Continuous Personalized Diffusion Model by Spinor-Component Forward Geometry

GeonWoo Jeong Sang Beom Chu Wonjun Kim
Seung Hun Oh Bokyoon Na
Tech University of Korea
{sgtj78, 2024152036, kdi1124, sho, bkna}@tukorea.ac.kr

Abstract

Existing conditional diffusion models typically inject condition information into the reverse-time denoising process, either as model inputs or through guidance. Although effective for conditional fidelity, this design may burden the denoiser when compact continuous signals, such as user preference ratios or mixture degrees, must support both image reconstruction and condition-wise separation, often degrading sample quality. In this work, we reinterpret this limitation not only as a problem of condition *representation*, but also as a problem of condition *placement*. We propose the **Continuous Personalized Diffusion Model (CPDM)**, which shifts the locus of conditioning from reverse-time signal injection to forward geometry formation by coupling bounded spinor-component coordinates with normalized image-space drift directions. The resulting condition-dependent forward drift forms condition-specific terminal geometry, from which continuous changes along the spinor-component coordinate are reflected as continuous visual transitions rather than endpoint interpolation. Experiments show that, even with a single scalar condition, CPDM achieves competitive performance compared with standard and higher-capacity reverse-conditioning baselines and supports continuous generation under compact scalar control. These results suggest forward geometry design as a practical alternative to conventional reverse-time conditioning for compact continuous personalization. Code is available at `https://github.com/geonwooj/Continuous-Personalized-Diffusion-Model-by-Spinor-Component-Forward-Geometry`.

1 Introduction

Most existing conditional diffusion models treat condition information as a signal injected into the reverse-time denoising process, either as model inputs or through guidance [Dhariwal & Nichol, 2021, Ho & Salimans, 2022, Rombach et al., 2022]. This paradigm is effective for high-quality condition-aligned generation, but continuous personalization requires a different form of control. Signals such as user preference ratios or mixture degrees are intended to define a controllable path between conditions, rather than simply select one endpoint condition.

Interpolating between endpoint condition signals is a natural way to obtain such a path, but it does not necessarily produce meaningful intermediate samples: the reverse process still starts from a shared terminal space and must resolve both image reconstruction and condition-wise separation during denoising. Compact or low-dimensional signals provide a simpler continuous coordinate, but often degrade generation quality because the denoiser receives only a limited conditioning signal while being asked to separate different visual domains—an effect that worsens in inter-domain settings where endpoint distributions require different generative manifolds or geometries. These observations suggest that the limitation lies not only in condition representation, but also in condition placement.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

From this perspective, we propose the **Continuous Personalized Diffusion Model (CPDM)**, a forward-geometry-based conditional diffusion framework for compact continuous personalization. Instead of using the condition only as a reverse-time steering signal, CPDM introduces a deterministic condition-dependent drift into the DDPM forward mean while preserving the standard variance schedule. Here, personalization refers to user-selectable continuous control over the modeled domains or attributes. Our experiments evaluate this form of control, without learning user preferences from feedback or performing subject-identity adaptation. Guided by the scale–direction template of the spinor formulation, this drift combines an explicit drift strength, bounded spin-component coordinates, and normalized image-space drift bases. Forward geometry organization thus becomes an explicit design problem—how far terminal states should be separated and which image-space direction should define that separation—while remaining fully compatible with the standard DDPM reverse machinery.

Experiments show that this minimal scalar design outperforms several standard and higher-capacity reverse-conditioning baselines and supports ordered scalar traversal. Reverse-sampling diagnostics further show greater condition-wise separation after deterministic drift correction and earlier persistent intended-domain classifier evidence (Appendix G). Together, these results demonstrate that forward geometry shapes both generation performance and condition-specific trajectory evolution under compact scalar control.

2 Related Work

2.1 Diffusion / Score-Based Generative Models

Diffusion models were originally introduced as generative processes based on nonequilibrium thermodynamics [Sohl-Dickstein et al., 2015], and DDPM later established a practical framework for sample generation through fixed forward noising and learned reverse Markov transitions [Ho et al., 2020]. Score-based SDE formulations further generalized this framework to continuous-time dynamics [Song et al., 2021b]. This shared Gaussian-prior formulation has remained a standard backbone not only for unconditional diffusion models, but also for conditional generation.

2.2 Reverse-Time Conditioning and Guidance

Conditional diffusion models have primarily evolved by injecting condition information into the denoiser or by modulating the reverse process through guidance [Dhariwal & Nichol, 2021, Ho & Salimans, 2022, Rombach et al., 2022]. These approaches provide strong conditional fidelity, but they primarily treat the condition as a steering signal for the reverse generation trajectory. In contrast, they do not explicitly make condition-specific geometry formation a central design component of the generative process.

2.3 Conditional Forward and Trajectory Shifting

Several recent studies have explored incorporating condition information into the forward process or trajectory-level dynamics. Shift-DDPM introduces condition-dependent trajectory shifts into the diffusion process [Zhang et al., 2023]. Instead of relying only on reverse-time denoising with a condition input, it assigns condition-specific diffusion trajectories by learning a shift predictor that determines the direction and magnitude of the trajectory shift. In this way, condition information affects the diffusion trajectory itself, and the quality of conditional generation depends on whether the learned shift direction provides a reliable separation between conditions.

CCDM addresses continuous conditional generation from a different angle [Ding et al., 2026]. For a continuous label y, CCDM constructs condition-dependent embeddings, including a short embedding used for U-Net conditioning and a long embedding used to define a label-dependent covariance matrix. This covariance structure is incorporated into the forward transition, reverse process, training objective, and sampling procedure. The denoising network predicts x_0 rather than the noise, and the training loss is weighted by the inverse of the condition-dependent covariance, so that image denoising is adapted to the local neighborhood of the continuous condition.

These approaches indicate that continuous or condition-specific generation can benefit from modifying the diffusion trajectory or forward statistics, rather than treating the condition only as an additional reverse-time input.

2.4 Our Positioning

We position CPDM as a minimal forward-organization approach for compact continuous personalization. Rather than increasing the capacity of reverse-time conditioning or introducing an auxiliary learned shift predictor, CPDM couples the condition coordinate directly to the DDPM forward mean. Compact spinor-component values act as bounded coefficients for normalized image-space drift bases, while the standard DDPM variance schedule is left unchanged. Consequently, the condition reorganizes the mean geometry of the forward distribution, rather than changing its covariance structure or the denoiser architecture. This restriction makes the forward design space explicit: the key variables are the strength, temporal schedule, and spatial direction of a deterministic drift.

3 Method

3.1 Standard DDPM Formulation: Forward and Reverse Processes

We consider data $x_0 \sim p_{\mathrm{data}}(x)$, a timestep $t \in \{1, \dots, T\}$, and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$. Let $\{\beta_t\}_{t=1}^{T}$ denote the variance schedule of DDPM, and define

$$\alpha_t = 1 - \beta_t, \qquad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i. \tag{1}$$

The forward process of DDPM [Ho et al., 2020] is given by

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

so the conditional distribution of x_t takes the closed-form Gaussian

$$q(x_t \mid x_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, x_0,\ (1 - \bar{\alpha}_t) I\big)\,. \tag{3}$$

In the reverse process, a learned denoiser $\epsilon_\theta(x_t, t)$—or equivalently an estimator $x_{0,\theta}(x_t, t)$—approximates $p_\theta(x_{t-1} \mid x_t)$ as a Gaussian transition, and samples are generated sequentially from $t = T$ to $t = 1$. Standard DDPM can thus be understood as a generative framework that learns the reverse mean μ_θ on top of a fixed Gaussian forward process.

The DDPM formulation above provides a general and tractable Gaussian forward process. However, in conditional generation, this shared forward process leaves two roles to the reverse denoiser: separating condition-dependent distributions and reconstructing fine image details. To reduce this burden, we seek a way to organize condition-dependent displacement within the forward process itself. This motivates a spinor-component construction, which provides bounded continuous coordinates together with a scale–direction template for forward geometry design.

3.2 Forward Geometry from Spinor Components

CPDM implements this idea by mapping the condition c to spin-component coordinates and coupling them to a deterministic condition-dependent forward drift. Rather than numerically integrating an additional latent-time spinor trajectory during diffusion, CPDM uses the spinor-induced observable coordinates in closed form. This formulation preserves the Gaussian tractability of $q(x_t \mid x_0, c)$ and keeps the model compatible with the standard DDPM reverse machinery, while allowing the condition to shape the forward terminal geometry before reverse denoising begins.

3.2.1 Spinor-Induced Condition Coordinates

We define a reduced complex two-component spinor as

$$\tilde{\psi}(\tau, c) = \begin{bmatrix} \alpha(\tau, c) \\ \beta(\tau, c) \end{bmatrix} \in \mathbb{C}^2. \tag{4}$$

Formally, we consider the following Schrödinger equation:

$$i\, \frac{d}{d\tau} \tilde{\psi}(\tau, c) = \hat{H}\, \tilde{\psi}(\tau, c), \qquad \hat{H} = B\, \hat{U}, \tag{5}$$

where B denotes the strength of the interaction and $\hat{U}$ denotes an operator that specifies a preferred direction in the spinor space. This relation provides the scale–direction template used to construct bounded spin-component values.

In the practical construction, condition dependence is represented by the normalized spinor components $\alpha(c)$ and $\beta(c)$. Let

$$\alpha(c) = f_1(c) + if_2(c), \qquad \beta(c) = g_1(c) + ig_2(c), \tag{6}$$

where f_i and g_i, for $i = 1, 2$, are real-valued functions. Then the normalization condition is given by

$$|\alpha(c)|^2 + |\beta(c)|^2 = f_1(c)^2 + f_2(c)^2 + g_1(c)^2 + g_2(c)^2 = 1. \tag{7}$$

From this normalized spinor representation, we compute the expectation values of the spin components.[1] Thus, we define

$$s_i(c) = \tilde{\psi}(c)^\dagger \sigma_i \tilde{\psi}(c), \qquad i \in \{x, y, z\}. \tag{8}$$

In real-component form,

$$s_x(c) = 2\big(f_1(c)g_1(c) + f_2(c)g_2(c)\big), \tag{9}$$

$$s_y(c) = 2\big(f_1(c)g_2(c) - f_2(c)g_1(c)\big), \tag{10}$$

$$s_z(c) = \big(f_1(c)^2 + f_2(c)^2\big) - \big(g_1(c)^2 + g_2(c)^2\big). \tag{11}$$

In this work, condition information is encoded by the spin-component expectation values $(s_x(c), s_y(c), s_z(c))$, which lie in $[-1, +1]$ and serve as bounded coefficients for the condition-dependent drift. The spinor-component formulation therefore provides a normalized multi-axis coordinate interface, rather than an unstructured scalar condition.

3.2.2 From Spin Components to Multi-Axis Forward Drift

We translate spin-component expectation values into a condition-dependent drift applied to the DDPM forward process. Guided by the scale–direction structure of the Schrödinger-inspired relation, CPDM couples each spin component $s_i(c)$ to a fixed image-space drift direction $\hat{u}_i$. The general multi-axis drift is defined as

$$r_t^{\text{full}}(c) = A\, C_t \sum_{i \in \{x,y,z\}} s_i(c)\, \hat{u}_i, \tag{12}$$

where A controls the drift strength, C_t is the diffusion-time schedule, and each $\hat{u}_i \in \mathbb{R}^{H \times W \times 3}$ is a normalized image-space direction basis associated with the spin-component axis i. Thus, $s_i(c)$ acts as a signed coordinate coefficient for the corresponding image-space basis $\hat{u}_i$. As a result, the spinor-component formulation provides a structured basis for multi-axis and multi-style continuous control, where different semantic or stylistic directions can be assigned to different drift bases within a common bounded coordinate system.

In this work, we focus on the single-axis s_z case to test whether a minimal scalar condition can improve generation performance through forward geometry formation and support continuous transitions between two endpoint domains. The spinor-component formulation provides normalized condition coordinates that are coupled to image-space drift bases to construct condition-dependent forward geometry. In this single-axis setting, however, $s_z(c) = 2|\alpha(c)|^2 - 1$ is mathematically equivalent to a generic bounded scalar in $[-1, 1]$. KL/ELBO compatibility follows from deterministic mean translation rather than spinor dynamics. The scalar drift design used in our experiments is described next.

[1] Standard physical literature includes a factor of $1/2$ in the spin expectation value to reflect the nature of half-integer spin fermions. In the present work, however, this factor is omitted to simplify the notation.

3.3 Forward Geometry Design by Scalar Drift

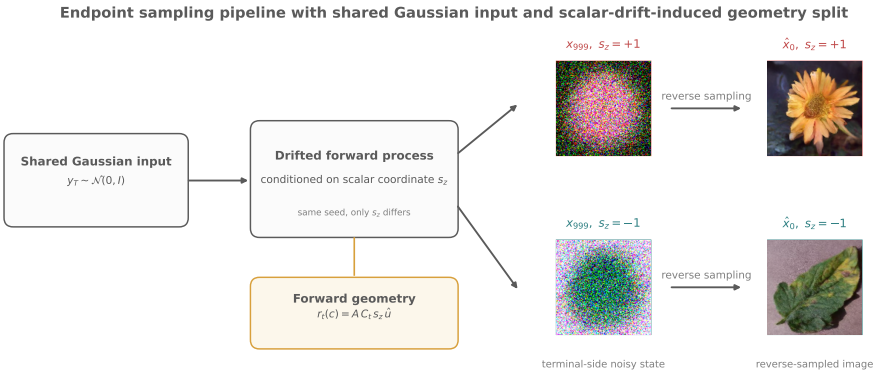


Figure 1: Practical scalar CPDM pipeline with shared Gaussian input and drift-induced forward geometry split. From a shared noise seed $y_T \sim \mathcal{N}(0, I)$, opposite scalar coordinates $s_z = +1$ and $s_z = -1$ induce separate terminal geometries, leading to visually distinct reverse-sampled outputs.

Our study focuses on the scalar case, where the condition-dependent drift is defined as

$$r_t(c) = A\,C_t\,s_z(c)\,\hat{u}_z, \tag{13}$$

where A controls the drift strength, C_t controls temporal accumulation, and $\hat{u}_z$ is the fixed image-space drift basis defined in Section 3.2.2. To avoid scale-driven differences across drift bases, each $\hat{u}$ basis is globally rescaled to have an average pixel-wise ℓ_2 norm of 0.6 over its channel dimension, preserving the spatial structure of the basis while controlling the overall drift energy scale. We do not learn $\hat{u}_z$, because making the drift basis trainable would turn the forward perturbation itself into a moving target and would no longer correspond to the fixed-forward noise-prediction setting used by DDPM. For continuous conditioning, we directly use the spinor-induced coordinate on the z-axis.

$$s_z(c) = |\alpha(c)|^2 - |\beta(c)|^2. \tag{14}$$

Because the spinor is normalized as $|\alpha(c)|^2 + |\beta(c)|^2 = 1$, we have $s_z(c) \in [-1, 1]$. The endpoints $s_z = +1$ and $s_z = -1$ correspond to the two endpoint domains, while $s_z = 0$ represents their balanced midpoint. Thus, $s_z(c)$ is not used as a binary label, but as a continuous signed coordinate that directly controls the condition-dependent forward geometry.

Adding this scalar drift to the standard DDPM forward process gives

$$x_t = \sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,\big(\epsilon + r_t(c)\big), \qquad \epsilon \sim \mathcal{N}(0, I). \tag{15}$$

Accordingly, the conditional forward distribution becomes

$$q(x_t \mid x_0, c) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,r_t(c),\ (1-\bar{\alpha}_t)I\big)\,. \tag{16}$$

Taking expectation over ϵ, the terminal mean is

$$\mathbb{E}_\epsilon[x_T \mid x_0, c] = \sqrt{\bar{\alpha}_T}\,x_0 + \sqrt{1-\bar{\alpha}_T}\,r_T(c), \tag{17}$$

so the condition-dependent terminal shift is $\sqrt{1-\bar{\alpha}_T}\,r_T(c)$. Thus, CPDM preserves the variance structure of the Gaussian forward process while reorganizing the terminal geometry by shifting only the condition-dependent mean coordinate.

The time schedule $C_t \in [0, 1]$ controls how the drift accumulates along the diffusion time axis, with $C_0 = 0$ and $C_T = 1$. Together with the drift strength A and the image-space basis $\hat{u}_z$, it determines how the scalar coordinate $s_z(c)$ is translated into controllable terminal geometry.

3.4 η-Target Learning Objective

CPDM is trained on endpoint boundary bands rather than interior mixture coordinates, so continuous responses at $s_z \in (-1, +1)$ are treated as generalization from real endpoint-domain data rather than as a consequence of explicit interpolation or mixture-target supervision.

3.4.1 Base CPDM

We define the drift-augmented target as $\eta_t(s_z) = \epsilon + r_t(s_z)$ and train the network to predict this combined noise-drift quantity. Base CPDM samples s_z uniformly from the boundary band $[-1.0, -0.95] \cup [0.95, 1.0]$, and minimizes the following η-target loss:

$$\mathcal{L}_{\text{main}} = \mathbb{E}_{x_0,\, \epsilon,\, t,\, s_z \sim q_{\text{bd}}} \Big[\|\hat{\eta}_\theta(x_t(s_z),\, t,\, s_z) - (\epsilon + r_t(s_z))\|_2^2 \Big]. \tag{18}$$

This objective exposes the model not only to the exact endpoints, but also to nearby continuous values of s_z, encouraging η-prediction to remain stable under small coordinate changes.

3.4.2 Continuous CPDM

To improve local smoothness along the scalar control axis, Continuous CPDM preserves the endpoint-band objective of Base CPDM and adds a pairwise consistency regularizer between nearby scalar coordinates. Specifically, we construct nearby condition pairs (s_1, s_2) that share the same x_0, timestep t, and noise ϵ, with both coordinates lying in the same boundary band defined in Section 3.4.1 and $|s_1 - s_2| \in [0.01, 0.05]$. We then introduce the pair consistency regularizer:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}\left[\frac{\Big\| \big(\hat{\eta}_\theta(x_t(s_2), t, s_2) - \hat{\eta}_\theta(x_t(s_1), t, s_1)\big) - \big(r_t(s_2) - r_t(s_1)\big) \Big\|_2^2}{\|r_t(s_2) - r_t(s_1)\|_2^2 + \varepsilon_{\text{pair}}} \right]. \tag{19}$$

and define the final objective as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \lambda\, \mathcal{L}_{\text{diff}}. \tag{20}$$

Here, $\mathcal{L}_{\text{diff}}$ enforces local consistency around the endpoint anchors and stabilizes the model response to small scalar-coordinate changes.

3.5 Reverse Sampling and KL Alignment in the DDPM-Equivalent y-World

The forward process of CPDM shifts only the conditional mean through a deterministic drift:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \big(\epsilon + r_t(c)\big). \tag{21}$$

Define $m_t(c) := \sqrt{1 - \bar{\alpha}_t}\, r_t(c)$, and the translated variable

$$y_t := x_t - m_t(c). \tag{22}$$

Substituting gives

$$y_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \tag{23}$$

which exactly recovers the standard DDPM forward form. Therefore, the drift-augmented forward process of CPDM is aligned with standard DDPM in the translated y-world. Since this translation has unit Jacobian for each fixed t and c, the corresponding KL terms can be analyzed in the standard DDPM y-world, and CPDM reuses the standard DDPM reverse machinery without introducing a new reverse-time Gaussian family. The full derivation and sampling algorithm are provided in Appendices A and B.

4 Experiments

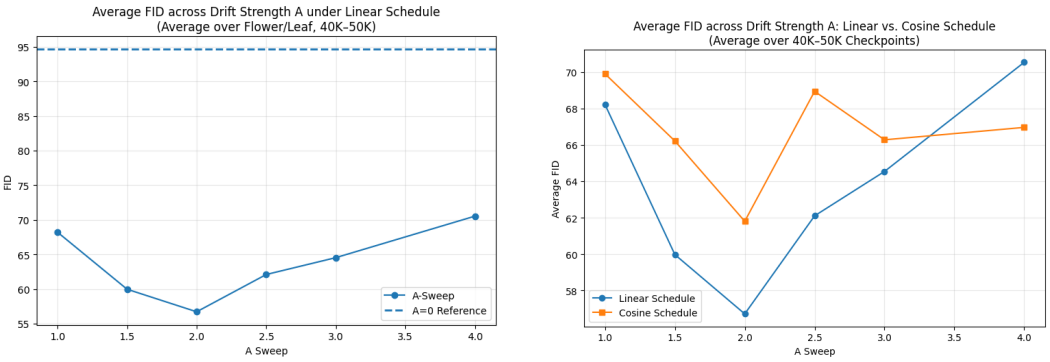
4.1 Experimental Settings

We use the Plant Disease Leaf dataset [Hughes & Salathé, 2015], the Oxford Flowers 102 dataset [Nilsback & Zisserman, 2008], and the CelebA-HQ 128×128 dataset [JJuOn, 2022] with 10,000 male and 10,000 female images. The Leaf/Flower setting is used as a coarse inter-domain conditional generation task, whereas CelebA-HQ gender is used as an intra-domain attribute-control task within the same face domain. Since CelebA-HQ involves a more subtle attribute-level separation, the detailed CelebA-HQ quantitative results are provided in Appendix E.4. All models are trained with a learning rate of 1×10^{-4} and a gradient clipping threshold of 3. For ablation studies, we use 1,000 generated samples per domain. For the main comparison with baseline models, we generate 5,000 samples per domain under training seeds 777 and 2026. For a fair comparison, all DDPM-based baselines and CPDM variants share the same denoising backbone.

4.2 Scalar CPDM: Forward Geometry Design Ablations

In this section, we analyze how changes in forward geometry design affect generation performance in scalar CPDM, while keeping the same compact scalar condition s_z. To this end, we conduct ablation studies by varying the drift strength A, the drift time schedule C_t, and the drift basis $\hat{u}_z$. All ablations in this section were conducted under training seed 777, using 1,000 generated samples per domain.

4.2.1 Effect of Drift Strength A



(a) Linear schedule with drift-off reference.

(b) Linear vs. τ-cosine schedules.

Figure 2: Effect of drift strength A on average FID. FID is averaged over Flower/Leaf checkpoints from 40K to 50K. Left: the linear schedule consistently improves over the drift-off reference and achieves the best result near $A = 2$. Right: both linear and τ-cosine schedules show an optimum around $A = 2$, while larger drift strengths degrade generation quality.

All drift-on settings achieved lower FID than the drift-off baseline. The best performance was observed around $A = 2$, while both smaller and larger drift strengths degraded performance. Under the linear schedule, $A = 2$ achieved the lowest Avg FID of 56.72. Comparing linear and τ-cosine schedules further shows that both time-dependent schedules reached their optimum near $A = 2$, while τ-cosine exhibited more gradual degradation at larger A. This suggests that drift strength controls the balance between terminal separation and reverse reconstruction. Under our normalization $\|\hat{u}_z\|_{\text{avg}} = 0.6$, $A = 2$ corresponds to a maximum average drift norm of approximately 1.2, where the best performance was observed.

4.2.2 Effect of Drift Time Schedule

To isolate the effect of temporal drift design, we compare linear, τ-cosine, and step-constant time schedules with matched terminal scale. Linear achieved the best peak performance with an Avg FID of **56.72**, outperforming τ-cosine (**61.80**) and constant-time scheduling (**72.32**). The step-constant schedule performs worst, indicating that explicit temporal drift accumulation matters for forward geometry formation: matching the terminal magnitude alone is insufficient, and the intermediate trajectory also affects generation quality. Detailed definitions of the drift time schedules, their induced mean-shift trajectories, and the schedule comparison summary are provided in Appendix D.

4.2.3 Effect of Drift Basis $\hat{u}_z$

To further examine whether the performance gain comes from the spatial structure of the forward drift direction, we compare different choices of the image-space basis $\hat{u}$ while fixing the drift strength and time schedule. The fine basis is constructed from the difference between Flower and Leaf prototypes, each obtained by averaging 512 training images from the corresponding endpoint domain. The random basis is sampled from a random image-space direction, and the constant basis is a spatially uniform direction vector. All bases are globally rescaled to the same average per-pixel ℓ_2 norm of 0.6, so the comparison isolates spatial structure rather than drift magnitude.

With $A = 2$ and a linear schedule fixed, we compare fine, random, and constant $\hat{u}$ bases. Fine achieves the best Avg FID of 56.72, while random $\hat{u}$ still improves over the drift-off reference (88.25 vs.

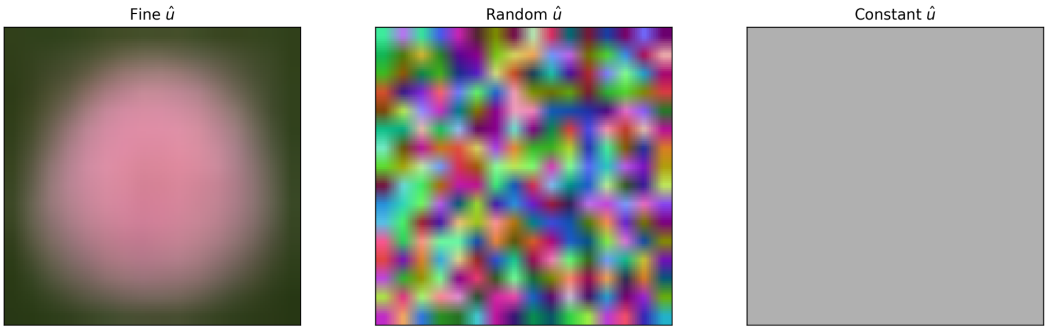


Figure 3: Visualization of image-space drift bases $\hat{u}$ used in the drift-basis ablation. All bases are globally rescaled to the same average per-pixel norm, so the comparison isolates the effect of spatial structure rather than total drift magnitude.

94.61), and constant $\hat{u}$ slightly underperforms it (95.66). These results show that CPDM benefits from spatially structured forward geometry. Fine bases provide the most effective structured directions, and even random $\hat{u}$ can improve over the drift-off setting. In contrast, the spatially uniform constant basis fails to provide the same benefit, suggesting that the proposed drift is not merely a nonzero mean shift but depends on how conditional energy is spatially allocated in the forward process.

4.2.4 Effect of Conditioning Information Placement

Using the drift constructed in the preceding sections, we compare performance when the same conditioning information is supplied to the reverse denoiser of a standard DDPM. Reverse-s_z provides only the scalar condition without forward drift. Reverse-r_t concatenates the complete $128 \times 128 \times 3$ drift map with x_t, without embedding compression, to form a six-channel input, while Reverse-$(r_t + s_z)$ additionally provides the scalar condition. In contrast, CPDM incorporates r_t into the forward state and the η-target without supplying it through separate input channels.

Table 1: Performance averaged over the two endpoint domains and the common 40K, 45K, and 50K checkpoints. Lower is better.

Model	**Avg FID↓**	**Avg KID↓**
Reverse-s_z (drift-off)	94.61	0.07565
Reverse-r_t	94.06	0.07311
Reverse-$(r_t + s_z)$	102.55	0.08213
CPDM-s_z (drift-on)	**56.72**	**0.03877**

Providing the full drift map and scalar to the reverse denoiser did not reproduce CPDM's gains. These results show that, even when the same conditioning information is provided, where it is placed has an important effect on generation performance.

To understand these performance gains, we analyze the relationship between reconstruction and condition-wise separation through one-step $\hat{x}_0$ predictions and complete reverse trajectories. Compared with drift-off, CPDM establishes persistent intended-domain classifier evidence substantially earlier during sampling, with final features closer to their intended real-domain centroids. These findings support the interpretation that forward geometry can improve generation by reducing the denoiser's joint reconstruction–separation burden, even though one-step predictions exhibit higher absolute reconstruction MSE alongside stronger intended-domain classifier margins. Detailed analyses are reported in Appendices F and G.

4.3 Comparison with Standard and Stronger Baselines on Leaf and Flower

Based on the ablation results in Section 4.2, we fix the selected scalar CPDM configuration and compare it with standard and stronger conditional baselines on the Leaf/Flower setting. This comparison evaluates whether the proposed forward-geometry design can remain competitive against higher-dimensional condition embeddings and stronger reverse-time predictors. We compare against

Shift-DDPM [Zhang et al., 2023], CLIP-based conditioning [Radford et al., 2021], and LLaVA-caption conditioning [Liu et al., 2023].

Table 2: Comparison with baselines on the Leaf and Flower datasets. Avg FID/KID average the two endpoint domains, while FID Gap measures their absolute FID difference at the reported best checkpoint. Scores are averaged over training seeds 777 and 2026. $\pm$ denotes half the seed range for FID/FID Gap and the within-run SD averaged over domains and seeds for KID. Lower is better.

Model	**Avg FID↓**	**Avg KID↓**	**FID Gap ↓**	**Best Step**
Standard baselines				
Shift-DDPM	134.27 ± 10.31	$0.1315 \pm .0025$	31.05 ± 8.33	45K
Cond. DDPM (one-hot)	83.76 ± 9.30	$0.0772 \pm .0022$	48.87 ± 30.44	40K
Base CPDM (ours)	$\mathbf{49.68 \pm 5.96}$	$\mathbf{0.0412 \pm .0016}$	27.20 ± 15.59	50K
Stronger baselines				
Shift-DDPM++	126.94 ± 4.39	$0.1211 \pm .0024$	19.61 ± 19.12	35K
Jointly-learned 256D	84.60 ± 2.76	$0.0767 \pm .0023$	39.99 ± 31.67	40K
CLIP 512D image emb.	$\mathbf{29.74 \pm 0.27}$	$\mathbf{0.0184 \pm .0009}$	19.70 ± 0.67	50K
LLaVA-1.5 + CLIP text emb.	76.01 ± 0.02	$0.0686 \pm .0022$	38.26 ± 17.71	40K

Here, FID Gap measures the absolute difference between Flower-domain FID and Leaf-domain FID at the reported best checkpoint, so it reflects whether a model improves average quality by favoring only one endpoint domain.

Base CPDM substantially improves over standard baselines and also outperforms the larger shift-predictor variant Shift-DDPM++, jointly-learned 256D conditioning, and LLaVA-caption + CLIP text embedding, despite using only a compact scalar condition and a fixed drift basis. This suggests that generation quality is not determined only by the richness of the condition input; rather, forward geometry organization can provide an effective generative structure even when the condition signal is reduced to a single scalar coordinate.

However, the CLIP image-embedding baseline remains the strongest baseline in this comparison, highlighting the value of rich image-level semantic features. Nevertheless, adding forward geometry to the same CLIP-conditioned model further improves average endpoint FID/KID without any additional reverse-conditioning input. Detailed results are provided in Appendix H.

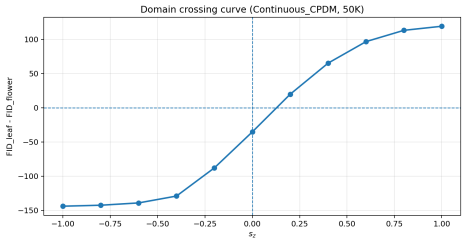
4.4 Quantitative Comparison and Scalar Traversal

4.4.1 Endpoint Comparison: Base CPDM vs. Continuous CPDM

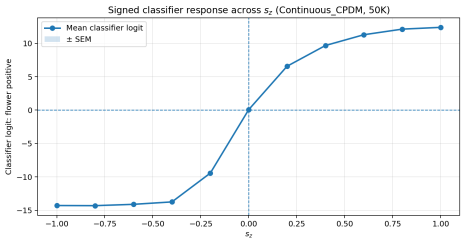
Continuous CPDM achieves an Avg FID of 45.29 on the Leaf/Flower setting under Training seed 777, compared to 43.71 for Base CPDM. While endpoint-domain FID scores remain comparable, Continuous CPDM yields more stable non-collapsed responses along the scalar control axis. A direct comparison and s_z trajectory with Base CPDM under the identical seed is provided in Appendix I.

4.4.2 Monotonic Traversal Analysis

We evaluate Continuous CPDM along $s_z \in \{-1.0, -0.8, \ldots, 0.8, 1.0\}$ using two complementary signals: the signed domain-wise FID difference $\mathrm{FID}_{Leaf}(s_z) - \mathrm{FID}_{Flower}(s_z)$ and a mean signed endpoint-classifier logit (positive for Flower). These measurements are used to examine whether the scalar coordinate produces an ordered traversal between the two endpoint domains. Although Continuous CPDM incurs a small endpoint-fidelity trade-off relative to Base CPDM, the monotonic behavior of both the signed FID-difference curve and the mean signed classifier-logit response provides direct evidence that s_z acts as an ordered distributional control axis. Rather than producing unordered mixtures or endpoint collapse, changing s_z progressively shifts the generated distribution from one endpoint domain toward the other. Nevertheless, this shift is not uniformly smooth across the full interval: the dominant distributional transition occurs around $s_z \in [-0.4, 0.4]$, indicating that CPDM learns an ordered but relatively sharp crossing region between the endpoint distributions.



(a) Signed FID difference vs. s_z



(b) Endpoint classifier response vs. s_z

Figure 4: Traversal analysis for Continuous CPDM across $s_z \in \{-1.0, -0.8, \ldots, 0.8, 1.0\}$. **Left**: Signed FID difference $\mathrm{FID}_{Leaf}(s_z) - \mathrm{FID}_{Flower}(s_z)$ computed using 1,000 generated samples per coordinate. **Right**: Mean signed classifier logit (positive for Flower) estimated by a classifier trained only on real endpoint images and applied to the same 1,000 generated samples; the classifier is used solely for evaluation.

5 Limitations

Our main empirical results focus on inter-domain generation, where a prototype-difference drift basis provides a clear separation direction between visually distinct endpoint domains. However, designing the image-space drift basis $\hat{u}$ becomes more challenging in intra-domain attribute settings. On CelebA-HQ, mixed-basis CPDM outperforms one-hot and jointly learned 256D conditioning, while remaining behind LLaVA-caption + CLIP text embedding and CLIP image embedding. The improvement over the raw-only basis highlights the importance of aligning $\hat{u}$ with the semantic and spatial structure of the target condition. A detailed analysis of this drift-basis behavior is provided in Appendix E.

Coupled multi-axis spinor geometry has not yet been empirically validated. Appendices F and G demonstrate greater condition-wise trajectory separation and earlier persistent intended-domain classifier evidence. However, burden reduction remains an indirectly supported mechanistic hypothesis without direct measurement. Endpoint quality and domain balance vary across training seeds. The trajectory diagnostics and CLIP hybrid results are based on a single training seed.

These observations indicate that improving CPDM's performance and robustness on subtle intra-domain attributes requires more systematic drift-basis design. CPDM has also not yet been validated on larger-scale benchmarks such as ImageNet. These extensions, along with evaluating whether ordered traversal at unseen interior s_z coordinates is preserved when forward geometry is combined with high-capacity conditioning, remain future work.

6 Conclusion

In this work, we proposed CPDM, a conditional diffusion framework that organizes condition-specific geometry through the forward mean. Under compact scalar control, CPDM outperforms several standard and stronger reverse-conditioning baselines. Reverse-sampling analyses show greater condition-wise trajectory separation after drift correction, smaller final target-centroid distances, and earlier persistent intended-domain classifier evidence. These findings support the interpretation that forward geometry can reduce the denoiser's joint reconstruction–separation burden. With pairwise consistency regularization, CPDM also supports ordered scalar traversal without explicit interior-mixture supervision or endpoint interpolation. Together, these results demonstrate the value of forward geometry design for generation quality and conditional trajectory organization under compact scalar control.

Future work will extend CPDM along two directions. First, we will investigate automatic construction of the drift basis $\hat{u}$ by using CLIP image-embedding space to select or mix candidate bases such as $\hat{u}_{\text{diff_raw}}$, $\hat{u}_{\text{diff_lowpass}}$, $\hat{u}_{\text{edge}}$, and $\hat{u}_{\text{common}}$, enabling more transferable drift design across inter-domain and intra-domain settings. Second, we will extend the scalar coordinate to multi-axis personalization by aligning (s_x, s_y, s_z) and $(\hat{u}_x, \hat{u}_y, \hat{u}_z)$ with distinct semantic axes such as gender, bangs, and eyeglasses.

Acknowledgments and Disclosure of Funding

No external funding was received for this work, and the authors declare no competing interests.

We express our sincere gratitude to the pioneers who laid the foundations of diffusion models.

References

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pp. 2256–2265, 2015.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pp. 8162–8171, 2021.

J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. *NeurIPS*, 34:8780–8794, 2021.

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

Z. Zhang, Z. Zhao, J. Yu, and Q. Tian. ShiftDDPMs: Exploring conditional diffusion models by shifting diffusion trajectories. In *AAAI*, vol. 37, no. 3, pp. 3552–3560, 2023.

X. Ding, Y. Wang, K. Zhang, and Z. J. Wang. CCDM: Continuous conditional diffusion models for image generation. *IEEE Trans. Multimedia*, 2026.

A. Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

V. De Bortoli, J. Thornton, J. Heng, and A. Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. *NeurIPS*, 34:17695–17709, 2021.

Y. Xu, Z. Liu, M. Tegmark, and T. Jaakkola. Poisson flow generative models. *NeurIPS*, 35:16782–16795, 2022.

H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *NeurIPS*, 36:34892–34916, 2023.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

D. P. Hughes and M. Salathé. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015.

M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

JJuOn. CelebAHQ-Gender: CelebAHQ with annotations on genders. `https://github.com/JJuOn/CelebAHQ-Gender`, 2022.

E. Schrödinger. Quantisierung als Eigenwertproblem. *Annalen der Physik*, 385(13):437–490, 1926.

E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Physical Review*, 28(6):1049, 1926.

W. Pauli Jr. Zur Quantenmechanik des magnetischen Elektrons. *Zeitschrift für Physik*, 43(9):601–623, 1927.

R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023.

N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

A KL/ELBO Alignment in the DDPM-Equivalent y-World

A.1 Setup and assumptions

In this appendix, we provide a rigorous justification that the proposed drift-augmented forward process remains variationally aligned with standard DDPM after translation to the DDPM-equivalent y-world.

For each fixed condition c, we assume that the drift term $r_t(c)$ is deterministic and depends only on the condition and time index, not on x_t, x_0, or ϵ. For notational convenience, define the corresponding mean-shift term

$$m_t(c) := \sqrt{1-\bar{\alpha}_t}\, r_t(c). \tag{24}$$

Throughout this appendix, $r_t(c)$ denotes the drift in the eta-target or noise-space parameterization, whereas $m_t(c)$ denotes the actual mean shift in x_t-space.

Then the CPDM forward process can be written as

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + m_t(c) + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{25}$$

Equivalently,

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\big(\epsilon + r_t(c)\big). \tag{26}$$

We also extend the cumulative drift schedule by convention with

$$C_0 := 0. \tag{27}$$

Since the drift is defined as

$$r_t(c) = AC_t s_z(c)\hat{u}_z \tag{28}$$

in the scalar case, it follows that

$$r_0(c) = 0 \quad \text{and} \quad m_0(c) = \sqrt{1-\bar{\alpha}_0}\, r_0(c) = 0. \tag{29}$$

Therefore,

$$x_0 = y_0 \tag{30}$$

holds exactly, which will be important for the reconstruction term in the ELBO.

To match the practical sampling procedure

$$y_T \sim \mathcal{N}(0, I), \qquad x_T = y_T + m_T(c), \tag{31}$$

the generative factorization in x-space should be written as

$$p_\theta(x_{0:T} \mid c) = p(x_T \mid c) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t, c), \tag{32}$$

with the conditional prior

$$p(x_T \mid c) = \mathcal{N}\big(m_T(c), I\big). \tag{33}$$

This is the x-space counterpart of the standard Gaussian prior

$$p(y_T) = \mathcal{N}(0, I) \tag{34}$$

in the translated y-world.

Thus, the prior is unconditional in the translated y-world, but condition-dependent in x-space because the practical sampler initializes x_T by translating y_T with $m_T(c)$.

A.2 Forward marginal in the translated y-world

Define the translated variable

$$y_t := x_t - m_t(c). \tag{35}$$

Substituting the CPDM forward expression gives

$$y_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon. \tag{36}$$

Hence the conditional forward marginal in y-space is

$$q_y(y_t \mid x_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, x_0,\, (1-\bar{\alpha}_t)I\big), \tag{37}$$

which is exactly the standard DDPM forward marginal.

By contrast, the forward marginal in x-space is

$$q_x(x_t \mid x_0, c) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, x_0 + m_t(c),\, (1-\bar{\alpha}_t)I\big). \tag{38}$$

Thus CPDM preserves the Gaussian closed form of DDPM, while shifting only the conditional mean geometry through $m_t(c)$.

A.3 KL invariance under condition-fixed translation

For each fixed t and c, consider the translation map

$$T_t^{(c)} : x_t \mapsto y_t = x_t - m_t(c). \tag{39}$$

Because $m_t(c)$ is deterministic, this map is bijective, with inverse

$$\big(T_t^{(c)}\big)^{-1}(y_t) = y_t + m_t(c). \tag{40}$$

Its Jacobian is the identity:

$$J_{T_t^{(c)}} = I, \qquad \left|\det J_{T_t^{(c)}}\right| = 1. \tag{41}$$

Therefore, for any two densities q_x and p_x on x_t, and their pushforwards q_y and p_y under $T_t^{(c)}$, KL divergence is preserved:

$$D_{\mathrm{KL}}(q_x \| p_x) = D_{\mathrm{KL}}(q_y \| p_y). \tag{42}$$

This is the basic reason why the KL terms of CPDM in x-space can be analyzed through the standard DDPM terms in the translated y-world.

A.4 Terminal KL term

At the terminal step,

$$q_x(x_T \mid x_0, c) = \mathcal{N}\big(\sqrt{\bar{\alpha}_T}\, x_0 + m_T(c),\, (1-\bar{\alpha}_T)I\big), \tag{43}$$

while the x-space prior is

$$p(x_T \mid c) = \mathcal{N}\big(m_T(c), I\big). \tag{44}$$

Now apply the translation

$$y_T = x_T - m_T(c). \tag{45}$$

Then,

$$q_y(y_T \mid x_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_T}\, x_0,\, (1-\bar{\alpha}_T)I\big), \tag{46}$$

and

$$p(y_T) = \mathcal{N}(0, I). \tag{47}$$

Since the translation has Jacobian 1,

$$D_{\mathrm{KL}}(q_x(x_T \mid x_0, c) \,\|\, p(x_T \mid c)) = D_{\mathrm{KL}}(q_y(y_T \mid x_0) \,\|\, p(y_T))\,. \tag{48}$$

Therefore, the terminal KL term of CPDM is exactly the standard DDPM terminal KL term in the translated y-world.

A.5 Intermediate posterior correspondence

We next show that the intermediate forward posterior in x-space is a translated version of the standard DDPM posterior in y-space.

Define

$$y_t = x_t - m_t(c), \qquad y_{t-1} = x_{t-1} - m_{t-1}(c). \tag{49}$$

Note that the shifts at t and $t-1$ are generally different, i.e.,

$$m_t(c) \neq m_{t-1}(c). \tag{50}$$

Therefore, the posterior equivalence should be justified through the joint density, rather than by a single-step informal translation argument.

Consider the joint transformation

$$(x_{t-1}, x_t) \mapsto (y_{t-1}, y_t) = \big(x_{t-1} - m_{t-1}(c),\, x_t - m_t(c)\big). \tag{51}$$

Its Jacobian is block-diagonal:

$$J = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \qquad |\det J| = 1. \tag{52}$$

Hence,

$$q_x(x_{t-1}, x_t \mid x_0, c) = q_y(y_{t-1}, y_t \mid x_0). \tag{53}$$

Similarly,

$$q_x(x_t \mid x_0, c) = q_y(y_t \mid x_0). \tag{54}$$

Therefore,

$$\begin{aligned} q_x(x_{t-1} \mid x_t, x_0, c) &= \frac{q_x(x_{t-1}, x_t \mid x_0, c)}{q_x(x_t \mid x_0, c)} \\ &= \frac{q_y(y_{t-1}, y_t \mid x_0)}{q_y(y_t \mid x_0)} \\ &= q_y(y_{t-1} \mid y_t, x_0). \end{aligned} \tag{55}$$

Since the posterior in the y-world is exactly the standard DDPM posterior,

$$q_y(y_{t-1} \mid y_t, x_0) = \mathcal{N}\big(y_{t-1}; \tilde{\mu}_t(y_t, x_0), \tilde{\beta}_t I\big). \tag{56}$$

Thus, the intermediate posterior of CPDM is a condition-fixed translated version of the standard DDPM posterior.

A.6 Reverse parameterization with η-prediction

In CPDM, the forward stochastic term is not ϵ alone but

$$\eta := \epsilon + r_t(c). \tag{57}$$

The network is trained to predict

$$\hat{\eta}_\theta(x_t, t, c) \approx \eta. \tag{58}$$

Because $r_t(c)$ is deterministic for each fixed condition c, we can recover the standard DDPM noise prediction by

$$\hat{\epsilon}_\theta(x_t, t, c) = \hat{\eta}_\theta(x_t, t, c) - r_t(c). \tag{59}$$

In the translated y-world, the reverse Gaussian can therefore be parameterized in the standard DDPM form:

$$p_\theta^{(y)}(y_{t-1} \mid y_t, c) = \mathcal{N}\Big(y_{t-1}; \mu_\theta^{(y)}(y_t, t, c), \sigma_t^2 I\Big), \tag{60}$$

with mean

$$\mu_\theta^{(y)}(y_t, t, c) = \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon}_\theta(x_t, t, c) \right). \tag{61}$$

Substituting $\hat{\epsilon}_\theta = \hat{\eta}_\theta - r_t(c)$ yields:

$$\mu_\theta^{(y)}(y_t, t, c) = \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \big(\hat{\eta}_\theta(x_t, t, c) - r_t(c)\big) \right). \tag{62}$$

Returning to the original x-space gives

$$x_{t-1} = y_{t-1} + m_{t-1}(c), \tag{63}$$

and hence

$$p_\theta^{(x)}(x_{t-1} \mid x_t, c) = \mathcal{N}\Big(x_{t-1}; \mu_\theta^{(x)}(x_t, t, c), \sigma_t^2 I\Big), \tag{64}$$

where

$$\mu_\theta^{(x)}(x_t, t, c) = \mu_\theta^{(y)}(x_t - m_t(c), t, c) + m_{t-1}(c). \tag{65}$$

Therefore, CPDM does not require a new reverse-time Gaussian form. Instead, it reuses the standard DDPM reverse parameterization in the translated y-world and maps the result back to x-space.

A.7 ELBO equivalence

We now show that the CPDM variational objective in x-space is equivalent to the standard DDPM objective in the translated y-world.

The conditional ELBO in x-space can be written as

$$\begin{aligned}\mathcal{L}_x(x_0, c) := \mathbb{E}_{q_x}\Big[-\log p_\theta^{(x)}(x_0 \mid x_1, c)\Big] \\ + \sum_{t=2}^{T} D_{\mathrm{KL}}\Big(q_x(x_{t-1} \mid x_t, x_0, c) \,\|\, p_\theta^{(x)}(x_{t-1} \mid x_t, c)\Big) \\ + D_{\mathrm{KL}}(q_x(x_T \mid x_0, c) \,\|\, p(x_T \mid c))\,. \end{aligned} \tag{66}$$

By Section A.4, the terminal KL term is identical to the standard DDPM terminal KL term in the y-world:

$$D_{\mathrm{KL}}(q_x(x_T \mid x_0, c) \,\|\, p(x_T \mid c)) = D_{\mathrm{KL}}(q_y(y_T \mid x_0) \,\|\, p(y_T))\,. \tag{67}$$

By Section A.5 and the same Jacobian-1 translation argument, each intermediate KL term satisfies

$$\begin{aligned} D_{\mathrm{KL}}\Big(q_x(x_{t-1} \mid x_t, x_0, c) \,\|\, p_\theta^{(x)}(x_{t-1} \mid x_t, c)\Big) \\ = D_{\mathrm{KL}}\Big(q_y(y_{t-1} \mid y_t, x_0) \,\|\, p_\theta^{(y)}(y_{t-1} \mid y_t, c)\Big)\,. \end{aligned} \tag{68}$$

Finally, because $C_0 = 0$ implies $r_0(c) = 0$ and hence $m_0(c) = 0$, we have

$$x_0 = y_0. \tag{69}$$

Therefore the reconstruction term is also unchanged under the translation:

$$\mathbb{E}_{q_x}\Big[-\log p_\theta^{(x)}(x_0 \mid x_1, c)\Big] = \mathbb{E}_{q_y}\Big[-\log p_\theta^{(y)}(x_0 \mid y_1, c)\Big]\,. \tag{70}$$

Combining the three parts, we obtain

$$\mathcal{L}_x(x_0, c) = \mathcal{L}_y(x_0, c), \tag{71}$$

where $\mathcal{L}_y$ is the standard DDPM variational objective in the translated y-world. Hence CPDM preserves the Gaussian tractability and ELBO/KL structure of DDPM under the deterministic condition-fixed translation induced by $m_t(c)$.

A.8 Scope of the proof and relation to the Schrödinger-inspired coordinate system

The proof above relies only on the following facts:

1. for each fixed condition c, the drift $r_t(c)$ is deterministic,
2. the forward perturbation remains Gaussian after translation,
3. the map $x_t \leftrightarrow y_t$ is a bijective translation with unit Jacobian.

Therefore, the KL/ELBO alignment result does not depend on integrating any additional latent-time Schrödinger dynamics. The role of the Schrödinger-inspired spinor-component formalism in this work is instead to provide a bounded and interpretable coordinate construction from which the condition-dependent drift is built.

In this sense, the mathematical contribution of CPDM is not to introduce a new stochastic family beyond DDPM, but to reorganize conditional generation as a deterministic mean-geometry shift in the forward process while preserving the standard DDPM variational machinery in the translated y-world.

B Practical Sampling Algorithm

Algorithm 1 summarizes practical reverse sampling of CPDM in the DDPM-equivalent y-world. At each timestep, the model first predicts η in the original x-space, recovers the corresponding DDPM noise prediction, performs the standard DDPM reverse update in the translated y-world, and maps the result back to the original x-space.

Algorithm 1 Sampling of CPDM in the DDPM-equivalent y-world

Require: Trained network $\hat{\eta}_\theta$, condition c

Sample $y_T \sim \mathcal{N}(0, I)$

Set $x_T = y_T + \sqrt{1-\bar{\alpha}_T}\, r_T(c)$

for $t = T, T-1, \ldots, 1$ **do**

  Compute $r_t(c)$

  Predict the drift-augmented target:

$$\hat{\eta}_\theta(x_t, t, c)$$

  Recover the DDPM noise prediction:

$$\hat{\epsilon}_\theta = \hat{\eta}_\theta(x_t, t, c) - r_t(c)$$

  Move to the translated world:

$$y_t = x_t - \sqrt{1-\bar{\alpha}_t}\, r_t(c)$$

  Compute the DDPM mean in the y-world:

$$\mu_\theta(y_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(y_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\epsilon}_\theta \right)$$

  Sample $z \sim \mathcal{N}(0, I)$ if $t > 1$, and set $z = 0$ if $t = 1$

  Update

$$y_{t-1} = \mu_\theta(y_t, t) + \sigma_t z$$

  Return to the original world:

$$x_{t-1} = y_{t-1} + \sqrt{1-\bar{\alpha}_{t-1}}\, r_{t-1}(c)$$

end for

return x_0

C Training and Evaluation Protocol

C.1 Dataset preprocessing

All images are resized to 128×128 resolution before training and evaluation. For the main Leaf/Flower experiments, we use the Plant Disease Leaf dataset and the Oxford Flowers 102 dataset as the two endpoint domains. This setting is used as the primary benchmark for evaluating whether forward geometry formation improves conditional generation under compact scalar control.

For the additional CelebA-HQ validation experiments reported in the appendix, we use the CelebAHQ-Gender annotation dataset, which provides male/female labels for CelebA-HQ images. The original CelebA-HQ dataset contains 30,000 high-quality face images, and the CelebAHQ-Gender repository provides gender annotations obtained through classifier-based labeling followed by manual inspection. The annotated dataset contains 19,057 female images and 10,943 male images. From this dataset, we sort the image files deterministically and use 10,000 images per gender class to construct a balanced male/female training set.

The same preprocessing pipeline is applied across CPDM and all DDPM-based baselines. This ensures that the comparison focuses on the effect of condition placement and forward-geometry design, rather than differences in data resolution, class balance, or preprocessing.

C.2 Training setup

All models are trained using the Adam optimizer with a learning rate of 1×10^{-4}, global-norm gradient clipping with threshold 3, and a batch size of 32. We use mixed-precision training with a loss-scaled optimizer. Unless otherwise stated, all DDPM-based baselines and CPDM variants share the same denoising backbone. This fixed-backbone setting is used to isolate the effect of the proposed forward drift and conditioning strategy, rather than architectural differences.

For the Leaf/Flower experiments, models are trained up to 50K steps. We evaluate checkpoints from 30K to 50K at 5K-step intervals. For the additional CelebA-HQ validation experiments, models are trained up to 90K steps to allow more detailed intra-domain attribute learning. We evaluate checkpoints from 50K to 90K at 10K-step intervals.

Training was performed on an NVIDIA L4 GPU. The Leaf/Flower experiments required approximately 20 hours per run, while the CelebA-HQ experiments required approximately 35 hours per run. Sampling was performed on an NVIDIA A100 GPU and required approximately 70 minutes per 1,000 generated samples.

C.3 Checkpoint selection

For each model, we first train and evaluate checkpoints using training seed 777. Within the predefined evaluation range, we select the best-performing checkpoint step according to Avg FID over the two endpoint domains. In the Leaf/Flower setting, the checkpoint candidates are selected from 30K, 35K, 40K, 45K, and 50K steps. In the additional CelebA-HQ validation setting, the checkpoint candidates are selected from 50K, 60K, 70K, 80K, and 90K steps.

After the best checkpoint step is identified using seed 777, we train the same model again with an independent training seed, 2026. For this second run, we evaluate the checkpoint at the same selected step rather than reselecting the best checkpoint independently. This protocol tests whether the performance observed at the selected training step is maintained under a different random initialization and training trajectory.

The same checkpoint-selection protocol is applied consistently across CPDM and all baseline models within each dataset setting.

C.4 FID/KID evaluation

We evaluate generation quality using FID and KID. For the main comparison with baseline models, we generate 5,000 samples per domain from each model's selected (best-step) checkpoint. For ablation studies, we use 1,000 generated samples per domain unless otherwise specified.

To avoid variation caused by repeatedly estimating real-data statistics, we precompute the real-domain feature statistics from the full training set of each domain and store them as NPZ files. During evaluation, generated samples are compared against the corresponding fixed real-domain statistics. This makes the FID computation consistent across different checkpoints, models, and training seeds.

For the main reported results, we compute domain-wise FID/KID for the seed-777 selected checkpoint and for the independently trained seed-2026 model at the same checkpoint step. The final reported scores are averaged across these two training seeds. KID is computed under the same domain-wise evaluation protocol, using the same real/generated domain pairing.

C.5 Seed protocol

We use two independent training seeds, 777 and 2026, to evaluate whether the observed performance is robust to different model initializations and training trajectories. Training seed 777 is used for checkpoint selection: each model is evaluated over the predefined checkpoint range, and the best checkpoint step is selected according to Avg FID over the two endpoint domains.

Training seed 2026 is then used as an independent validation run. The model is trained from a different initialization, and evaluation is performed at the same checkpoint step selected from seed 777. Thus, seed 2026 is not used to select a new best checkpoint, but to test whether the performance at the seed-777 selected step remains stable under an independent training run.

For the Leaf/Flower experiments, this protocol is applied consistently to CPDM and all baseline models in the main comparison. For the additional CelebA-HQ validation experiments, the same protocol is applied to a balanced subset of 10,000 male and 10,000 female images selected by deterministic file sorting. This is especially important for CelebA-HQ, where gender conditioning corresponds to a subtle intra-domain attribute separation rather than a coarse cross-domain separation.

D Time Scheduling and r_t Trajectory

In CPDM, the conditional drift is not applied as a static perturbation independent of time. Instead, its strength follows a time-dependent cumulative trajectory C_t, which determines how strongly the condition-dependent drift is expressed at each diffusion step. For the scalar drift setting, we define

$$r_t(c) = A\, C_t\, s_z(c)\, \hat{u}_z, \tag{72}$$

where A controls the overall drift strength, $s_z(c) \in [-1, 1]$ is the signed condition coordinate, and $\hat{u}_z$ is a fixed image-space drift direction. The schedule $C_t \in [0, 1]$ satisfies $C_0 = 0$ and $C_T = 1$, so that the drift is absent near the clean data endpoint and fully expressed near the terminal noise endpoint.

Equivalently, the drift trajectory can be written in terms of discrete increments

$$\tau_t = C_t - C_{t-1}, \qquad C_t = \sum_{k=0}^{t} \tau_k. \tag{73}$$

Thus, τ_t controls how fast the drift trajectory moves at each timestep, while C_t controls the accumulated drift position. The actual mean displacement in the forward distribution is

$$\sqrt{1-\bar{\alpha}_t}\, r_t(c) = \sqrt{1-\bar{\alpha}_t}\, A\, C_t\, s_z(c)\, \hat{u}_z. \tag{74}$$

Therefore, the effective condition separation is determined by both the DDPM noise scale $\sqrt{1-\bar{\alpha}_t}$ and the drift schedule C_t. For two opposite conditions $s_z = +1$ and $s_z = -1$, the drift-induced mean separation becomes

$$\Delta\mu_t = 2\sqrt{1-\bar{\alpha}_t}\, A\, C_t\, \hat{u}_z. \tag{75}$$

This shows that the time schedule does not merely change the magnitude of drift, but directly controls the trajectory by which conditional terminal geometry is formed.

D.1 Cosine-τ Schedule

The cosine-τ schedule uses a smooth cumulative trajectory:

$$\tilde{C}_t = \frac{1 - \cos\left(\pi \dfrac{t}{T}\right)}{2}, \tag{76}$$

followed by discrete differencing and normalization:

$$\tau_t = \tilde{C}_t - \tilde{C}_{t-1}, \qquad C_t = \sum_{k=0}^{t} \tau_k. \tag{77}$$

In implementation, a small lower bound τ_0 is applied to nonzero increments before renormalization, preventing extremely small update regions from becoming numerically inactive. This schedule produces a smooth S-shaped cumulative trajectory: the drift changes slowly near the endpoints and more strongly in the middle timesteps.

This design is useful because reverse sampling proceeds from $t = T$ to $t = 0$. Therefore, the early reverse stage begins from a strongly separated terminal geometry, while the late reverse stage gradually reduces the drift effect, allowing the denoiser to focus more on fine-grained image restoration. In this sense, cosine-τ scheduling separates the role of drift into two phases: terminal geometry organization at high-noise steps and detail restoration at low-noise steps.

D.2 Linear Schedule

The linear schedule is defined as

$$C_t = \frac{t}{T}, \qquad \tau_t = C_t - C_{t-1}. \tag{78}$$

Unlike the cosine-τ schedule, the linear schedule changes the accumulated drift at a constant rate. This provides a simple baseline for testing whether the temporal shape of the drift trajectory matters. If linear scheduling performs similarly to cosine-τ, then the main benefit may come from terminal separation itself. If cosine-τ performs better, then the result suggests that the intermediate trajectory, not only the final terminal drift, is important.

D.3 Step-Constant Schedule

The step-constant schedule is defined as

$$C_0 = 0, \qquad C_t = 1 \quad \text{for } t > 0. \tag{79}$$

Equivalently,

$$\tau_1 = 1, \qquad \tau_t = 0 \quad \text{for } t > 1. \tag{80}$$

This schedule immediately applies the full drift after the first timestep and keeps it constant throughout almost the entire trajectory. Therefore, it removes gradual trajectory formation and tests whether simply adding a nonzero terminal drift is sufficient.

Compared with the cosine-τ and linear schedules, the step-constant schedule is a stronger but less flexible intervention. It maximizes condition separation early, but because the full drift remains active even in low-noise regions, it may interfere with fine-detail reconstruction during the late reverse process. Thus, this schedule serves as an ablation for distinguishing terminal separation from temporally controlled trajectory formation.

D.4 Interpretation

The three schedules compare different hypotheses about how conditional drift should be introduced.

Table 3: Summary of drift time schedules and their main roles.

Schedule	C_t **behavior**	**Main role**
Cosine-τ	Smooth nonlinear accumulation	Separates terminal geometry while reducing late-stage interference
Linear	Uniform accumulation	Tests whether a simple gradual trajectory is sufficient
Step-constant	Immediate full drift	Tests whether terminal drift alone is enough

The important distinction is that CPDM does not only introduce a condition-dependent shift $r_T(c)$ at the terminal point. Instead, it defines an entire trajectory $r_t(c)$ over diffusion time. Therefore, different time schedules can produce the same final drift strength $r_T(c)$, while inducing different intermediate paths. This is important because the denoiser is trained and sampled across all timesteps, not only at $t = T$. As a result, the temporal formation of the drift can affect both conditional separation and reconstruction fidelity.

In our experiments, this schedule ablation is used to verify whether the performance gain comes merely from applying a nonzero drift, or from forming a structured forward geometry over time. If the cosine or linear schedule outperforms the step-constant schedule, it suggests that the trajectory by which the conditional geometry is formed matters, rather than only the final magnitude of $r_T(c)$.

D.5 Forward Trajectory under Different Time Schedules

We resize $\hat{u}_z$ to $128 \times 128 \times 3$ and apply a global scaling factor such that its average pixel-wise ℓ_2 norm becomes 0.6. With $A = 2$ and $|s_z| = 1$, we plot the effective mean displacement $\left\|\sqrt{1-\bar{\alpha}_t}\, r_t\right\|_{\text{avg}}$ during reverse sampling. Unlike the raw r_t trajectory, this curve includes the DDPM noise scale $\sqrt{1-\bar{\alpha}_t}$, and therefore directly represents the magnitude of the condition-dependent forward mean shift.

D.6 Performance Comparison by Drift Time Schedule

All settings use the same drift strength $A = 2$ and the same drift direction $\hat{u}_z$, while only the temporal schedule C_t is changed. The reported value is the average FID score computed from the Flower and Leaf domains.

The linear schedule achieves the lowest average FID of **56.72**. The cosine-τ schedule obtains an average FID of **61.80**, while the step-constant schedule shows the worst performance at **72.32**.

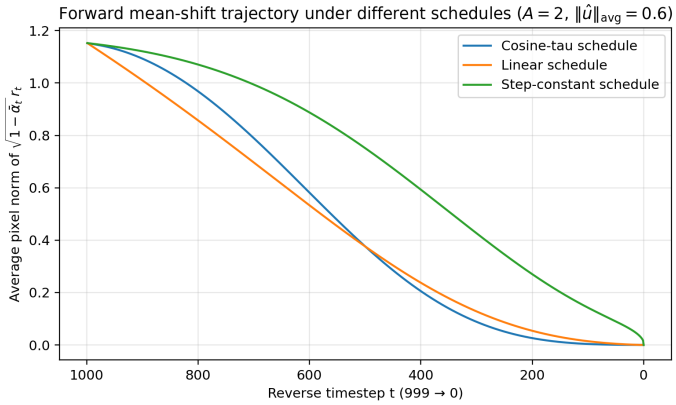


Figure 5: Forward mean-shift trajectory induced by different drift time schedules ($A = 2$, $\|\hat{u}_z\|_{\text{avg}} = 0.6$). The y-axis shows the effective mean displacement $\left\|\sqrt{1-\bar{\alpha}_t}\, r_t\right\|_{\text{avg}}$ during reverse sampling (t=999 $\rightarrow$ 0), which includes the DDPM noise scale and therefore directly represents the magnitude of the condition-dependent forward mean shift.

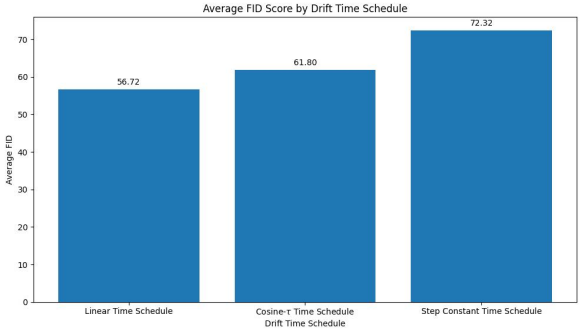


Figure 6: Average FID by drift time schedule (Flower/Leaf). Lower is better. The linear schedule achieves the best average FID (56.72), followed by cosine-τ (61.80). The step-constant schedule performs worst (72.32), suggesting that maintaining the full drift throughout most of the trajectory can degrade generation quality.

This result indicates that the final drift magnitude alone is not sufficient to explain the generation quality. Although all schedules reach the same terminal drift strength, they produce different intermediate trajectories during the reverse process. The linear schedule provides the most stable trajectory in this setting, suggesting that a gradual and uniform reduction of the drift effect during denoising better balances conditional separation and image reconstruction.

In contrast, the step-constant schedule applies the full drift for most of the trajectory. This produces strong condition separation, but it can also interfere with fine image restoration in later denoising steps. Therefore, the result supports the view that CPDM benefits not only from terminal geometry separation, but also from the way the drift-induced geometry is formed and relaxed over time.

E Types of $\hat{u}$ basis and their spatial energy characteristics

E.1 Analysis of $\hat{u}_z$ type and FID score

To examine how the choice of drift basis affects forward geometry formation, we compare several types of $\hat{u}$ bases: raw-only, lowpass-only, and their convex combinations. In all cases, the basis is projected to image space at 128×128 resolution and globally rescaled so that the average pixel-wise ℓ_2 norm is fixed at 0.6. This allows the comparison to focus on spatial energy distribution rather than total magnitude.

Table 4: Quantitative comparison of different $\hat{u}$ basis types under a fixed drift strength. All results are computed using 1,000 generated samples per domain, and the best checkpoint is selected by the lowest Avg FID within each dataset setting. Lower FID/KID is better. The comparison evaluates the effect of spatial energy allocation, but does not fully decouple basis design from the optimal choice of $A = 2$.

Dataset	$\hat{u}$ **basis**	**Avg FID** ↓	**Avg KID** ↓	**FID gap** ↓	**Best Step**
Flower/Leaf	Raw only (fine $\hat{u}$)	**51.94**	**.0344**	**11.39**	50K
Flower/Leaf	Lowpass only	67.09	.0465	30.04	40K
Flower/Leaf	$0.7\hat{u}_{raw} + 0.3\hat{u}_{lp}$	56.84	.0381	18.12	50K
Flower/Leaf	$0.3\hat{u}_{raw} + 0.7\hat{u}_{lp}$	69.77	.0484	40.59	40K
CelebA-HQ	Raw only (fine $\hat{u}$)	56.88	.0521	2.75	90K
CelebA-HQ	Lowpass only	55.66	.0530	6.85	70K
CelebA-HQ	$0.7\hat{u}_{raw} + 0.3\hat{u}_{lp}$	**49.07**	**.0437**	**2.43**	70K
CelebA-HQ	$0.3\hat{u}_{raw} + 0.7\hat{u}_{lp}$	54.26	.0500	4.55	70K

Table 4 suggests that mixed bases provide a robust compromise across inter-domain and intra-domain settings. For Flower/Leaf, the raw-only basis performs best, showing that direct prototype differences are effective for coarse domain separation. Nevertheless, the $0.7\hat{u}_{raw} + 0.3\hat{u}_{lp}$ basis is the second-best configuration, indicating that moderate low-pass smoothing preserves most of the useful domain-separating structure. For CelebA-HQ, the same mixed basis achieves the best Avg FID, suggesting that intra-domain facial attribute control benefits from smoothing excessive high-frequency or identity-sensitive drift energy while retaining condition-relevant raw structure. The next subsections analyze this trend through the corresponding heatmap patterns.

E.2 Leaf and Flower $\hat{u_z}$ Heatmap Analysis

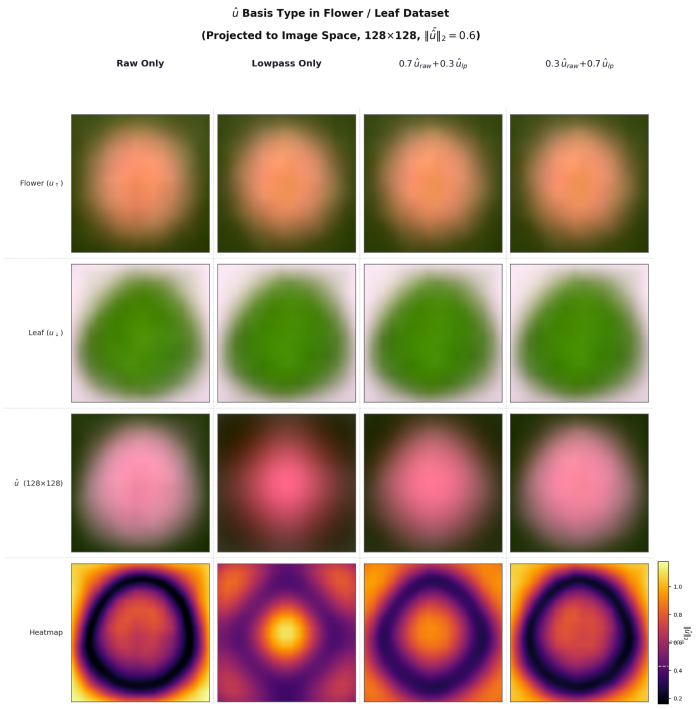


Figure 7: Comparison of $\hat{u}$ basis types in the Flower/Leaf dataset. The raw-only, lowpass-only, and mixed bases are all projected to 128×128 image space and globally normalized to the same average pixel-wise ℓ_2 norm.

The figures show that different basis constructions preserve different geometric properties. The raw-only basis retains more of the original domain difference structure, including sharper spatial contrast and stronger boundary-related variation. The lowpass-only basis suppresses high-frequency fluctuations and instead emphasizes broad, smooth, coarse-scale structure. The mixed bases interpolate between these two extremes, preserving part of the raw structural contrast while maintaining the smoother large-scale organization induced by the lowpass component.

A useful way to interpret these differences is through the heatmaps of $\|\hat{u}_{h,w,:}\|_2$, which reveal where the drift energy is concentrated. In the Flower/Leaf example, the raw-only basis produces a strong ring-like or boundary-oriented energy profile, indicating that the basis emphasizes global object-versus-background contrast and shape boundary information. By contrast, the lowpass-only basis yields a more centrally concentrated energy profile, reflecting smoother coarse object structure. The mixed bases produce intermediate distributions, where coarse object-level geometry remains visible while excessive concentration on a single region is moderated.

E.3 CelebA-HQ Gender $\hat{u_z}$ Heatmap Analysis

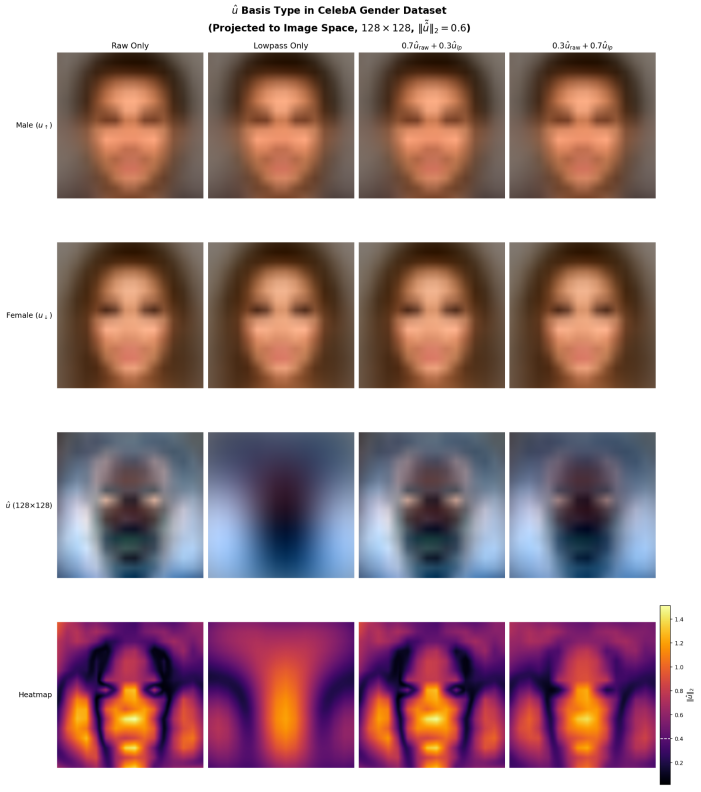


Figure 8: Comparison of $\hat{u}$ basis types in the CelebA-HQ gender dataset. Unlike Flower/Leaf, gender separation is an intra-domain attribute-control problem within the shared face manifold, and the drift energy is distributed over facial contour, hair, and peripheral support regions rather than forming a simple object/background separation.

In the CelebA-HQ gender example, the interpretation is more subtle. Unlike Flower/Leaf, male versus female is not a cross-domain separation between fundamentally different object classes, but rather an attribute-level separation within the same overall face domain. Accordingly, the drift energy tends to be distributed less as a simple object/background contrast and more as an imbalanced spatial pattern over facial contour, hair region, and peripheral facial support regions. In this case, the heatmap often shows relatively stronger energy allocation in outer or side regions, while the central facial region may exhibit comparatively reduced drift energy.

E.4 Comparison with Baselines on CelebA-HQ

Table 5: Quantitative comparison with baselines on CelebA-HQ gender under training seed 777. All results are evaluated using 5,000 generated samples per domain, and the reported checkpoint is selected by the lowest Avg FID over the CelebA-HQ endpoint domains. Lower FID/KID is better.

Model	Avg FID ↓	Avg KID ↓	FID gap ↓	Best Step
Cond. DDPM (one-hot)	46.11	.0474	7.20	90K
Jointly-learned 256D	44.03	.0452	10.21	90K
CPDM ($1.0\hat{u}_{raw}$)	48.34	.0506	0.52	90K
CPDM ($0.7\hat{u}_{raw} + 0.3\hat{u}_{lp}$)	40.83	.0429	3.23	70K
Shift-DDPM++	70.92	.0798	0.55	50K
LLaVA-1.5 + CLIP text emb.	35.50	.0334	8.05	90K
CLIP 512D image emb.	22.14	.0186	4.22	90K

Table 5 reports the CelebA-HQ baseline comparison under the 5K-sample evaluation protocol. The mixed-basis CPDM variant outperforms one-hot conditioning, jointly-learned 256D conditioning, and Shift-DDPM++, showing that forward geometry design can remain competitive in an intra-domain attribute-control setting. However, LLaVA-caption with CLIP text embedding and CLIP image embedding achieve lower FID, indicating that high-capacity semantic conditioning remains stronger for subtle face-attribute control.

This phenomenon is important for interpreting the role of the drift basis. When the task is to separate different semantic domains, a strong central or object-aligned drift may directly support coarse class-level separation. However, when the task is to separate attributes within the same domain, over-concentrating the drift on fine central facial details may be undesirable. Under the $\epsilon + r_t$ target, such a basis may force the forward drift to encode identity-sensitive facial details, which can interfere with reconstruction fidelity or induce overly rigid separation. Instead, a basis whose energy is more concentrated around outer facial or peripheral regions may serve as a coarse geometric bias, while leaving fine central facial details to be resolved by the denoising network through η-target prediction.

In this sense, reduced central energy should not necessarily be interpreted as a weakness of the basis. Rather, it may indicate an appropriate division of labor: the forward drift handles the large-scale conditional geometry, while the reverse denoiser refines the residual fine-grained details. For same-domain attribute separation, such as CelebA-HQ gender, this distinction can be especially relevant because many discriminative cues are subtle and entangled with identity-preserving facial structure. By leaving some of these details as a secondary burden for η-target prediction, the model may achieve a more stable balance between conditional separability and image realism.

Overall, these observations suggest that the design of $\hat{u}$ should be understood not merely as choosing a direction of mean shift, but as selecting a spatial allocation strategy for condition-dependent forward geometry. Raw, lowpass, and mixed bases induce different geometric priors in the forward process, and their suitability may depend on whether the target condition corresponds to coarse cross-domain separation or subtle intra-domain attribute control.

These results also suggest a broader forward-design space beyond the fixed-basis comparison considered here. Because the spatial structure of $\hat{u}$ interacts with the drift strength A and the $\epsilon + r_t$ target, alternative designs such as a smoother common basis with stronger drift strength may form different and potentially more transferable forward geometries. Exploring such basis-strength combinations remains an interesting direction for future work.

F One-Step $\hat{x}_0$ Prediction Diagnostics

Section 4.2.4 showed that incorporating conditioning information into forward geometry yielded better generation performance than supplying that information directly to the reverse denoiser. This appendix and Appendix G examine this difference through the relationship between reconstruction and condition-wise separation. The central question is whether condition-specific forward placement can reduce the denoiser's burden of performing these two tasks together during the reverse process.

As a first step, we compare one-step predictions from drift-off ($A = 0$) and CPDM ($A = 2$) using matched real images and noise. Measuring reconstruction error and intended-domain evidence together, we observe that CPDM produces stronger intended-domain evidence at intermediate and high noise levels despite higher absolute reconstruction error. This finding shows that reconstruction error and domain evidence can change in different ways, motivating a joint analysis that distinguishes their respective behavior.

Evaluation setup and metrics. We evaluate $t \in \{100, 300, 500, 700, 900\}$ using paired real images x_0 and noise, with two evaluation seeds. These seeds vary evaluation inputs rather than model training. We use the existing 30K, 40K, and 50K checkpoints, keeping the evaluation images, stateless noise, timesteps, and frozen classifier fixed across checkpoint comparisons. No additional training is performed.

Absolute $\hat{x}_0$ MSE measures the error between the unclipped clean-image prediction and its original image. We use the prediction target η_t from Section 3.4, which reduces to ϵ when $A = 0$ because the drift in Eq. (13) vanishes. Target normalized mean squared error (NMSE) measures prediction error relative to target energy. We normalize the error separately for each sample and then average:

$$\text{Target NMSE}(t) = \frac{1}{N} \sum_{i=1}^{N} \frac{\|\hat{\eta}_{t,i} - \eta_{t,i}\|_2^2}{\|\eta_{t,i}\|_2^2}. \tag{81}$$

Here, N is the number of evaluated images, $\eta_{t,i}$ is the prediction target for sample i, and $\hat{\eta}_{t,i}$ denotes the corresponding network prediction under the evaluated model. Lower values indicate smaller prediction error relative to target energy. Because the two models use different prediction targets, this metric complements the absolute $\hat{x}_0$ reconstruction error.

Domain accuracy measures the proportion of reconstructed images assigned to the correct domain by the frozen classifier; true margin measures the logit difference between the correct and opposite domains.

Table 6: One-step diagnostics on a balanced evaluation set containing 1,000 Flower and 1,000 Leaf images, averaged over the 30K, 40K, and 50K checkpoints.

A	t	Absolute $\hat{x}_0$ MSE↓	Target NMSE↓	Domain accuracy (%)↑	True margin↑
0	100	0.003421	0.115377	99.70	12.0615
2	100	0.003447	0.115597	99.85	12.1159
0	300	0.010274	0.037593	98.68	10.5377
2	300	0.010402	0.036177	98.85	10.6406
0	500	0.018372	0.017803	98.20	9.4118
2	500	0.018695	0.015829	98.33	9.4828
0	700	0.031630	0.007998	96.68	7.6227
2	700	0.032871	0.006477	97.22	7.8222
0	900	0.067698	0.002391	88.90	3.6067
2	900	0.075456	0.001815	90.18	3.6911

Prediction error and target energy. CPDM exhibits higher absolute $\hat{x}_0$ reconstruction error, with a larger difference at high noise levels. Its structured prediction target also has greater energy. At $t = 900$ in the 50K checkpoint, target energy is approximately 47% higher than under drift-off, while absolute target MSE increases by approximately 18%. At the same checkpoint, Target NMSE decreases by approximately 3.9%, 10.9%, 18.4%, and 19.9% at $t = 300, 500, 700, 900$, respectively. Prediction–target cosine similarity also improves at intermediate and high noise levels.

Thus, the model predicts a larger structured perturbation with lower relative error, despite the increase in absolute clean-image reconstruction error. Because the normalization uses different target energies, relative and absolute errors describe distinct aspects of prediction.

Domain evidence across checkpoints. The frozen classifier reveals stronger intended-domain evidence at intermediate and high noise levels. At $t = 700$ in the 50K checkpoint, balanced accuracy

increases from 95.90% to 97.25%, and true margin increases from 7.71 to 8.02. Lower target-normalized MSE and higher prediction–target cosine similarity are also observed at the 30K and 40K checkpoints. In the checkpoint averages at $t = 900$, Target NMSE decreases by approximately 24.1%, while balanced accuracy increases by 1.28 percentage points (Table 6).

These observations show that stronger domain evidence can coexist with higher absolute reconstruction error. Classifier metrics vary across checkpoints, so these results do not imply monotonic improvement with training.

Connection to the drift-strength ablation. Increasing A enlarges the separation between condition-dependent forward means and increases the magnitude of the structured prediction target. The $A = 0$ versus $A = 2$ diagnostics reveal a trade-off between forward placement and absolute reconstruction error, while showing stronger domain evidence and lower target-relative error under $A = 2$.

This pattern is consistent with Section 4.2.1, where performance improves around $A = 1$–2 and deteriorates at $A = 3$–4. These observations are consistent with a balance between condition-specific placement and the demands of structured-target prediction. The explanation of degradation at $A = 3$–4 remains an interpretation of the strength ablation, since the present diagnostics evaluate only $A = 0$ and $A = 2$.

Transition to reverse trajectories. These one-step results establish that stronger intended-domain evidence can coexist with higher absolute reconstruction error in predictions from forward-corrupted real images. The remaining question is whether this difference also appears during actual generation and, if so, when intended-domain evidence becomes persistent.

Appendix G therefore follows the full reverse process, examining condition-specific trajectory geometry, proximity to real-domain centroids, and sample-level persistent commitment. This connects the one-step relationship between reconstruction error and domain evidence to the temporal development of domain identification during sampling.

G Reverse-Trajectory Diagnostics

The one-step analysis in Appendix F showed stronger intended-domain evidence under CPDM despite higher absolute reconstruction error. We now examine how this difference develops along complete generation trajectories. In particular, we ask when evidence for the intended domain emerges, whether it persists along individual trajectories, and how these temporal differences relate to the geometry of the generated features.

To address these questions, we track the full reverse trajectories of drift-off ($A = 0$) and CPDM ($A = 2$). We first examine condition-specific trajectories and endpoint distributions, then measure their relationship to real-domain centroids, and finally quantify the temporal persistence of domain evidence. Together, these analyses provide complementary observations for interpreting the relationship between reconstruction and condition-wise separation introduced in Appendix F.

Evaluation setup. We use the existing 50K checkpoints without additional training. For each model, we generate 500 samples per domain under each of three sampling seeds, yielding 1,500 flower and 1,500 leaf samples per model and 6,000 final images across both models. The initial base noise and all subsequent transition noise are matched across models and between flower/leaf conditions. Both models retain their original scalar-condition inputs. The three seeds represent different sampling noise realizations rather than independent training runs. We follow Algorithm 1 and observe all 1,000 reverse steps immediately before each update. Here, $t = 999, \ldots, 0$ denotes the zero-based implementation index, corresponding to diffusion timesteps $T, \ldots, 1$ in the main formulation, and $k = 999 - t$ counts the observed reverse steps. The final outputs are evaluated separately after the last update.

At each step, we evaluate the predicted clean image $\hat{x}_0(t)$. For semantic analyses, we use 2,048-dimensional Inception features and a frozen classifier trained on real data. For feature extraction, predictions are clipped to $[-1, 1]$ and converted to 8-bit RGB images, whereas pixel-space geometric measurements use unclipped states and predictions. Unless otherwise specified, time-dependent curves represent the median and interquartile range of 500 samples within each seed and domain.

We also measure the fraction of raw pixel-channel values in $\hat{x}_0(t)$ that fall outside $[-1, 1]$. The stepwise medians remain close to zero across the two models, three seeds, and two domains, and the largest median among model–seed–domain–step groups is approximately 0.60% ($A = 2$, flower). This indicates that clipping alters only a small fraction of pixel-channel values in a typical prediction; it does not imply that the effect of clipping is equally small for every individual prediction.

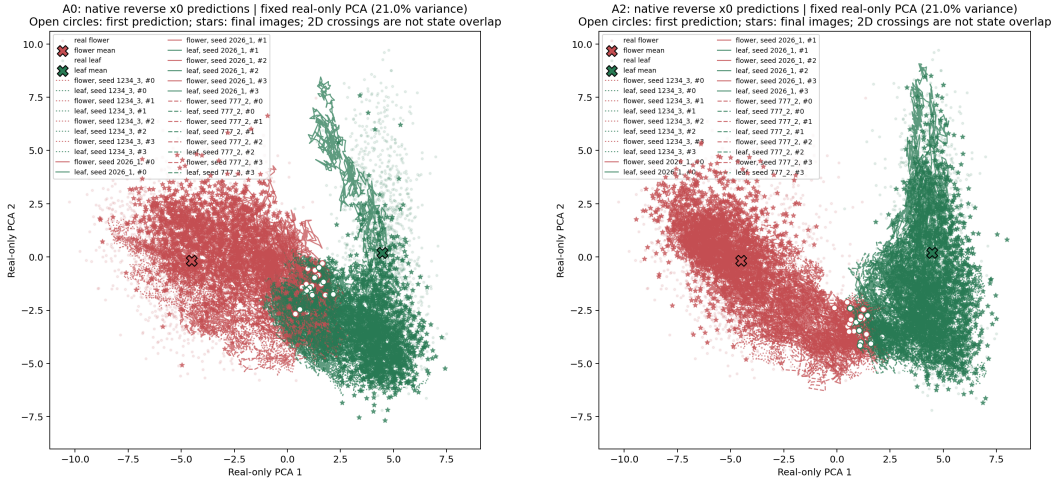
G.1 Trajectory Geometry and Endpoint Distributions

Semantic trajectories. We first examine how condition-specific generation paths evolve within the feature space of the real domains. We project the Inception features extracted from $\hat{x}_0(t)$ at each step onto a shared PCA plane fitted to balanced real flower and leaf features.

In Figure 9, the flower and leaf trajectories under $A = 0$ show substantial bending and overlap in the central region of the projected space. By contrast, the trajectories under $A = 2$ form more distinct condition-dependent branches and progress toward their corresponding real-domain regions. This difference is particularly pronounced for leaf. The displayed $A = 0$ leaf trajectories remain concentrated in a relatively lower region, whereas $A = 2$ extends into a wider vertical region occupied by real leaf features.

To examine whether these trajectory differences are also reflected in the final generated distributions, Figure 10 compares the features of the final outputs on the same PCA axes. The final leaf features under $A = 0$ are concentrated mostly below the real leaf centroid, whereas those under $A = 2$ extend farther into the upper region occupied by real leaf features. Differences in the location and spread of the final flower features are also observed. Thus, Figures 9 and 10 show that differences in condition-specific generation paths occur alongside changes in the spatial arrangement of the final feature distributions.

The first two principal components explain approximately 21.0% of the variance in the real features, so crossings or separation in the projected space alone cannot establish the relationships in the original high-dimensional space. The following analyses therefore separate the effect of the prescribed drift in image space and examine the relationship between generation paths and real domains in the original 2,048-dimensional feature space.



(a) Drift-off ($A = 0$) (b) CPDM ($A = 2$)

Figure 9: Semantic trajectories of predicted clean images in a fixed PCA projection fitted to real data. Red and green represent flower and leaf, respectively. Four trajectories per seed and domain are displayed for each model. Open circles indicate the initial clean-image predictions, and stars represent all 3,000 final samples for each model. Background points and large crosses indicate real-data features and their domain centroids. The first two principal components explain approximately 21.0% of real-feature variance; intersections of trajectories in the projected space do not establish overlap in the original feature space.

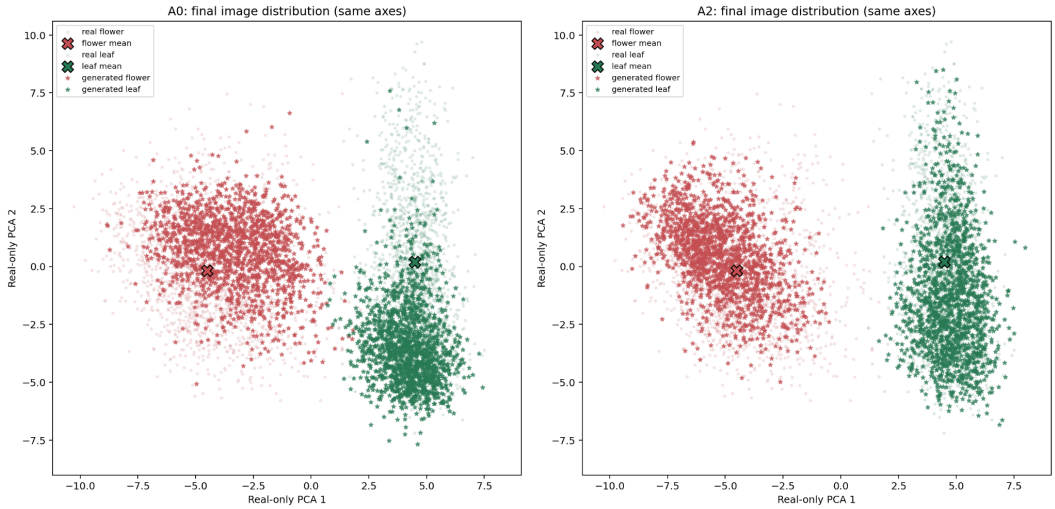


Figure 10: Final generated feature distributions under $A = 0$ and $A = 2$, compared using the same real-data PCA projection and axes. Each panel contains 1,500 flower and 1,500 leaf samples. Real-data feature distributions and domain centroids provide a common spatial reference for comparing the two models' final distributions.

Native and drift-corrected geometry. We next examine whether the differences in condition-specific trajectories merely reflect positional differences induced by the prescribed drift. Because CPDM directly introduces a condition-dependent displacement into the forward mean, separation observed in the native state x_t includes the deterministic drift.

To distinguish these effects, we use the drift-corrected coordinates y_t defined in Section 3.5 and the recovered noise prediction $\hat{\epsilon}_\theta$ defined in Appendix A.6. With these existing definitions, the clean-image prediction can be expressed identically in the two coordinate systems:

$$\hat{x}_0(t) = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\eta}_\theta(x_t, t, c)}{\sqrt{\bar{\alpha}_t}} = \frac{y_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\epsilon}_\theta(x_t, t, c)}{\sqrt{\bar{\alpha}_t}}. \tag{82}$$

This is an exact identity that holds regardless of the network's prediction. Thus, y_t represents the same sampling trajectory after removing the prescribed deterministic displacement, without changing the model's inference procedure. Actual inference continues to use the native state x_t and the conditioning input.

Figure 11 projects x_t and y_t onto the image-space drift direction and a fixed orthogonal direction. In the native x_t coordinates, $A = 2$ starts from condition-separated initial states, whereas $A = 0$ starts from shared initial states. In the drift-corrected y_t coordinates, the initial states of noise-matched flower and leaf conditions coincide for both models. This is the expected result in the DDPM-equivalent coordinate representation.

Nevertheless, the two models' trajectories behave differently even after this correction. In Figure 11, the corrected coordinates under $A = 2$ develop condition-dependent displacements along the drift axis as sampling proceeds, whereas $A = 0$ remains relatively close to zero along that axis.

To examine whether this difference extends beyond representative trajectories, Figure 12 measures pixel RMS distances between flower–leaf conditions in x_t, y_t, and $\hat{x}_0(t)$ for all 1,500 noise-matched pairs per model. For both models, the distance between paired conditions in y_t starts at zero. At $k = 399$, however, the median corrected-state distance is 0.318 for $A = 2$, compared with 0.105 for $A = 0$; at $k = 999$, the corresponding values are 0.605 and 0.268. At both steps, $A = 2$ shows **greater corrected-state separation than** $A = 0$ **for all 1,500 noise-matched pairs**. Greater condition-wise separation under $A = 2$ is also maintained in the pixel distances between predicted clean images.

Thus, CPDM's condition-specific trajectory differences cannot be explained solely by the explicit positional difference imposed on the initial terminal geometry. Greater condition-wise separation persists even in coordinates from which the prescribed deterministic drift has been removed, showing

that the two models exhibit different condition-dependent evolution during reverse sampling. These observations alone, however, cannot disentangle the respective contributions of adaptation during training and reverse dynamics.

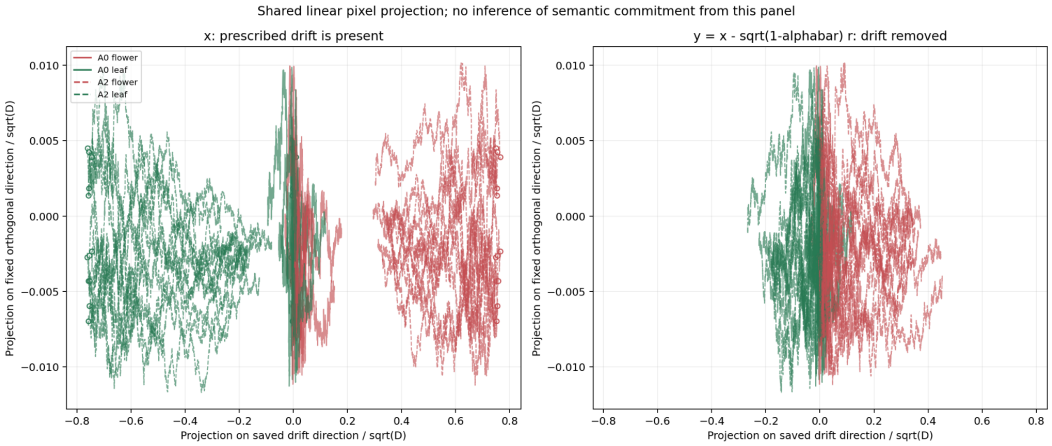


Figure 11: Representative trajectories of native states x_t and drift-corrected states y_t in a fixed image-space projection. Red and green represent flower and leaf, and solid and dashed lines represent $A = 0$ and $A = 2$, respectively. Open circles mark the initial states. The projection axes correspond to the prescribed drift direction and a fixed orthogonal direction, with both coordinates divided by the square root of the number of pixel dimensions.

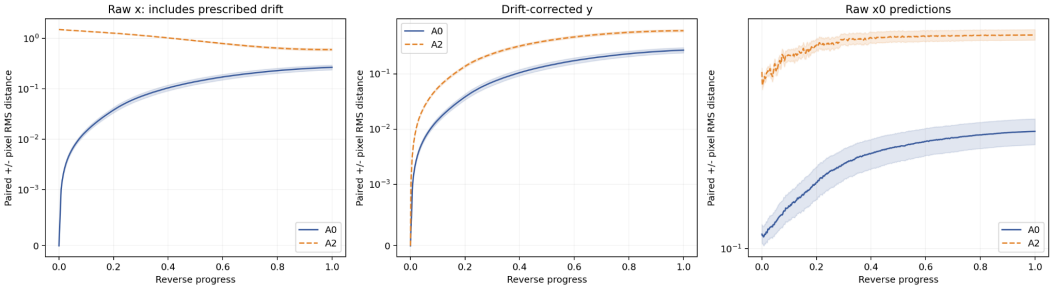


Figure 12: Pixel RMS distances between noise-matched flower–leaf pairs in native states, drift-corrected states, and unclipped clean-image predictions. Curves and shaded regions show the median and interquartile range across 1,500 matched pairs per model. The vertical axis uses a symmetric logarithmic scale. Distances between predictions do not represent reconstruction errors relative to ground-truth images.

G.2 Real-Domain Proximity and Semantic Evidence

G.2.1 Distances to Real-Domain Centroids

Section G.1 showed that CPDM maintains greater condition-wise trajectory separation even after removing the prescribed drift, while also exhibiting differences in the spatial arrangement of the final feature distributions. However, **greater separation between condition-specific trajectories does not, by itself, establish that those trajectories move closer to their intended real domains**. Even if the two conditions move farther apart, the generated features could move into regions away from the real data.

We therefore examine how these generation paths relate to the real domains. Figure 10 compared real and final generated feature distributions in a shared PCA space. Here, we extend this analysis to the original 2,048-dimensional feature space by measuring distances between features predicted during generation and real-domain centroids.

Let f denote the Inception feature extractor and μ_c the mean feature of real images from domain c. For the intended domain c and the opposite domain $\bar{c}$, we define

$$d_{\text{target}}(t) = \|f(\hat{x}_0(t)) - \mu_c\|_2,$$
$$d_{\text{other}}(t) = \|f(\hat{x}_0(t)) - \mu_{\bar{c}}\|_2. \tag{83}$$

We first measure d_{target}, the distance to the intended real-domain centroid. In Figure 13, $A = 2$ produces features closer to the intended real-domain centroid for both domains throughout substantial portions of the middle and late reverse process. The overall pattern recurs across all three sampling seeds.

At the final output, the median target distance decreases from 11.863 to 10.222 for flower and from 12.591 to 10.862 for leaf (Table 7). These correspond to reductions of 13.8% and 13.7%, respectively. In noise-matched comparisons of final outputs, $A = 2$ yields a smaller distance to the intended-domain centroid for 1,168 of 1,500 pairs (77.9%) in each domain.

Thus, the changes in generated-feature locations observed in Figure 10 are also reflected in the original feature space: **improved proximity to the intended real-domain centroid is observed for both flower and leaf**. Greater condition-wise trajectory separation occurs alongside smaller target-centroid distances. Centroid proximity characterizes one aspect of feature geometry and does not by itself establish distributional fidelity.

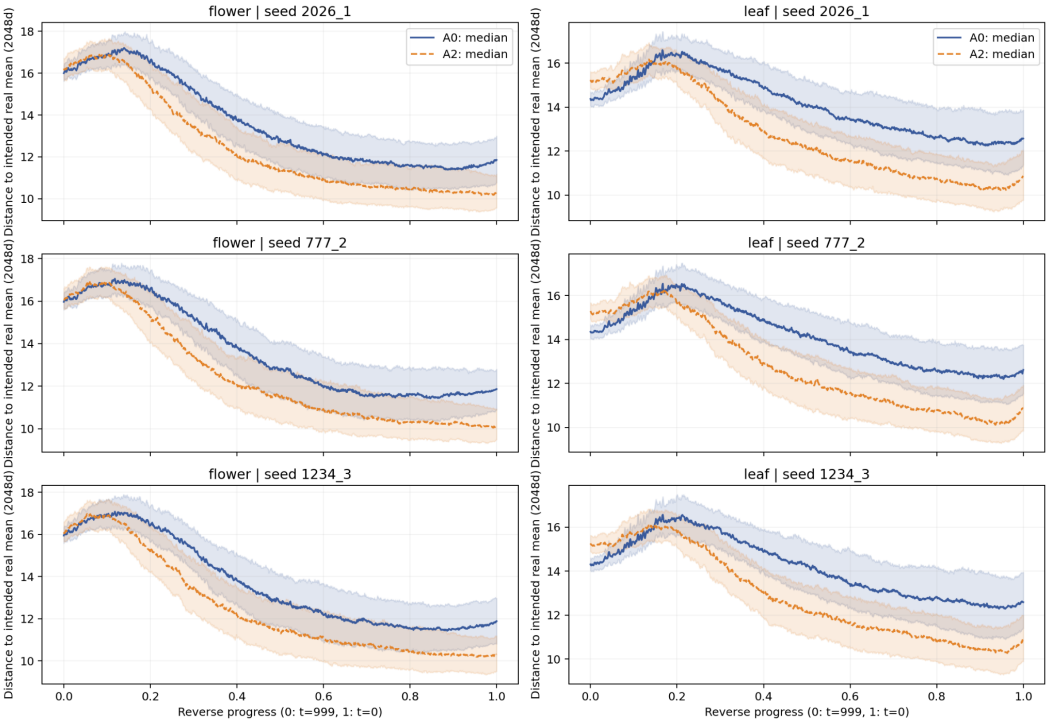


Figure 13: Distance between the predicted clean image at each step and its intended real-domain centroid. Columns correspond to flower and leaf, and rows correspond to the three sampling seeds. Curves show the median across 500 samples per model in each panel, and shaded regions indicate the 25th–75th percentile range. Distances are computed in the original 2,048-dimensional Inception feature space.

We next examine how this approach toward the intended domain is reflected in the relationship to the opposite domain. For this purpose, we measure d_{other}, the distance between generated features and the opposite real-domain centroid.

A larger distance to the opposite-domain centroid does not necessarily indicate more faithful domain separation. This distance can also be large when generated features lie far from both domains. We therefore calculate the distance to the opposite-domain centroid for real images from the intended

domain as well, using it as a reference for interpreting generated trajectories. The median distance from 8,189 real flower features to the leaf centroid is 14.557, while the median distance from 8,000 real leaf features to the flower centroid is 13.959.

In Figure 14, the flower trajectories of both models approach the real-data reference, with relatively small differences in the late stage. For leaf, $A = 2$ moves below the real-data reference during the middle-to-late stage and approaches it again near the final endpoint, whereas $A = 0$ finishes at a greater distance than the real-data reference. At the final endpoint, $A = 2$ is closer to the real opposite-domain distance reference for leaf, but $A = 0$ is slightly closer for flower (Table 7).

These differences concern an individual geometric property—the distance to the opposite-domain centroid—and do not imply an ordering of distributional fidelity or FID. The target-centroid distance describes proximity to the intended real domain, while the opposite-domain distance and its real-data reference describe how that proximity is expressed in relation to the other domain.

Taken together, the trajectory differences observed in Section G.1 occur alongside improved target-centroid proximity in both domains. The real-data references further distinguish these geometric changes from simply increasing the distance between the two generation paths.

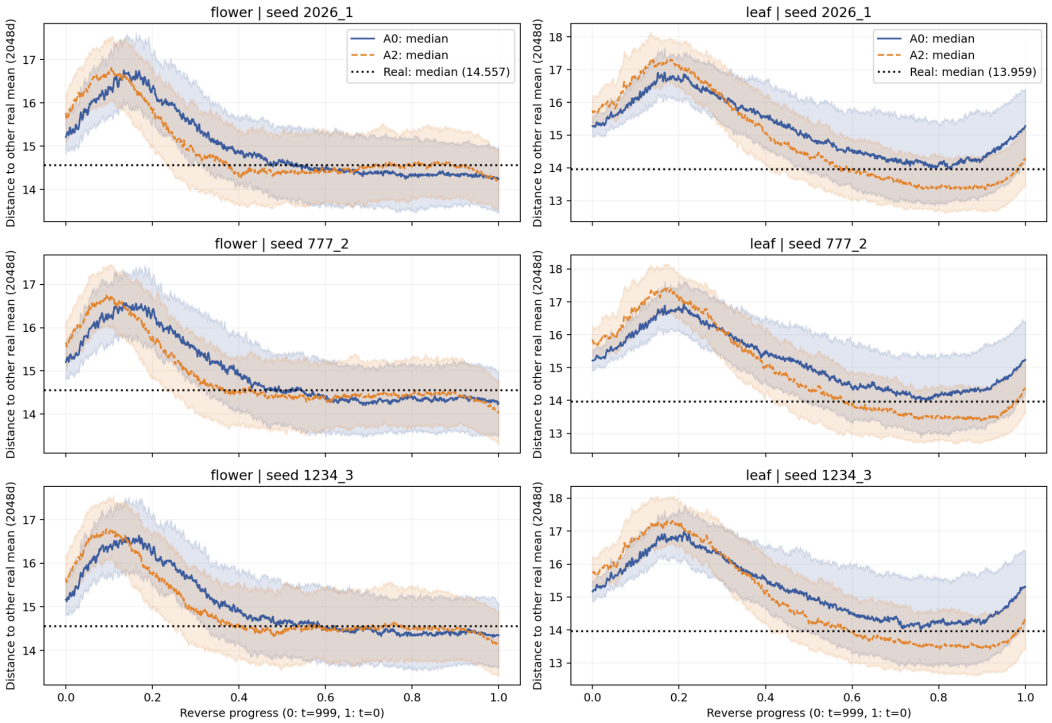


Figure 14: Distance to the opposite real-domain centroid, together with a clean real-data reference. The black dashed line indicates the median distance from real samples of the intended domain to the opposite-domain centroid, not the distance between the two domain centroids. Model curves and shaded regions use the same median and interquartile-range representation as Figure 13.

G.2.2 Classifier-Margin Evolution

The preceding analysis showed that CPDM maintains greater condition-wise trajectory separation while producing features closer to their intended real-domain centroids. We next examine how evidence for the requested domain develops along these trajectories, extending the one-step classifier analysis in Appendix F to actual reverse sampling.

We measure the intended-domain classifier margin of the predicted clean image at each step. Let ℓ_F denote the signed flower logit of the frozen classifier (positive for flower), and define

$$m(t) = s_z \, \ell_F(f(\hat{x}_0(t))). \tag{84}$$

Table 7: Median feature distances for final generated outputs pooled across three sampling seeds (1,500 samples per model and domain). The real-data reference is the median distance from clean real images of the intended domain to the opposite-domain centroid. Proximity to this reference provides context for geometric interpretation and is not an independent image-quality metric.

Domain	A	Target distance	Opposite-domain distance	Real opposite-domain distance reference
Flower	0	11.863	14.278	14.557
Flower	2	10.222	14.119	14.557
Leaf	0	12.591	15.269	13.959
Leaf	2	10.862	14.327	13.959

Here, $s_z = +1$ corresponds to flower and $s_z = -1$ to leaf, so a positive margin represents evidence supporting the requested domain.

In Figure 15, $A = 2$ shows stronger intended-domain evidence over a substantial portion of reverse sampling. For flower, the median margin under $A = 2$ becomes positive earlier. For leaf, the difference between the two models is more pronounced.

The leaf predictions under $A = 0$ initially have a high positive margin, but their domain evidence weakens as the reverse process proceeds before recovering. By contrast, although $A = 2$ initially has a comparatively lower margin, its intended-domain evidence strengthens rapidly after the early stage.

Among the 1,500 leaf samples per model, the fraction classified as the intended domain decreases from 98.3% at $k = 0$ to 48.7% at $k = 199$ for $A = 0$. For $A = 2$, the fraction increases from 68.3% to 98.0% over the same interval. The corresponding median margins decrease from 4.868 to -0.145 for $A = 0$, but increase from 0.684 to 5.665 for $A = 2$. This overall pattern recurs across the three sampling seeds.

Thus, CPDM does not show stronger domain evidence on every metric from the very first clean-image prediction. The important difference lies in **how intended-domain evidence evolves and stabilizes as sampling proceeds**.

At the final output, the median margin increases from 11.066 to 13.671 for flower and from 13.486 to 15.196 for leaf, comparing $A = 0$ with $A = 2$. Together with the target-centroid proximity results in Section G.2.1, these observations show that CPDM's trajectories lead to final features with stronger intended-domain classifier evidence and smaller target-centroid distances.

The margin measured here is the response of a frozen classifier applied to predicted images; it does not directly measure when semantic representations emerge within the neural network. Moreover, a positive margin at a particular step may become negative again later. A higher group-level margin therefore cannot establish that domain classification remains stable along individual trajectories. The next section examines this persistence at the sample level.

G.3 Persistent Domain Commitment

Section G.2.2 showed that CPDM tends to strengthen intended-domain evidence more rapidly during reverse sampling. The leaf results under $A = 0$ also demonstrate why an initially high margin is insufficient to establish persistent domain identification. We therefore distinguish the first occurrence of positive evidence from the point at which that evidence remains above a specified threshold throughout the rest of generation.

For sample i and threshold θ, we define the persistent commitment step as

$$k_i^*(\theta) = \min \left\{ k \in \{0, \dots, 999\} : \begin{array}{l} m_i(j) > \theta \quad \text{for every } j \in \{k, \dots, 999\}, \\ \text{and } m_i^{\text{final}} > \theta \end{array} \right\}. \tag{85}$$

Here, $m_i(j)$ denotes the margin at reverse step $j = 999 - t$, and m_i^{final} is measured on the final output after the last update. Samples for which no step satisfies these conditions are recorded as uncommitted. We evaluate $\theta = 0$ and $\theta = 1$ across all reverse steps. A smaller k_i^* indicates earlier persistent intended-domain evidence. If a sample exceeds the threshold at an early step but later falls to or below it, that early crossing is not counted as commitment.

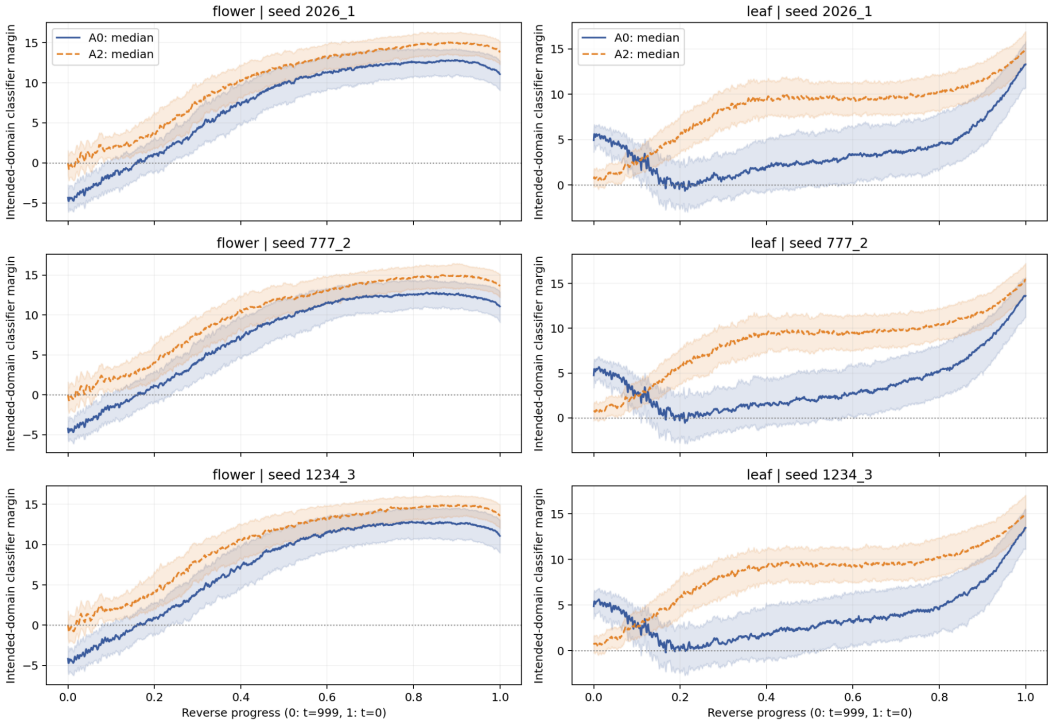


Figure 15: Intended-domain classifier margins throughout reverse sampling. Positive values support the requested domain. Curves and shaded regions show the median and interquartile range across 500 samples per model, seed, and domain. The horizontal reference line indicates a margin of zero.

In Figure 16 and Table 8, both models eventually achieve commitment for nearly all samples. At $\theta = 0$, 2,998 of 3,000 samples under $A = 0$ and all 3,000 samples under $A = 2$ reach commitment. At $\theta = 1$, the corresponding counts are 2,995 and 3,000.

The main difference is **when intended-domain evidence begins to persist**. At $\theta = 0$, the median commitment step for flower decreases from 245.5 under $A = 0$ to 131 under $A = 2$, while for leaf it decreases from 596.5 to 85. Among matched pairs in which both models reach commitment, $A = 2$ commits earlier in 85.7% of flower pairs and 92.6% of leaf pairs. The median paired commitment-step differences are -111 and -485.5 steps, respectively.

These results persist under the stricter threshold $\theta = 1$. At that threshold, $A = 2$ reaches commitment earlier for 84.2% of flower pairs and 93.8% of leaf pairs.

Table 8: Persistent commitment analysis for 1,500 samples per model and domain. Each model's median commitment step is calculated using samples committed under that model. Paired differences and earlier-commitment rates include only pairs in which both models reach commitment. The paired step difference is defined as $k^*_{A=2} - k^*_{A=0}$; its median need not equal the difference between the models' individual medians.

Domain	θ	Median step $A = 0$	Median step $A = 2$	Median paired difference	Earlier under $A = 2$	Both committed	Uncommitted $A = 0$ / $A = 2$
Flower	0	245.5	131	-111	85.7%	1,498	2 / 0
Leaf	0	596.5	85	-485.5	92.6%	1,500	0 / 0
Flower	1	276	168	-105	84.2%	1,496	4 / 0
Leaf	1	684	128	-518	93.8%	1,499	1 / 0

The difference is also evident during early sampling. At $\theta = 0$, by $k = 199$, 73.8% of flower and 87.6% of leaf samples under $A = 2$ have reached commitment, compared with 29.6% and 10.5%, respectively, under $A = 0$. These fractions use all 1,500 samples per domain as the denominator, including uncommitted samples. Thus, substantially more CPDM trajectories establish persistent

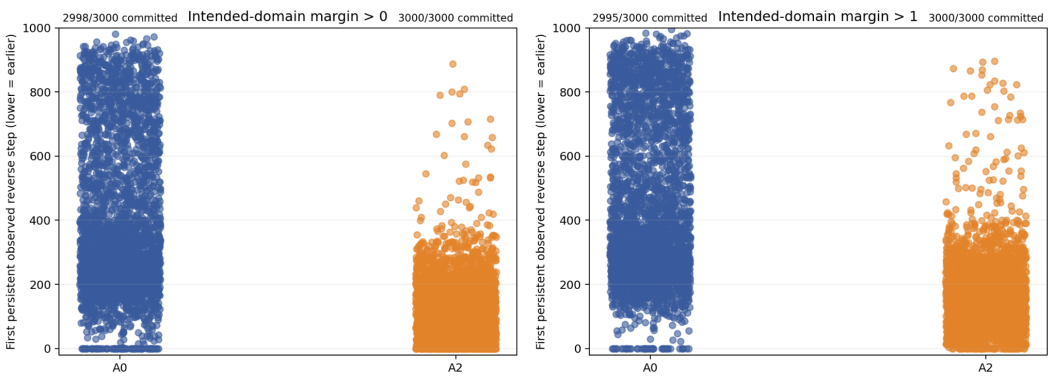


Figure 16: Persistent domain commitment steps measured at thresholds $\theta = 0$ and $\theta = 1$. Each point represents one committed sample from the pooled set of both domains and three sampling seeds. Horizontal jitter is used only for visibility. Smaller values indicate earlier commitment. Uncommitted samples are excluded from the scatter plot and reported separately in Table 8.

positive evidence for the intended domain within the first 200 observed reverse steps ($t = 999$ through 800).

Conversely, late commitment becomes less frequent. At $\theta = 0$, the fraction reaching commitment after $k = 400$ falls from 8.8% to 0.7% for flower and from 69.3% to 1.7% for leaf. These fractions also use all 1,500 samples per model and domain as the denominator; the two uncommitted $A = 0$ flower samples are reported separately in Table 8.

To complement the commitment-step analysis, we also count margin sign changes over each full reverse trajectory. The mean number of sign changes decreases from 18.1 to 13.4 for flower and from 32.8 to 9.8 for leaf.

These results connect to the margin evolution in Section G.2.2. For leaf under $A = 0$, initially strong domain evidence weakens during the intermediate stage, whereas CPDM maintains intended-domain evidence more persistently from a comparatively early point. This difference appears in both group-level margin curves and sample-level commitment statistics.

Thus, CPDM exhibits a clear temporal difference in external domain classification: **intended-domain evidence becomes persistent earlier as reverse sampling continues, with fewer margin sign changes over the full reverse trajectory**. This result concerns the temporal stability of the external classifier's decisions on predicted images.

Joint interpretation with Section 4.2.4 and Appendix F. The analyses now return to the question introduced at the beginning of Appendix F: how can forward placement improve generation performance when it does not reduce absolute one-step reconstruction error?

Section 4.2.4 and Table 1 established the generation-performance difference: directly supplying the scalar condition, full drift map, or both to the reverse denoiser did not reproduce the FID/KID gains obtained through forward placement.

Appendix F then showed that stronger intended-domain evidence and lower target-normalized prediction error can coexist with higher absolute one-step reconstruction error. Appendix G extended this observation to actual generation trajectories. Condition-wise separation remained greater after deterministic drift correction, final features lay closer to their intended real-domain centroids in both domains, and sample-level classifier evidence became persistent substantially earlier.

Together, these results show that CPDM changes both the geometric development of condition-specific trajectories and the timing of persistent domain evidence. Stable external-classifier evidence is established earlier while sampling continues, alongside the endpoint-quality gains reported in Section 4.2.4.

These observations provide indirect support for the interpretation that condition-specific forward geometry **can reduce the reverse denoiser's dual-task burden of reconstruction and condition-wise separation**. This interpretation concerns how establishing condition-wise separation and reconstructing an image unfold together during sampling; it does not require the standalone errors of both tasks to decrease. The higher absolute one-step reconstruction error in Appendix F and the earlier stabilization of intended-domain evidence in Appendix G are therefore compatible. Together with the improved generation performance, they form a coherent pattern consistent with a geometric mechanism by which condition-specific forward placement can reduce the joint demands of reconstruction and separation.

Scope of the interpretation. These diagnostics do not directly decompose the training-time optimization demands associated with reconstruction and separation or quantify their interaction through gradient analysis. The observed geometric and temporal differences therefore support the proposed interpretation without directly establishing the causal mechanism underlying a reduction in the joint burden.

H Complementarity of Forward Geometry and CLIP Image Conditioning

In the main comparison, CLIP image embedding achieved better endpoint performance than scalar CPDM, confirming the effectiveness of rich reverse conditioning. We therefore treat reverse-conditioning capacity and forward-geometry placement as distinct design axes, and evaluate whether adding forward geometry to the same CLIP conditioning can provide further improvement.

The CLIP-only model ($A = 0$) and the model with linear forward geometry ($A = 2$) used the same CLIP 512D image embedding for reverse conditioning. No separate s_z embedding was added to the reverse input of the hybrid model; s_z was used only in the forward state, structured prediction target, and s_z-dependent regularization. Both models were evaluated at the 50K checkpoint under training seed 777, using 5,000 generated images per domain.

Table 9: CLIP image conditioning with and without forward geometry, evaluated at the 50K checkpoint under training seed 777 using 5,000 generated images per domain. Avg FID and Avg KID are averaged over the two endpoint domains. FID Gap is the absolute Flower–Leaf FID difference. Lower is better.

Model	Flower FID↓	Leaf FID↓	Avg FID↓	Avg KID↓	FID Gap↓
CLIP ($A = 0$)	39.66	**19.28**	29.47	0.01809	20.38
CLIP + forward ($A = 2$)	**31.88**	22.56	**27.22**	**0.01733**	**9.32**

Adding forward geometry reduced average FID from 29.47 to 27.22 and average KID from 0.01809 to 0.01733. Flower FID improved from 39.66 to 31.88, while the FID Gap between the two domains decreased from 20.38 to 9.32. However, Leaf FID increased from 19.28 to 22.56, so the improvement was not uniform across domains.

These results show that forward geometry can complement strong CLIP conditioning to improve average endpoint performance without any additional reverse-conditioning input. They support the possibility that condition-wise organization through forward geometry and semantic information from rich reverse conditioning play complementary roles. However, these are endpoint results from a single training seed. Whether ordered traversal at unseen interior s_z coordinates is preserved under high-capacity conditioning has not been evaluated and remains future work.

I Base CPDM and Continuous CPDM Scalar Traversal

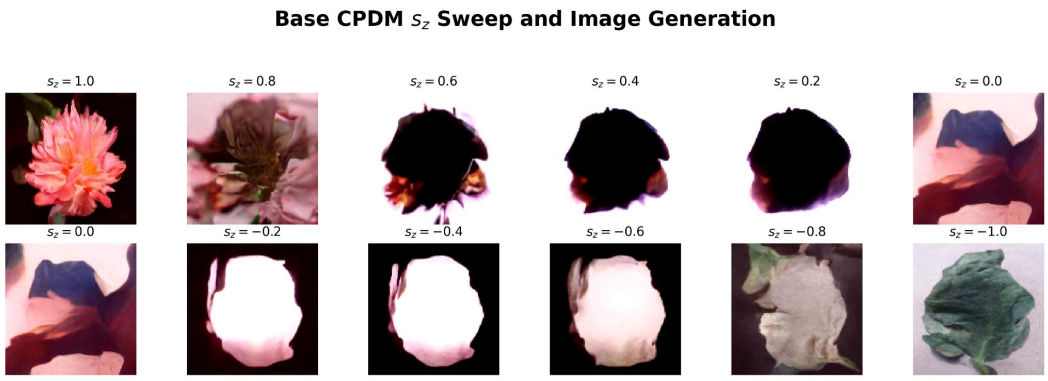
Table 10 reports the endpoint-domain performance of Base CPDM and Continuous CPDM under Training seed 777 at the selected checkpoint (50K steps). This comparison examines how the pairwise consistency objective affects endpoint fidelity and scalar-coordinate traversal behavior. The endpoint metrics show that the best Continuous CPDM setting introduces only a small endpoint-fidelity trade-off relative to Base CPDM. In particular, $\lambda = 0.1$ slightly increases Avg FID from 43.71 to 45.29,

while maintaining a comparable Avg KID. This suggests that the consistency objective does not primarily improve endpoint generation quality, but instead affects how the model responds to scalar coordinates between the endpoint regions.

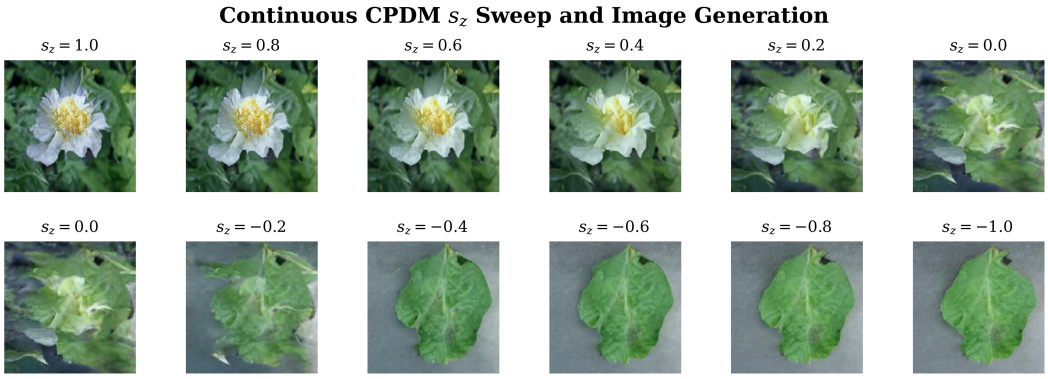
Table 10: Endpoint performance of Base CPDM and Continuous CPDM on the Leaf/Flower setting under Training seed 777. FID/KID are evaluated using 5,000 generated samples per endpoint domain. Lower is better.

Model	Avg FID ↓	Avg KID ↓	FID Gap ↓
Base CPDM	43.71	0.0345	11.60
Continuous CPDM ($\lambda = 0.1$)	45.29	0.0333	17.71
Continuous CPDM ($\lambda = 0.2$)	47.66	0.0367	32.69
Continuous CPDM ($\lambda = 0.3$)	62.08	0.0514	52.47

Figure 17 compares the s_z traversal from $s_z = +1.0$ to $s_z = -1.0$ at intervals of 0.2 under a shared fixed noise seed $y_T \sim \mathcal{N}(0, I)$. Base CPDM remains relatively stable near the endpoint side, including around $|s_z| = 0.8$, but it does not reliably reflect the unseen interior scalar coordinates after inference moves away from the training boundary bands. In contrast, Continuous CPDM produces non-collapsed outputs across the sweep and more consistently reflects changes in the s_z coordinate, indicating that the pairwise consistency objective stabilizes the scalar control axis with only a small endpoint-fidelity trade-off.



(a) Base CPDM



(b) Continuous CPDM

Figure 17: s_z traversal under a fixed noise seed (y_T shared) for Base CPDM (top) and Continuous CPDM (bottom). Panels are ordered from left to right and top to bottom, corresponding to $s_z \in \{+1.0, +0.8, \ldots, -0.8, -1.0\}$. Base CPDM remains plausible near endpoint-side coordinates but becomes less reliable in reflecting unseen interior scalar coordinates, whereas Continuous CPDM maintains non-collapsed outputs that better reflect changes along the s_z axis.

This comparison also motivates a mechanistic interpretation of forward conditioning. Unlike an s_z sweep with $A = 0$, where no condition-dependent displacement is prescribed by the forward process, CPDM maps the scalar coordinate into the noisy state x_t itself. Thus, varying s_z changes both the explicit conditioning input and the geometry of the state to be denoised. With denoiser capacity held fixed, this construction may allow forward geometry to support part of the condition-wise separation otherwise learned through reverse conditioning. The interior collapse of Base CPDM and its mitigation with $\mathcal{L}_{\text{diff}}$ are consistent with the importance of learning coherent responses along this coupled state–condition path. These observations provide indirect support for the hypothesis that forward geometry can alleviate the denoiser's joint reconstruction and separation burden; however, the traversal changes x_t and the explicit s_z input together and therefore does not isolate their individual contributions.

J Training-Step FID Dynamics Across Baselines

This appendix reports checkpoint-wise FID dynamics to support the checkpoint-selection protocol used in the main experiments. For each setting, all models are evaluated over the same predefined checkpoint range, and the best checkpoint reported in the main tables is selected from that common range. At each checkpoint, FID is evaluated using 1,000 generated images per domain.

J.1 Flower/Leaf setting: 30K–50K checkpoints

For the Flower/Leaf experiments, we evaluate checkpoints from 30K to 50K training steps at 5K-step intervals. Specifically, the evaluated checkpoints are 30K, 35K, 40K, 45K, and 50K. The same checkpoint range is used for CPDM and all DDPM-based baselines.

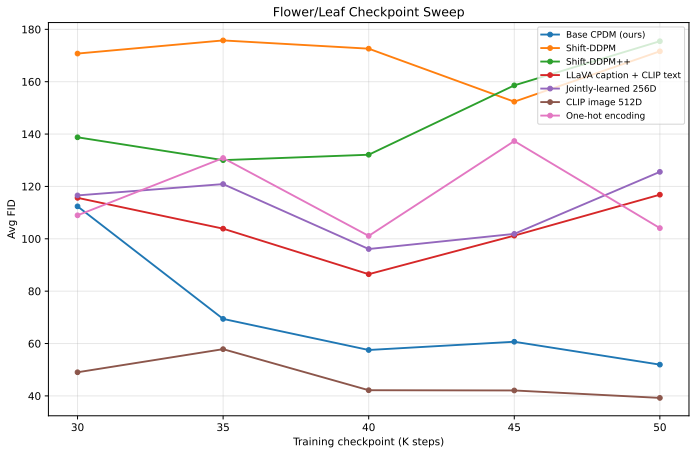


Figure 18: Training-step Avg FID dynamics on the Flower/Leaf setting. All models are evaluated over the same checkpoint range from 30K to 50K steps. Lower FID is better.

The Flower/Leaf dynamics show how each method changes across the common 30K–50K evaluation window. This analysis verifies that the reported best checkpoints are selected under a consistent checkpoint-selection protocol, rather than from model-specific evaluation ranges.

J.2 CelebA-HQ Gender setting: 50K–90K checkpoints

For the CelebA-HQ experiments, we evaluate checkpoints from 50K to 90K training steps at 10K-step intervals. Specifically, the evaluated checkpoints are 50K, 60K, 70K, 80K, and 90K. The same checkpoint range is used for CPDM and all DDPM-based baselines.

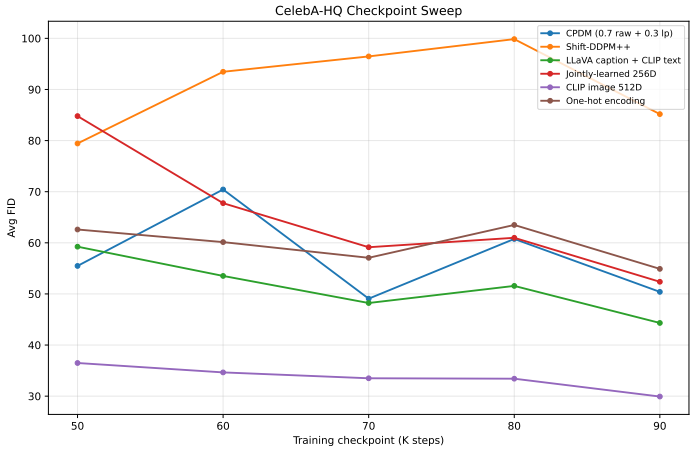


Figure 19: Training-step Avg FID dynamics on the CelebA-HQ Gender setting. All models are evaluated over the same checkpoint range from 50K to 90K steps. Lower FID is better.

These checkpoint-sweep curves are intended to document the common evaluation window and checkpoint-selection protocol, rather than to introduce additional model-specific tuning ranges.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Sections 1 and 3 distinguish the general multi-axis formulation from the scalar setting evaluated experimentally and define personalization as user-selectable continuous control rather than learned user preferences or subject-identity adaptation. Sections 4–5 and Appendices F–I distinguish the demonstrated endpoint and traversal results from mechanistic interpretations and extensions that remain unvalidated.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses drift-basis dependence, intra-domain limitations, training-seed variability in endpoint quality and domain balance, and the lack of validation of multi-axis control and larger-scale benchmarks. It also identifies the single-training-seed scope of the trajectory diagnostics and CLIP hybrid results. Appendices F–I further delimit the mechanistic interpretations and endpoint-fidelity trade-offs.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 3.5 and Appendix A derive KL/ELBO invariance under deterministic, condition-fixed translation of the DDPM forward chain, including the translated terminal prior, intermediate posterior correspondence, and unchanged clean-data endpoint. Appendix A.8 states the scope of this result, and Appendix B gives the corresponding reverse-sampling algorithm.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3–4 and Appendices B–D describe the algorithm, data preparation, training settings, checkpoint selection, seed protocol, and domain-wise FID/KID evaluation. Appendices F–G specify the diagnostic protocols. The public implementation linked in the abstract and its environment and workflow documentation provide the implementation details for training, sampling, and evaluation from the source datasets.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The public repository linked in the abstract provides CPDM and baseline code together with installation and training, sampling, and evaluation instructions; the source datasets are identified in Section 4.1 and Appendix C. Pretrained checkpoints are not distributed, so reproduction requires obtaining the datasets from their original sources and retraining the models using the documented protocol.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

Justification: Section 4 and Appendix C report the dataset construction, image resolution, optimizer, learning rate, batch size, gradient clipping, training and checkpoint ranges, sample counts, and checkpoint-selection rule. Section 4.2 and Appendices D–E and I describe the drift-design and consistency-weight comparisons, with implementation-level settings provided in the public code.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 2 reports the main Leaf/Flower results averaged over training seeds 777 and 2026. Its caption defines $\pm$ as half the seed range for FID/FID Gap and the within-run SD averaged over domains and seeds for KID. Appendix C describes the seed protocol; single-training-seed results and the evaluation and sampling seeds in Appendices F and G, respectively, are identified separately.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C.2 reports the hardware and execution times used for training and sampling. Training was performed on an NVIDIA L4 GPU, requiring approximately 20 hours per Leaf/Flower run and 35 hours per CelebA-HQ run, while sampling on an NVIDIA A100 GPU required approximately 70 minutes per 1,000 generated images.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses existing publicly available research datasets and released pretrained models, credits their sources, and does not recruit participants, conduct crowdsourcing, or collect private user-preference data. The paper reports methodological limitations and the scope of the experiments and does not present a deployed personalization system.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on forward-process design and controlled generation experiments and does not include a separate discussion of potential positive and negative societal impacts.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The release consists of research code for the controlled experimental settings described in the paper; it does not distribute pretrained image-generation checkpoints, newly collected datasets, or a deployed generation service. Accordingly, the release does not include the high-risk model or dataset assets targeted by this question.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The paper credits the datasets and pretrained models used in the experiments, but does not explicitly enumerate the applicable licenses and terms for all third-party assets. The article's CC BY 4.0 license applies to the paper and does not replace the licenses or terms of those external assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released asset is the implementation linked in the abstract, whose accompanying documentation describes the repository structure, installation, tested environment, and training, sampling, and evaluation workflows. No new dataset or pretrained checkpoint is distributed as part of this release.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments, participant recruitment, or studies with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work uses existing public research datasets and does not involve participant recruitment, human-subject experiments, or collection of private participant data, so participant-study approval is not applicable to the experiments described.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: CPDM's forward-drift construction, objectives, and sampling method do not use an LLM as a core methodological component. LLaVA-1.5 is used to construct caption-based comparison baselines, as described in Section 4.3 and Appendix E.4, rather than as part of CPDM itself.