditions can make it difficult to assemble enough relevant reference particles, while larger video representations can increase the memory and computation required for particle interactions. We leave efficient particle sampling and neighbour selection in these settings to future work.

6 Conclusion

We introduced Conditional Particle Matching (CPM), which targets the distribution of clean explanations for each noisy input through particle interactions, an analytic likelihood, and native-state affinities. Its clean-output interface supports both one-step generation and iterative refinement through re-noising. Experiments on synthetic distributions, MNIST, FFHQ, and ImageNet demonstrate the effectiveness of this approach for few-step generation. Additional sampling steps improve sample geometry in the synthetic examples and reduce latent distributional error on MNIST and FFHQ, showing that the same trained generator can benefit from a larger inference budget. Future work will focus on more efficient particle estimation and explicit posterior target construction to improve training efficiency and scale to richer conditioning.

AI use statement

OpenAI Codex was used as a general-purpose assistant for prose editing, manuscript organization, LaTeX revision, consistency checks between the manuscript and the implementation, and experimental scripting and execution for the two-dimensional and MNIST studies. Their reported measurements were computed by the recorded training and evaluation code. The authors reviewed the mathematical claims, citations, implementation descriptions, and experimental results and retain responsibility for the final paper.

REFERENCES

Jiarui Cao, Zixuan Wei, and Yuxin Liu. Gradient flow drifting: Generative modeling via Wasserstein gradient flows of KDE-approximated divergences. *arXiv preprint arXiv:2603.10592*, 2026. URL https://arxiv.org/abs/2603.10592.

Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995. doi: 10.1109/34.400568. URL https://doi.org/10.1109/34.400568.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/940392f5f32a7ade1cc201767cf83e31-Abstract.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. URL https://www.image-net.org/static_files/papers/imagenet_cvpr09.pdf.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026. URL https://arxiv.org/abs/2602.04770.

Maria Esteban-Casadevall, Jorge Carrasco-Pollo, Max Welling, Jan-Willem van de Meent, Erik J. Bekkers, and Floor Eijkelboom. Kernel-gradient drifting models. *arXiv preprint arXiv:2605.10727*, 2026. URL https://arxiv.org/abs/2605.10727.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OlzB6LnXcS.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6d13e085b79d454da5910e4ca82a3d9d-Abstract-Conference.html.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b. URL https://arxiv.org/abs/2512.02012.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Rob J. Hyndman, David M. Bashtannyk, and Gary K. Grunwald. Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics*, 5(4):315–336, 1996. doi: 10.1080/10618600.1996.10474715. URL https://www.tandfonline.com/doi/abs/10.1080/10618600.1996.10474715.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL https://www.jmlr.org/papers/v6/hyvarinen05a.html.

Chieh-Hsin Lai, Bac Nguyen, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, Stefano Ermon, and Molei Tao. A unified view of drifting and score-based models. *arXiv preprint arXiv:2603.07514*, 2026. URL https://arxiv.org/abs/2603.07514.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1718–1727, 2015. URL https://proceedings.mlr.press/v37/li15.html.

Shanchuan Lin, Ceyuan Yang, Zhijie Lin, Hao Chen, and Haoqi Fan. Adversarial flow models. *arXiv preprint arXiv:2511.22475*, 2025. URL https://arxiv.org/abs/2511.22475.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/b3ba8f1bee1238a2f37603d90b58898d-Abstract.html.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f115f619b62833aadc5acb058975b0e6-Abstract-Conference.html.

Chenrui Ma, Xi Xiao, Tianyang Wang, Xiao Wang, and Yanning Shen. Learning straight flows: Variational flow matching for efficient generation. *arXiv preprint arXiv:2511.17583*, 2026a. URL https://arxiv.org/abs/2511.17583.

Chenrui Ma, Xi Xiao, Lin Zhao, Tianyang Wang, Ferdinando Fioretto, and Yanning Shen. Drift flow matching. *arXiv preprint arXiv:2605.17244*, 2026b. URL https://arxiv.org/abs/2605.17244.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40, 2024.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. URL https://arxiv.org/abs/1802.03426.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/spotlight/6538.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/c428adf74782c2092d254329b6b02482-Abstract-Conference.html.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

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*, 2021. URL https://iclr.cc/virtual/2021/poster/3177.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142. URL https://doi.org/10.1162/NECO_a_00142.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15703–15712, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.html.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. URL https://arxiv.org/abs/2410.06940.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. URL https://arxiv.org/abs/2510.11690.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78651–78686, 2025. URL https://proceedings.mlr.press/v267/zhou25c.html.

A Conditional matching and marginal ambiguity

A.1 Formal comparison

Marginal matching instead pools the outputs across observations. For fixed (t, c), this gives

$$\bar{\rho}_{\theta,t,c}(A) := \int \rho_\theta(A \mid x_t, t, c)\, q_t(dx_t \mid c) \tag{8}$$

for measurable A. A marginal particle objective compares this pooled law with $p_{\mathrm{data}}(\cdot \mid c)$, discarding which input produced each output. The following proposition formalizes why that comparison is weaker.

Proposition 1 (Conditional matching is stronger than marginal matching)**.** *Fix* (t, c) *and define*

$$R_{\theta,t,c}(x_t, x_0) = q_t(x_t \mid c)\rho_\theta(x_0 \mid x_t, t, c), \tag{9}$$

$$P_{t,c}(x_t, x_0) = q_t(x_t \mid c)p(x_0 \mid x_t, t, c). \tag{10}$$

Then

$$D_{\mathrm{KL}}(R_{\theta,t,c}\|P_{t,c}) = \mathbb{E}_{x_t\sim q_t(\cdot|c)} D_{\mathrm{KL}}(\rho_\theta(\cdot \mid x_t, t, c)\|p(\cdot \mid x_t, t, c))\,. \tag{11}$$

Also,

$$D_{\mathrm{KL}}(\bar{\rho}_{\theta,t,c}\|p_{\mathrm{data}}(\cdot \mid c)) \le D_{\mathrm{KL}}(R_{\theta,t,c}\|P_{t,c}). \tag{12}$$

The converse fails: $\rho_\theta(x_0 \mid x_t, t, c) = p_{\mathrm{data}}(x_0 \mid c)$ *matches the clean marginal but generally not the corruption posterior.*

Proof. Both $R_{\theta,t,c}$ and $P_{t,c}$ have marginal $q_t(x_t \mid c)$. The KL chain rule therefore gives

$$D_{\mathrm{KL}}(R_{\theta,t,c}\|P_{t,c}) = D_{\mathrm{KL}}(q_t(\cdot \mid c)\|q_t(\cdot \mid c)) \tag{13}$$

$$+ \mathbb{E}_{x_t\sim q_t(\cdot|c)} D_{\mathrm{KL}}(\rho_\theta(\cdot \mid x_t, t, c)\|p(\cdot \mid x_t, t, c))\,, \tag{14}$$

which proves Eq. 11. The map $(x_t, x_0) \mapsto x_0$ is measurable, so KL data processing proves Eq. 12. Finally, substituting the input-independent kernel $p_{\mathrm{data}}(x_0 \mid c)$ into Eq. 8 gives the correct clean marginal. Except at exact pure noise or another degenerate case, Bayes' rule makes $p(x_0 \mid x_t, t, c)$ depend on x_t, so the converse fails. □

Combining Eqs. 11 and 12 gives the direct conditional–marginal comparison:

$$D_{\mathrm{KL}}(\bar{\rho}_{\theta,t,c}\|p_{\mathrm{data}}(\cdot \mid c)) \le \mathbb{E}_{x_t\sim q_t(\cdot|c)} D_{\mathrm{KL}}(\rho_\theta(\cdot \mid x_t, t, c)\|\pi_{t,c}(\cdot \mid x_t))\,. \tag{15}$$

A.2 An input-dependent counterexample

Two input-dependent rules can also share the same marginals. Let A sample the true posterior and B flip the sign of an A sample. For the symmetric prior $p_{\mathrm{data}}(x_0) = \frac{1}{2}\mathcal{N}(-m, s^2) + \frac{1}{2}\mathcal{N}(m, s^2)$, with $m > 0$, these are the reverse conditionals of

$$\text{A: } x_t = \alpha_t x_0 + \sigma_t \boldsymbol{\varepsilon}, \qquad \text{B: } x_t = -\alpha_t x_0 + \sigma_t \boldsymbol{\varepsilon}. \tag{16}$$

With A as the known corruption and inputs drawn from their common q_t, both rules yield the correct pooled clean law: A does so by Bayes' rule, and B preserves that law because the prior is invariant under a sign flip. The same symmetry makes their noisy marginals identical. At informative noise levels, their posteriors differ, so population conditional KL distinguishes the rules. Appendix G gives the closed-form posteriors.

A.3 Pure-noise endpoint

These examples concern observations that retain information about the clean sample. At the pure-noise endpoint, that information is absent.

Corollary 1 (Pure-noise endpoint)**.** *If* $\alpha_T = 0$*, then* x_T *is independent of* x_0 *conditional on* c *and*

$$p(x_0 \mid x_T, T, c) = p_{\mathrm{data}}(x_0 \mid c). \tag{17}$$

Thus the posterior target contains no state-specific clean information at exact pure noise. Literal conditional KL still requires $\rho_\theta(\cdot \mid x_T, T, c) = p_{\mathrm{data}}(\cdot \mid c)$ *for* $q_T(\cdot \mid c)$*-almost every* x_T*, including independence from* x_T*; pooled endpoint matching constrains only the integrated clean law.*

Proof. When $\alpha_T = 0$, the likelihood $q_T(x_T \mid x_0)$ is independent of x_0. It cancels from the numerator and denominator in Bayes' rule, giving the stated posterior. Zero conditional KL requires equality of the conditional laws almost everywhere, whereas marginal matching requires only equality after integrating over x_T. □

B ADDITIONAL CPM THEORY

B.1 POPULATION OBJECTIVE

Equation 6 suppresses the shared conditioning of the two clean-output laws. Explicitly, sample $c \sim p(c)$, $t \sim \nu$ on $(0, 1]$, and $x_t \sim q_t(\cdot \mid c)$. The population objective is

$$\mathcal{L}_{\mathrm{CPM}}(\theta) = \mathbb{E}_c \mathbb{E}_{t\sim\nu} \mathbb{E}_{x_t\sim q_t(\cdot|c)} \left[D_{\mathrm{KL}}(\rho_\theta(\cdot \mid x_t, t, c) \| \pi_{t,c}(\cdot \mid x_t)) \right]. \tag{18}$$

If zero loss is globally attainable within the stochastic family and a parameter value attains it, nonnegativity of KL implies posterior correctness for $p(c)$-almost every c, ν-almost every t, and $q_t(\cdot \mid c)$-almost every x_t. This almost-everywhere, per-level property does not select a cross-time coupling or guarantee fixed-latent path consistency.

An idealized population hybrid restricts Eq. 6 to interior levels and mixes it with a pooled endpoint KL,

$$\mathcal{L}_{\mathrm{end}}(\theta) = \mathbb{E}_c D_{\mathrm{KL}}(\bar{\rho}_{\theta,T,c} \| p_{\mathrm{data}}(\cdot \mid c)), \qquad \mathcal{L}_{\mathrm{hybrid}} = (1 - \omega_{\mathrm{end}}) \mathcal{L}_{\mathrm{CPM,int}} + \omega_{\mathrm{end}} \mathcal{L}_{\mathrm{end}}. \tag{19}$$

For $0 < \omega_{\mathrm{end}} < 1$, a zero hybrid objective identifies the interior posteriors almost everywhere and the pooled clean endpoint law. Corollary 2 additionally requires exactness at each queried interior level; almost-everywhere equality under ν alone does not ensure this for an arbitrary fixed sampling grid.

B.2 EXACT POPULATION GRADIENT IN THE INTERIOR

For a common clean-space coordinate u, define

$$\boldsymbol{s}_\rho(u \mid x_t, t, c) := \nabla_u \log \rho_\theta(u \mid x_t, t, c), \tag{20}$$

$$\boldsymbol{s}_p(u \mid x_t, t, c) := \nabla_u \log p(u \mid x_t, t, c). \tag{21}$$

Proposition 2 (Conditional score-difference gradient)**.** *Assume that F_θ is differentiable in θ; the generated and target conditional laws have differentiable densities; $\rho_\theta \ll p(\cdot \mid x_t, t, c)$; and differentiation may be exchanged with integration under a common dominating function, with no parameter-dependent boundary contribution. Then the exact population pathwise gradient of Eq. 6 is*

$$\nabla_\theta \mathcal{L}_{\mathrm{CPM}} = \mathbb{E}_{c,t,x_t,z} \left[J_\theta F_\theta(x_t, t, z, c)^\top \left\{ \boldsymbol{s}_\rho(\hat{x}_0 \mid x_t, t, c) - \boldsymbol{s}_p(\hat{x}_0 \mid x_t, t, c) \right\} \right]. \tag{22}$$

The density-shape derivative of $\log \rho_\theta$ has zero expectation by the log-derivative identity; the remaining pathwise derivative is the generator Jacobian contracted with a score difference. The corresponding clean-space *descent velocity* has the opposite orientation, $v = \boldsymbol{s}_p - \boldsymbol{s}_\rho$. Appendix C gives the derivation. This identity assumes smooth densities and regular support. Empirical image laws and low-dimensional generators need not satisfy those assumptions. The stopped-regression gradient in Appendix F is distinct from this population identity.

B.3 TARGET POSTERIOR SCORE AND ANALYTIC ANCHOR

Bayes' rule gives

$$p(u \mid x_t, t, c) = \frac{q_t(x_t \mid u) p_{\mathrm{data}}(u \mid c)}{q_t(x_t \mid c)}. \tag{23}$$

Taking $\nabla_u \log$ and using the Gaussian forward kernel yields

$$\boldsymbol{s}_p(u \mid x_t, t, c) = \underbrace{\nabla_u \log p_{\mathrm{data}}(u \mid c)}_{\text{class-conditional data-prior score}} + \underbrace{\frac{\alpha_t}{\sigma_t^2}(x_t - \alpha_t u)}_{\text{forward-likelihood score}}. \tag{24}$$

Evaluated at the generated sample, the likelihood score is realized by

$$\mathcal{L}_{\mathrm{like}} = \frac{1}{2\sigma_t^2} \|x_t - \alpha_t \hat{x}_0\|_2^2, \qquad \nabla_\theta \mathcal{L}_{\mathrm{like}} = -J_\theta F_\theta^\top \frac{\alpha_t}{\sigma_t^2}(x_t - \alpha_t \hat{x}_0). \tag{25}$$

We call this term the *analytic anchor*. It is the exact likelihood component of the target score, not the complete posterior objective. Its likelihood-only optimum is $\hat{x}_0 = x_t/\alpha_t$, which is generally not the posterior mean and amplifies corruption noise by σ_t/α_t when α_t is small. It is therefore

useful only in balance with the data-prior and model-entropy fields. At $\alpha_t = 0$, its gradient vanishes. This decomposition supplies input compatibility directly from the known corruption model rather than requiring it to emerge solely from learned particle geometry. Its exactness applies only to the likelihood summand: the clean-data prior field and conditional model self-field remain finite-particle approximations, and the complete practical update is not an exact posterior-score estimator.

B.4 Generator score from conditional particles

The implicit generator has no tractable conditional density. At a common (t, c), consider generated pairs $\{(\hat{x}_0^{(j)}, x_t^{(j)})\}_{j=1}^M$. Let d_t^x be a normalized Euclidean distance between native corruption latents at the common level, and let $g(t)$ be a declared schedule-dependent selectivity factor. A kernel conditional-density estimator is

$$\widehat{\rho}_\theta(u \mid x_t, t, c) = \frac{\sum_j K_h(u - \hat{x}_0^{(j)}) A_t^x(x_t, x_t^{(j)})}{\sum_j A_t^x(x_t, x_t^{(j)})}, \tag{26}$$

where K_h is a normalized output-density kernel and

$$A_t^x(x_t, x_t^{(j)}) = \exp\left(-\frac{g(t)}{h_x} d_t^x(x_t, x_t^{(j)})\right). \tag{27}$$

All pairs in this derivation share (t, c). The affinity depends on their native corruption coordinates and is constant with respect to u. Because the denominator is independent of u, the conditional score is

$$\nabla_u \log \widehat{\rho}_\theta(u \mid x_t, t, c) = \frac{\sum_j A_t^x(x_t, x_t^{(j)}) \nabla_u K_h(u - \hat{x}_0^{(j)})}{\sum_j A_t^x(x_t, x_t^{(j)}) K_h(u - \hat{x}_0^{(j)})}. \tag{28}$$

Evaluating at particle i and excluding $j = i$ gives a leave-one-out estimate. For d-dimensional clean coordinates and $K_h(v) = (2\pi h^2)^{-d/2} \exp(-\|v\|^2/(2h^2))$,

$$\widehat{\boldsymbol{s}}_\rho^{(i)} = -\frac{1}{h^2} \frac{\sum_{j \neq i} A_{t,ij}^x K_{ij} (\hat{x}_0^{(i)} - \hat{x}_0^{(j)})}{\sum_{j \neq i} A_{t,ij}^x K_{ij}}. \tag{29}$$

The input affinity changes the relative contribution of generated neighbours according to their noisy origins. In finite batches, the denominator can have negligible effective mass; adding a numerical floor stabilizes the estimate but introduces bias. Classical KCDE consistency also requires sample-size and bandwidth conditions that are unrealistic in raw high-dimensional image space.

Combining the exact score identities gives the population clean-space descent field in Eq. 7, where $\boldsymbol{s}_{\text{data}}(u \mid c) := \nabla_u \log p_{\text{data}}(u \mid c)$.

B.5 Geometry of Gaussian conditioning

At fixed (t, c), expanding the Gaussian likelihood gives

$$p(x_0 \mid x_t, t, c) \propto p_{\text{data}}(x_0 \mid c) \exp\left[\frac{\alpha_t}{\sigma_t^2} x_t^\top x_0 - \frac{\alpha_t^2}{2\sigma_t^2} \|x_0\|_2^2\right]. \tag{30}$$

Thus $\eta_t(x_t) = \alpha_t x_t / \sigma_t^2$ is the likelihood's natural conditioning parameter. At a common t, Euclidean proximity in x_t is equivalent up to the schedule scalar α_t/σ_t^2 to proximity in $\eta_t(x_t)$ when $\alpha_t > 0$. This does not make Euclidean distance an exact posterior metric: the posterior also depends on the data prior and generally has anisotropic local geometry.

For two independent corruptions at a common level in D_x latent dimensions,

$$\mathbb{E}\left[\|x_t^{(i)} - x_t^{(j)}\|_2^2 \mid x_0^{(i)}, x_0^{(j)}\right] = \alpha_t^2 \|x_0^{(i)} - x_0^{(j)}\|_2^2 + 2D_x \sigma_t^2. \tag{31}$$

To obtain Eq. 31, expand $x_t^{(i)} - x_t^{(j)} = \alpha_t(x_0^{(i)} - x_0^{(j)}) + \sigma_t(\epsilon_i - \epsilon_j)$. The cross term has zero expectation, and $\epsilon_i - \epsilon_j \sim \mathcal{N}(0, 2I)$. Under the variance-preserving schedule, the selectivity factor in the main text obeys $g(t) = \alpha_t^2 = \text{SNR}(t)/(1 + \text{SNR}(t))$, which tends to zero as the signal vanishes.

B.6 MARGINAL PRESERVATION UNDER EXACT CONDITIONAL SAMPLING

For $s < t$, define a *posterior-resample/re-corrupt* transition: sample $\widehat{X}_0$ from the clean posterior associated with X_t, then draw a new state from $q_s(\cdot \mid \widehat{X}_0)$ using fresh independent forward noise.

Proposition 3 (Interior marginal preservation)**.** *Fix $s, t \in (0,1)$ and c. If $X_t \sim q_t(\cdot \mid c)$, $\widehat{X}_0 \sim p(\cdot \mid X_t, t, c)$, and $X_s \sim q_s(\cdot \mid \widehat{X}_0)$ conditionally independently of X_t, then $X_s \sim q_s(\cdot \mid c)$.*

The population sampler uses this transition between queried levels and fresh, mutually independent z_k and re-corruption noises. It is not the usual diffusion bridge $q(x_s \mid x_t, x_0)$ and should not be called a standard ancestral reverse transition.

Corollary 2 (Hybrid sampler)**.** *Initialize $X_T \sim q_T = \mathcal{N}(0, I)$. Suppose the endpoint generator produces $\widehat{X}_0 \sim p_{\mathrm{data}}(\cdot \mid c)$ after marginalizing over (X_T, z), and each subsequent interior generator is exact. Re-corrupting the endpoint output to the first interior level and repeatedly applying Proposition 3 preserves the correct marginal at every queried level. The final clean prediction has law $p_{\mathrm{data}}(\cdot \mid c)$.*

This result is deliberately limited. It does not recover the joint path law, does not guarantee visual identity along a realized path, and does not imply that increasing the number of steps improves an approximate model. In the exact limit every supported step count is already marginally correct. Whether extra steps correct or compound approximation error is therefore a central empirical question.

C DERIVATION OF THE CONDITIONAL SCORE-DIFFERENCE GRADIENT

Fix (x_t, t, c) and abbreviate the generated and target conditional densities by $\rho_\theta(u)$ and $p(u)$. Reparameterization gives

$$\mathcal{K}(\theta) = \int \rho_\theta(u)\big(\log \rho_\theta(u) - \log p(u)\big)\, du = \mathbb{E}_z\big[\log \rho_\theta(\hat{x}_0) - \log p(\hat{x}_0)\big]. \tag{32}$$

For the target term,

$$\nabla_\theta \log p(\hat{x}_0) = J_\theta F_\theta^\top \nabla_u \log p(u)|_{u=\hat{x}_0}. \tag{33}$$

The total derivative of the generated log-density separates into

$$\nabla_\theta \log \rho_\theta(\hat{x}_0) = \partial_\theta \log \rho_\theta(u)|_{u=\hat{x}_0} + J_\theta F_\theta^\top \nabla_u \log \rho_\theta(u)|_{u=\hat{x}_0}. \tag{34}$$

Under the domination and support assumptions,

$$\mathbb{E}_{u\sim\rho_\theta}\left[\partial_\theta \log \rho_\theta(u)\right] = \int \partial_\theta \rho_\theta(u)\, du = \partial_\theta 1 = 0. \tag{35}$$

Combining the remaining terms and restoring the outer expectations proves Proposition 2.

D PROOFS FOR MARGINAL PRESERVATION

Proof of Proposition 3. Writing kernels and measures consistently,

$$\Pr(X_s \in dx_s \mid c) = \iint q_t(dx_t \mid c)\, p(dx_0 \mid x_t, t, c)\, q_s(dx_s \mid x_0) \tag{36}$$

$$= \int p_{\mathrm{data}}(dx_0 \mid c)\, q_s(dx_s \mid x_0) = q_s(dx_s \mid c). \tag{37}$$

The second equality uses Bayes' rule followed by marginalization over x_t.

Proof of Corollary 2. By assumption, marginalizing the endpoint output over (X_T, z) gives $p_{\mathrm{data}}(\cdot \mid c)$. Re-corruption to the first interior level s with q_s therefore gives $q_s(\cdot \mid c)$. Proposition 3 then applies inductively to every interior transition. At the final queried level, marginalizing an exact posterior draw again gives p_{data}.

E WHY LITERAL CONDITIONAL MATCHING NEEDS STOCHASTIC OUTPUT

Proposition 4 (Stochasticity for literal conditional reverse KL)**.** *Assume $p_{\mathrm{data}}(\cdot \mid c)$ is absolutely continuous and $\sigma_t > 0$. Then $p(\cdot \mid x_t, t, c)$ is atomless for $q_t(\cdot \mid c)$-almost every x_t. A deterministic generator induces a Dirac conditional law and*

$$D_{\mathrm{KL}}\big(\delta_{F_\theta(x_t,t,c)} \,\big\|\, p(\cdot \mid x_t, t, c)\big) = +\infty. \tag{38}$$

Proof. For $\sigma_t > 0$, the likelihood $q_t(x_t \mid x_0) = \mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 I)$ is strictly positive. Bayes' rule therefore expresses the posterior as an absolutely continuous density proportional to the product of the data density and a positive Gaussian factor. It assigns zero mass to any singleton. A deterministic generator assigns unit mass to one singleton, so its law is not absolutely continuous with respect to the posterior and the reverse KL is infinite. □

The proposition establishes stochasticity only for the literal smooth-density conditional KL. It does not establish that an auxiliary latent is necessary for marginal generation: a deterministic generator may use the random x_1 as its marginal noise source. It also does not guarantee attainability. If $\dim z$ is too small or the map is rank-deficient, the generated conditional may remain singular.

F GRADIENT OF STOPPED-TARGET PARTICLE REGRESSION

Let $h_i(\theta) = \phi(F_\theta(x_t^{(i)}, t, z_i, c))/\kappa$ be a generated feature, where ϕ is frozen and the positive normalization κ is held constant during differentiation. For a particle displacement V_i, define

$$g_i = \mathrm{sg}[h_i(\theta) + V_i], \qquad \mathcal{L}_{\mathrm{reg}}(\theta) = \frac{1}{B}\sum_{i=1}^{B} \|h_i(\theta) - g_i\|_2^2. \tag{39}$$

Here sg holds its argument fixed in the backward pass. Applying the chain rule at the parameter value used to construct the targets gives

$$\nabla_\theta \mathcal{L}_{\mathrm{reg}} = \frac{2}{B}\sum_i J_\theta h_i^\top (h_i - g_i) \tag{40}$$

$$= -\frac{2}{B}\sum_i J_\theta h_i^\top V_i = -\frac{2}{B\kappa}\sum_i J_\theta F_\theta^\top J_\phi^\top V_i. \tag{41}$$

The Jacobians in the final sum are evaluated at the corresponding generated sample. Thus stopped-target regression maps the prescribed feature-space displacement through the feature and generator Jacobians. It does not differentiate through the rule that constructs V_i. Equality with the population conditional-KL gradient requires an appropriate native-space score difference; an arbitrary feature displacement or noninvertible feature map does not establish that equality.

G CLOSED-FORM GAUSSIAN-MIXTURE POSTERIOR

Consider a balanced J-component isotropic Gaussian mixture in d dimensions,

$$p_{\mathrm{data}}(x_0) = \frac{1}{J}\sum_{j=1}^{J} \mathcal{N}(x_0; \mu_j, \tau^2 I), \qquad \tau > 0, \tag{42}$$

and corruption $x_t = a\alpha_t x_0 + \sigma_t \epsilon$, where $a \in \{+1, -1\}$, $\sigma_t > 0$, and $\epsilon \sim \mathcal{N}(0, I)$. Conditioned on component j, the noisy state is Gaussian with mean $a\alpha_t \mu_j$ and covariance $(\alpha_t^2\tau^2 + \sigma_t^2)I$. Bayes' rule therefore gives the component weights below. Within each component, completing the square in

$$-\frac{\|x_0 - \mu_j\|^2}{2\tau^2} - \frac{\|x_t - a\alpha_t x_0\|^2}{2\sigma_t^2} \tag{43}$$

gives the posterior covariance and mean:

$$p^{(a)}(x_0 \mid x_t, t) = \sum_{j=1}^{J} \omega_j^{(a)}(x_t, t)\mathcal{N}(x_0; m_j^{(a)}(x_t, t), \Sigma_t), \tag{44}$$

$$\omega_j^{(a)}(x_t, t) = \frac{\mathcal{N}\big(x_t; a\alpha_t\mu_j, (\alpha_t^2\tau^2 + \sigma_t^2)I\big)}{\sum_{r=1}^{J}\mathcal{N}(x_t; a\alpha_t\mu_r, (\alpha_t^2\tau^2 + \sigma_t^2)I)}, \tag{45}$$

$$\Sigma_t = \left(\tau^{-2}I + \alpha_t^2\sigma_t^{-2}I\right)^{-1}, \tag{46}$$

$$m_j^{(a)}(x_t, t) = \Sigma_t\left(\tau^{-2}\mu_j + a\alpha_t\sigma_t^{-2}x_t\right). \tag{47}$$

If the component means occur in symmetric pairs, changing a permutes the components of the noisy marginal and leaves that marginal unchanged. For the posterior, the same change reverses the association between the observed state and its possible clean explanations, yielding the counterexample in Eq. 16.