Beyond the Manifold Hypothesis: Hybrid Spectral Parameterizations for Flow Matching

Anonymous authors
Paper under double-blind review

Abstract

Flow matching and diffusion can be trained to predict different quantities, most commonly the data x_1, the source noise x_0, or the velocity v. Although theoretically equivalent, these can lead to substantially different performances. We identify two main drivers for these differences: the source–data signal-to-noise ratio, and the information bottleneck induced by the neural architecture. We show that, beyond intrinsic data dimension, the factor affecting the optimal parametrization the most is a certain signal-to-noise ratio in each data covariance direction. From this analysis, we introduce new *spectral hybrid* parameterizations that adapt across time and data covariance directions; we show that these are optimal for Gaussian data. We also show that architecture-induced compression changes which parameterization is easier to learn, with v-prediction being more sensitive to discarded directions than x_1-prediction. Experiments across architectures and source scales show that our spectral parameterizations are robust across regimes, can substantially accelerate optimization, while incurring essentially no additional training cost compared with standard parameterizations.

The modern success of diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) heavily relies on a number of specific design choices whose impact are often not yet fully understood. Key design choices include specific neural architectures, such as U-Nets (Ronneberger et al., 2015) or ViTs (Dosovitskiy et al., 2021; Peebles and Xie, 2023), loss weighting schemes (Karras et al., 2022), optimizers (Kingma and Ba, 2015), and the choice of network parameterization, *i.e.*, which quantity the network is trained to predict.

Network parameterization has recently attracted renewed interest. The seminal diffusion works (Song and Ermon, 2019; Ho et al., 2020) originally parameterized the network to predict the noise added to the data.With the rise of flow matching (Lipman et al., 2023; Liu et al., 2023; Albergo and Vanden-Eijnden, 2023), predicting the velocity emerged as a strong contender. More recently, motivated by the *manifold hypothesis* (Chapelle et al., 2006, Chap. 1.2), which posits that natural images exhibit a low-dimensional structure, Li and He (2025) or Geng et al. (2026) have advocated for predicting the clean data.

Images exhibit structured regularities that have long been exploited through image-specific architectures (LeCun et al., 1998; Krizhevsky et al., 2012), as well as tailored training objectives. Modern approaches for incorporating image-specific information into generative models rely on two-stage procedures (Van Den Oord et al., 2017; Esser et al., 2021; Tian et al., 2024), in which a latent representation is first learned using image-specific perceptual losses (Zhang et al., 2018) that have been empirically shown to better preserve high frequencies. Incorporating the image statistics directly into one-stage training procedures has also been actively explored, for instance through anisotropic noising schemes and frequency-dependent noise schedules (Rissanen et al., 2023; Hoogeboom and Salimans, 2023; Falck et al., 2025).

Contributions. In this work, we propose a principled approach to network parameterization by elucidating how it connects to the spectral structure of the data. More precisely:

- First, we challenge the common intuition that the *manifold hypothesis* favors x_1-prediction: we show that rescaling the source distribution, while leaving the data manifold unchanged,

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

can reverse the performance ordering between x_0-, v-, and x_1-prediction (Figure 1). This highlights the source–data SNR as a key factor for parameterization performance.

- Second, based on these findings, we introduce two *spectral hybrid parameterizations* that adapt between x_0 and x_1 across time and data covariance directions. In a Gaussian-to-Gaussian model, we show that they are optimal.
- Third, we investigate the interactions between the parameterization and the architecture, highlighting the role of information compression. We show that velocity prediction is particularly sensitive to discarded information, while the spectral hybrids are very robust. Experiments on image datasets show that the proposed spectral hybrid parameterizations are competitive with the best fixed parameterization across source scales and architectures, with larger gains under strong architectural bottlenecks, and improve training convergence.

1 Reminders

Flow matching and diffusion transform a simple source distribution[1] P_0 into a target data distribution P_{data}, by mapping a sample $x_0 \sim P_0$ to the solution at time 1 of an Ordinary Differential Equation (ODE) governed by a velocity field V

$$\begin{cases} x(0) = x_0 \\ \partial_t x(t) = V(x(t), t) \quad \forall t \in [0,1] \end{cases} \tag{1}$$

In the most basic version of flow matching, for independent $X_0 \sim P_0$ and $X_1 \sim P_{\text{data}}$, V is learned as a neural network N_θ whose parameters θ solve

$$\min_\theta \mathbb{E}_{X_0, X_1, t\sim\text{unif}([0,1])}[\|N_\theta(X_t, t) - (X_1 - X_0)\|^2] \quad \text{where } X_t := (1-t)X_0 + tX_1. \tag{2}$$

If the neural network can approximate any measurable function and optimization is performed exactly, the optimal network is $\mathbb{E}[X_1 - X_0 \mid X_t = \cdot]$, for which the solutions of (1) at time 1 are guaranteed to be distributed according to P_{data} under mild regularity conditions.

Instead of learning the velocity (v-prediction), one can parametrize the network N_θ to predict, from X_t, either the data X_1 (x_1-prediction), the source X_0 (x_0-prediction). Still assuming infinite model capacity, the corresponding networks recover the conditional means $\mathbb{E}[X_1 \mid X_t = x]$, $\mathbb{E}[X_0 \mid X_t = x]$, and $\mathbb{E}[X_1 - X_0 \mid X_t = x]$. As detailed in App. A.1, these three quantities are algebraically related: knowledge of one implies knowledge of the other two. In practice however, each parametrization can lead to very different performances.

In order to study their respective performances, we compare parameterizations through their induced estimator of X_1 given x_t, which we refer to as the denoiser. Let N_θ^p denote the raw network output for a parametrization $p \in \{x_1, v, x_0\}$. The corresponding denoisers are

$$D_\theta^{x_1}(x,t) = N_\theta^{x_1}(x,t), \quad D_\theta^{v}(x,t) = x + (1-t)N_\theta^{v}(x,t), \quad D_\theta^{x_0}(x,t) = \frac{x-(1-t)N_\theta^{x_0}(x,t)}{t}. \tag{3}$$

In the infinite-capacity limit, all three recover the same optimal denoiser $D^\star(x,t) := \mathbb{E}[X_1 \mid X_t = x]$. To compare them fairly, we train all models against the same target, X_1:

$$\mathcal{L}^p(\theta) = \mathbb{E}_{t, X_0, X_1}\left[\tfrac{1}{(1-t)^2}\left\|D_\theta^p(X_t,t) - X_1\right\|^2\right], \qquad t \sim \mathcal{U}([0,1]). \tag{4}$$

We use the weighting $(1-t)^{-2}$ for all parameterizations in (4), as it was found to achieve the best performances across parameterizations (Li and He, 2025; Gagneux et al., 2026)[2]. This also ensures that our comparisons isolate parameterization rather than time weighting.

Remark 1. The parameterizations also differ in how prediction errors are propagated to the velocity used for sampling (1). Notably, under x_0-prediction, the corresponding velocity $V_\theta^{x_0} = (X_t - N_\theta^{x_0}(X_t,t))/t$ becomes singular when $t \to 0$[3]: sampling with this parameterization yields poor generative performances (Gagneux et al., 2026). This motivates the following evaluation protocol. We compare the parameterizations both through their denoising MSE as a function of t and through their generative performance (FID). Since sampling with x_0-pred is unstable, we only include x_1 and v-prediction for FID comparisons.

[1] In this paper, we adopt the flow matching convention: x_0 is the noise, x_1 is the clean data.

[2] In the vocabulary of Li and He (2025), this means we always use v-loss; see App. A.1.

[3] The amplification of prediction errors also depends on the interpolation schedule. Standard diffusion schedules behave differently from the path considered here; see App. A.2.

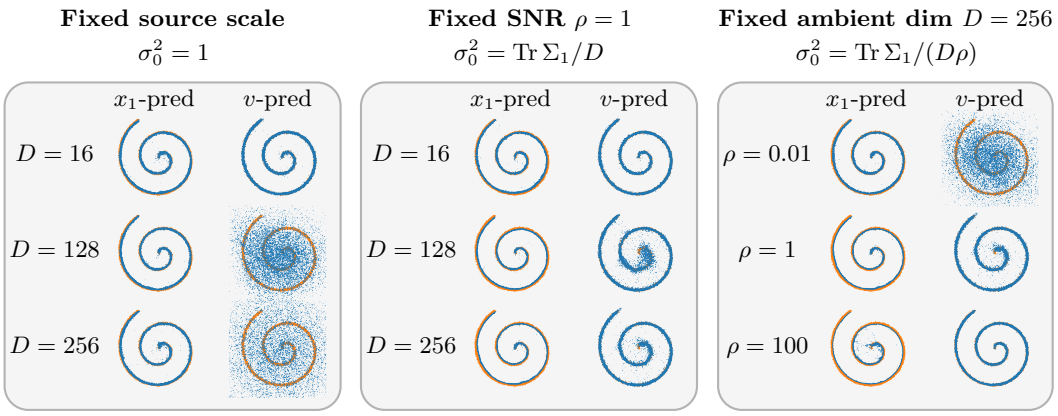


Figure 1: Disentangling ambient dimensionality and source–data SNR. True data are shown in orange and generated samples in blue. For all columns, $x_0 \sim \mathcal{N}(0, \sigma_0^2 I_D)$. *Left:* fixed $\sigma_0 = 1$; as the dimension D increases, the source–data SNR ρ decreases. *Center:* the source-data SNR $\rho = 1$ is kept fixed (by adapting σ_0) as D increases. *Right:* at a fixed $D = 256$, the source–data SNR varies as $\rho \in \{0.01, 1, 100\}$, by using $\sigma_0 \in \{0.6, 0.06, 0.006\}$. The conclusions of Li and He (2025) (left panel) no longer hold when the SNR is kept fixed (middle panel). **The right panel identifies the SNR as a critical factor**.

2 Beyond the manifold hypothesis: role of source–data SNR

A common explanation for the success of data prediction is the *manifold hypothesis* (Chapelle et al., 2006; De Bortoli, 2022): data is assumed to concentrate on a low-dimensional manifold of a much larger ambient space. This hypothesis has yielded the intuition that clean data X_1 may be simpler than predicting the source noise X_0 or the velocity $X_1 - X_0$, which both contain full-dimensional noise components (Li and He, 2025; Jin and Wang, 2026). While this argument captures an important property of natural data, we show in this section that **intrinsic dimensionality alone does not determine the best prediction target**.

Throughout this section, we assume $X_1 \in L^2$ and denote its covariance by $\Sigma_1 := \mathrm{Cov}(X_1)$, while the source is $X_0 \sim \mathcal{N}(0, \sigma_0^2 I_D)$. Following the usual definition of SNR as signal power divided by noise power, we define the source–data SNR as

$$\rho := \frac{\mathrm{Var}\, X_1}{\mathrm{Var}\, X_0} = \frac{\mathrm{Tr}(\Sigma_1)}{D\sigma_0^2}. \tag{5}$$

The SNR ρ can be changed by varying σ_0 without modifying the data scale and geometry.

Disentangling dimensionality and SNR. We revisit the experiment of Li and He (2025, Fig. 2), in which a two-dimensional spiral is embedded into an space of dimension D. They report that increasing D strongly favors x_1-prediction over v-prediction, and interpret this behavior through the increasing gap between intrinsic and ambient dimension. Because the spiral is embedded isometrically, $\mathrm{Tr}(\Sigma_1)$ remains independent of D, while Li and He use a standard Gaussian source, corresponding to $\sigma_0 = 1$ ($\mathrm{Var}\, X_0 = D$). Therefore, increasing the ambient dimension has the hidden side effect of decreasing the source–data SNR (5).

To disentangle these effects, in Figure 1 (left), we reproduce the setting of Li and He (2025) (fixed σ_0). In Figure 1 (middle) we vary D at fixed $\rho = 1$, using $\sigma_0^2 = \mathrm{Tr}(\Sigma_1)/D$, showing that *varying the dimension D at fixed SNR has little effect on the parameterization performance.* Finally, in Figure 1 (right) we fix $D = 256$ while varying σ_0, and observe that *changing the SNR with fixed dimension has a strong impact on the relative parameterization performance*: the source-data SNR is a critical factor in determining the optimal prediction.

3 From SNR-dependent prediction to spectral hybrid models

Having highlighted that the global SNR is a single statistic that critically affects parametrization performance, we now refine this perspective by providing a time- and direction-dependent

analysis. For the theoretical analysis in this section, we use the Gaussian model

$$X_1 \sim \mathcal{N}(0, \Sigma_1), \qquad X_0 \sim \mathcal{N}(0, \sigma_0^2 I), \tag{6}$$

with X_0 and X_1 independent. For clarity, we assume throughout the paper that the data distribution is centered and that the source is isotropic.[4].

3.1 A binary switch from directional SNR

In real anisotropic data, the energies vary substantially across the covariance spectrum. To model this, we consider the spectral decomposition of the data covariance: $\Sigma_1 = U \operatorname{diag}(\lambda_1, \dots, \lambda_D) U^\top$, where U has columns u_j. Along direction u_j, the source and data contributions to the observation $X_t = (1-t)X_0 + tX_1$ have variances $(1-t)^2\sigma_0^2$ and $t^2\lambda_j$ respectively. We therefore define the *directional* SNR at time t as

$$\rho_j(t) := \frac{t^2 \lambda_j}{(1-t)^2 \sigma_0^2}. \tag{7}$$

Based on Section 2, we expect the time at which the two contributions have equal energy (i.e. $\rho_j(t)$ equals 1) to play an important role on the optimal prediction choice. This time is

$$t_j^{\text{switch}} := \frac{\sigma_0}{\sigma_0 + \sqrt{\lambda_j}}. \tag{8}$$

Similarly to the global SNR case, we expect that $\rho_j(t) \leq 1$ should favor x_1-pred and $\rho_j(t) \geq 1$ should favor x_0-pred in the j-th direction.

Validation on a Gaussian model. We first test whether the switching time (8) predicts the time when the performances of separately trained x_1- and x_0-prediction models cross. We consider centered Gaussian data in dimension $D = 1024$, with covariance eigenvalues $\lambda_j \propto j^{-2}$, normalized such that $\sum_{j=1}^{D} \lambda_j = D$. For U, we use a DCT eigenbasis, yielding frequency-ordered directions with strongly anisotropic variances, as commonly observed in image data (van der Schaaf and van Hateren, 1996). The source is $X_0 \sim \mathcal{N}(0, \sigma_0^2 I_D)$.

We train two models with the same loss (4): one parameterized to predict x_0 and the other to predict x_1 (implementation details in App. C). To compare them fairly, we use for both the induced denoiser D_θ (3) and consider the *directional MSE*, i.e. the average of the residual error projected onto directions of the spectral basis, $\mathbb{E}\left|u_j^\top \left(D_\theta(X_t, t) - X_1\right)\right|^2$, which we track as a function of t. Typical curves are shown in Figure 2a; as expected from the above discussion, for each direction j, x_1-prediction outperforms x_0-prediction for small times (low ρ_j), while performance switches for larger times (high ρ_j).

Next, we define the *empirical transition time* in direction j as the smallest time for which the error of the x_0-model falls below that of the x_1-model. Figure 2b shows the alignment between this time and t_j^{switch} identified in Eq. (8): the relative performance of x_0- and x_1-prediction changes at a time that is almost perfectly predicted by the directional SNR.

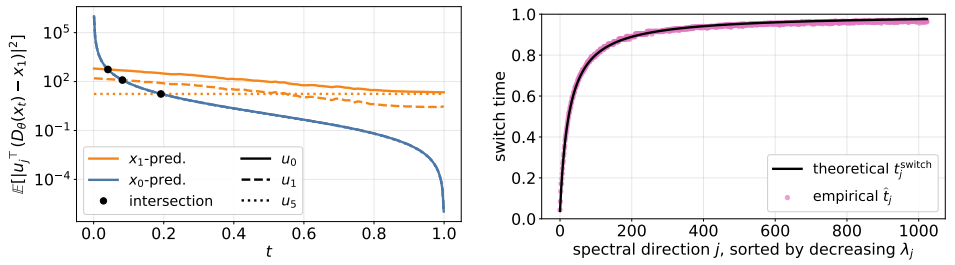
A first hybrid spectral model: the binary switch strategy. This motivates us to propose a first new training method, namely to switch from x_1- to x_0-prediction depending on time, but also on eigendirection, by outputting

$$\textbf{y_t^{bin}-prediction:} \qquad \begin{aligned} Y_t^{\text{bin}} &:= M_t X_1 + (I - M_t) X_0, \\ \text{where } M_t &:= U \operatorname{diag}\left(\mathbf{1}_{\{t < t_1^{\text{switch}}\}}, \dots, \mathbf{1}_{\{t < t_D^{\text{switch}}\}}\right) U^\top. \end{aligned} \tag{9}$$

Let $N_\theta^{\text{bin}}(x, t)$ be trained to predict Y_t^{bin}, i.e. X_1 on the subspace selected by M_t and X_0 on its complement. Converting the latter to X_1 using $X_1 = (X_t - (1-t)X_0)/t$ gives

$$D_\theta^{\text{bin}}(x, t) = M_t N_\theta^{\text{bin}}(x, t) + (I - M_t) \frac{x - (1-t) N_\theta^{\text{bin}}(x, t)}{t}, \tag{10}$$

[4]The extension to a non-zero data mean m is given in App. G, and amounts to applying the same analysis to the centered variables $X_1 - m$ and $X_t - tm$



(a) Directional MSE as a function of time (b) t_{switch}: practice confirms theory.

Figure 2: Validation of the theoretical switching time on the anisotropic Gaussian model, for $\rho = 1$ ($\sigma_0 = 1$). *Left:* directional MSEs of x_0- and x_1-prediction for three selected directions u_j (for x_0-pred, the curves are superposed). The crossing (black dot) occurs earlier in high-variance directions and later in low-variance directions. *Right:* For each spectral direction j, we compare the theoretical switching time t_j^{switch} with the empirical transition time (black dot in left figure). The theory agrees strongly with empirical observation.

that we train with the same denoising objective as in (4). Experiments evaluating this new prediction parameterization are reported in Section 5, where we show that this simple spectral hybrid is remarkably effective in practice.

3.2 What does the binary switch parametrization minimize?

The switching time identified above can also be derived as being optimal from a regression perspective in the Gaussian setting. Recalling that u_j is the j-th column of U, define the coordinates $X_{1,j} = u_j^\top X_1$, $X_{0,j} = u_j^\top X_0$, and $X_{t,j} = u_j^\top X_t$. From Eq. (3), for all parametrizations $p \in \{x_0, x_1, v\}$, we write the directional denoiser[5] as

$$\forall x \in \mathbb{R}, \quad D^p_{\theta,j}(x,t) = \underbrace{a_j^p(t)x}_{\text{analytical part}} + \underbrace{b_j^p(t)N^p_{\theta,j}(x,t)}_{\text{learnable correction}} . \tag{11}$$

For the three standard parameterizations, $(a_j^{x_1}, b_j^{x_1}) = (0,1)$, $(a_j^v, b_j^v) = (1, 1-t)$, and $(a_j^{x_0}, b_j^{x_0}) = (1/t, -(1-t)/t)$. We interpret the coefficient $a_j^p(t)$ as the analytical part imposed by the parameterization, while the network learns the remaining correction. Let $D_j^\star(x,t) := \mathbb{E}[X_{1,j} \mid X_{t,j} = x]$ denote the optimal squared-error denoiser. We define

$$\mathcal{R}^p_{t,j} := \mathbb{E}\left[\left(a_j^p(t)X_{t,j} - D_j^\star(X_{t,j},t)\right)^2\right]. \tag{12}$$

The residual $\mathcal{R}^p_{t,j}$ measures how far the analytical part is from the optimal denoiser, hence how much remains to be corrected by N^p_θ in direction j at time t.

In the Gaussian model, recall that $\lambda_j = \mathrm{Var}(\bar{X}_{1,j})$ and define $\sigma^2_{t,j} = (1-t)^2\sigma_0^2 + t^2\lambda_j = \mathrm{Var}(X_{t,j})$. In that setting, the optimal denoiser is linear and equals $D_j^\star(\bar{x},t) = t\lambda_j/\sigma^2_{t,j}$ (see App. C.4) and therefore $\mathcal{R}^p_{t,j} = \sigma^2_{t,j}\left(a_j^p(t) - t\lambda_j/\sigma^2_{t,j}\right)^2$. Explicitly, we get

$$\mathcal{R}^{x_0}_{t,j} = \frac{(1-t)^4\sigma_0^4}{t^2\sigma^2_{t,j}}, \qquad \mathcal{R}^{x_1}_{t,j} = \frac{t^2\lambda_j^2}{\sigma^2_{t,j}}, \qquad \mathcal{R}^{v}_{t,j} = \frac{(1-t)^2\big((1-t)\sigma_0^2 - t\lambda_j\big)^2}{\sigma^2_{t,j}}. \tag{13}$$

This allows comparing x_1- and x_0-prediction: $\mathcal{R}^{x_1}_{t,j} \le \mathcal{R}^{x_0}_{t,j}$ if and only if $t^2\lambda_j \le (1-t)^2\sigma_0^2$, namely $t \le t_j^{\text{switch}}$. Hence, the residual criterion recovers the same switching time and ordering as the directional SNR criterion in (8): x_1-prediction is *easier* for $t \le t_j^{\text{switch}}$ if one measures task hardness by the magnitude of what remains to be learned by the network (i.e., $\mathcal{R}_{t,j}$)[6]. The parametrization D^{bin} introduced in Eq. (10) precisely exploits this finding.

[5] Such writing is also used in Karras et al. (2022, Eqs. 7 and 8), but not direction-wise.

[6] The comparison including v-prediction is given in App. C.2.

3.3 From a binary switch to continuous spectral parametrization

So far, we have considered only a "binary" model, restricting the choice between x_1- and x_0-prediction. We now relax this discrete choice, by considering any possible scalar value for the coefficient $a_j^p(t)$, beyond $a_j^{x_0}(t) = 1/t$ and $a_j^{x_1}(t) = 0$. Minimizing the residual criterion $\mathcal{R}_{t,j}$ in (12) over $a_j(t)$ yields $a_j^\star(t) = t\lambda_j/\sigma_{t,j}^2$, to which we associate the linear operator

$$A^\star(t) := t\Sigma_1 \left((1-t)^2\sigma_0^2 I + t^2\Sigma_1\right)^{-1} = U \operatorname{diag}\left(a_1^\star(t), \dots, a_d^\star(t)\right) U^\top. \tag{14}$$

Our intuition for this choice of analytical part is that it makes the network's task the easiest, as it minimizes the magnitude of what remains to be learned by the network to approach the optimal denoiser $D^\star$. Minimizing $\mathcal{R}_{t,j}$ determines the analytical coefficient $a_j^\star(t)$, but not the scale of the learned correction, since any nonzero factor can be absorbed into the residual network. We choose the same coefficient $b(t) = 1 - t$ in every direction, which gives the desired limiting behavior: at $t = 0$, $A^\star(0) = 0$ and $b(0) = 1$, recovering x_1-prediction, while as $t \to 1$, $A^\star(t) \to I$ and $b(t) \to 0$, like x_0-prediction. This yields our second proposed model, the continuous denoiser

$$D_\theta^{\text{cont}}(x,t) = A^\star(t)x + (1-t)N_\theta^{\text{cont}}(x,t). \tag{15}$$

This new denoiser corresponds to the prediction target defined as

$$\boxed{y_t^{\text{cont}}\textbf{-prediction:} \qquad Y_t^{\text{cont}} = \frac{1}{1-t}\big(I - tA^\star(t)\big)X_1 - A^\star(t)X_0.} \tag{16}$$

The analytical term $A^\star(t)(x)$ is the Wiener estimator of X_1 from X_t, i.e. the optimal linear estimator under squared error. For Gaussian data, it is also the optimal estimator. Hence, if the data were Gaussian, the network wouldn't have anything to do; for real data, it must learn the nonlinear correction to this covariance-based prediction.

4 Architecture as an information bottleneck

The previous sections study prediction parameterization independently of architectural information loss, yet previous work indicates that architecture and optimal parametrization are connected (Gagneux et al., 2026). To characterize this connection, we consider architectural bottlenecks that discard input directions. This allows us to determine which directions each parameterization tends to preserve and how it behaves when some are unavailable, thus explaining some previously observed failures and successes in the training of flow matching.

Compression changes target preference. We first investigate empirically whether architectural bottlenecks change the relative performance of prediction parameterizations. Following Li and He (2025), we introduce a linear bottleneck of dimension r inside each ViT patch embedding. A patch $X_t^{(k)} \in \mathbb{R}^{Cp^2}$ obtained by flattening $p \times p$ pixels with C channels is mapped as $H_t^{(k)} = BAX_t^{(k)}$, with $A \in \mathbb{R}^{r \times Cp^2}$ and $B \in \mathbb{R}^{h \times r}$; the Transformer token dimension h is kept fixed and only r is varied. We quantify the amount of compression by $\beta := \frac{Cp^2}{r}$ with $\beta > 1$ imposing an explicit rank bottleneck.

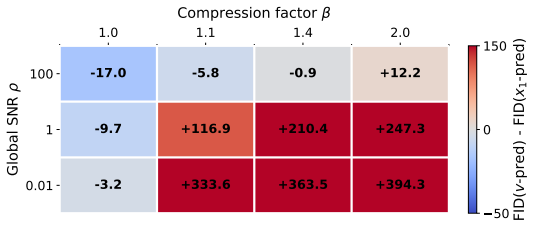


Figure 3: Joint effect of architectural compression and global SNR on CIFAR-10 generation. Increasing compression favors x_1-prediction, whereas increasing ρ favors velocity prediction. Generated samples are shown in Figure 10.

This construction provides a controlled way of discarding information from each patch while keeping the token dimension fixed. Without a bottleneck, the compression ratio becomes $(Cp^2)/h$: thus increasing the patch size p or decreasing the token dimension h also increases compression. Yet, varying either p or h changes the architecture while varying the bottleneck

dimension r does not, enabling us to study compression *in isolation*. This setting is also motivated by the benefits of bottlenecks for generation quality observed in Li and He (2025).

Using the FID (Heusel et al., 2017), Figure 3 reports the CIFAR-10 comparison through the difference $\mathrm{FID}(v) - \mathrm{FID}(x_1)$: positive values favor x_1-prediction and negative values favor v-prediction. Two clear effects emerge. First, at fixed ρ, increasing compression systematically shifts the comparison toward x_1-prediction, consistently with the observations of Li and He (2025). Second, at fixed compression, decreasing ρ produces the same shift toward x_1-prediction, in agreement with the SNR dependence identified in the previous section. Architectural and training details are given in App. D.1.

Which spectral directions does each parametrization preserve? We now connect this notion of compression to the spectral directional analysis conducted in Section 3. To understand why different prediction parameterizations react differently to compression, we consider a controlled Gaussian experiment based on the covariance of CIFAR-10 patches and learn a network including a rank-r linear bottleneck for each prediction parameterization.

In Figure 4 (row 1), we observe that the learned bottleneck depends on the target and aligns well with the PCA decomposition. **For x_1-prediction, the learned bottleneck preserves the r leading PCA directions. For velocity prediction, the retained directions vary more strongly with time**, indicating that different parts of the spectrum are needed at different stages of the trajectory. For figure with all predictions see App. D.2.

This behavior follows directly from the residual criterion $\mathcal{R}^p_{t,j}$ introduced in (12). Fix $t \in (0,1)$ and consider a rank-r bottleneck retaining the directions indexed by $S \subset \{1,\dots,D\}$, with $|S| = r$. The denoising MSE decomposes as

$$\mathbb{E}\Big[\|D^p_\theta(X_t,t) - X_1\|^2\Big] = \mathcal{L}^\star_t + \underbrace{\sum_{j\in S} \mathbb{E}\Big[\Big(D^p_{\theta,j}(X_{t,j},t) - D^\star_j(X_{t,j},t)\Big)^2\Big]}_{\text{error on retained dir.}} + \underbrace{\sum_{j\notin S} \mathcal{R}^p_{t,j}}_{\text{error on discarded dir.}},$$

where $\mathcal{L}^\star_t := \mathbb{E}[\|D^\star(X_t,t) - X_1\|^2]$ is the irreducible denoising error and is independent of S. For an ideal network, the error on retained directions vanishes. Therefore, minimizing the fixed-time denoising MSE over S amounts to minimizing $\sum_{j\notin S} \mathcal{R}^p_{t,j}$, so the optimal rank-$r$ bottleneck retains the r directions with largest $\mathcal{R}^p_{t,j}$. Comparing the two rows of Figure 4 shows that the learned bottlenecks closely follow this predicted retained direction.

What happens to discarded directions? The previous subsection studies which directions a bottleneck tends to retain. We now consider the complementary question: *what happens when a direction is discarded by the bottleneck?*
We return to the Gaussian model and consider a PCA direction u_j that is absent from a PCA-aligned bottleneck. The network can then predict the correction in direction u_j only from the retained coordinates. However, under the Gaussian model, PCA coordinates are independent, so the retained coordinates contain no information about the target in direction u_j. The network's optimal squared-error prediction in direction j is therefore zero.

Let $X^p(t)$ denote the inference-time trajectory generated by the trained model under parameterization p, initialized from $X^p(0) = X_0$, and let $X^p_j(t)$ denote its coordinate along u_j. Projecting the sampling dynamics onto the discarded direction and using the zero network prediction gives, at $t = 1$ (see App. D.3), $X^{x_1}_j(1) = 0$, $X^v_j(1) = X_{0,j}$, $X^{\mathrm{bin}}_j(1) = X^{\mathrm{cont}}_j(1) = \frac{\sqrt{\lambda_j}}{\sigma_0} X_{0,j}$. Consequently, since $X_{0,j} \sim \mathcal{N}(0,\sigma_0^2)$,

$$\mathrm{Var}\big(X^{x_1}_j(1)\big) = 0, \qquad \mathrm{Var}\big(X^v_j(1)\big) = \sigma_0^2, \qquad \mathrm{Var}\big(X^{\mathrm{bin}}_j(1)\big) = \mathrm{Var}\big(X^{\mathrm{cont}}_j(1)\big) = \lambda_j. \quad (17)$$

Thus, x_1-prediction suppresses variability along a discarded direction, collapsing its generated coefficient to 0 (i.e. the data mean), whereas velocity prediction preserves the source variance σ_0^2. **Remarkably, both spectral hybrids recover exactly the target variance λ_j, although the direction is never available to the learned residual.** Figure 5 shows remarkable alignment between this theory and practice, even on non-Gaussian CIFAR-10 patches. Experimental details and additional controls are given in App. D.3.

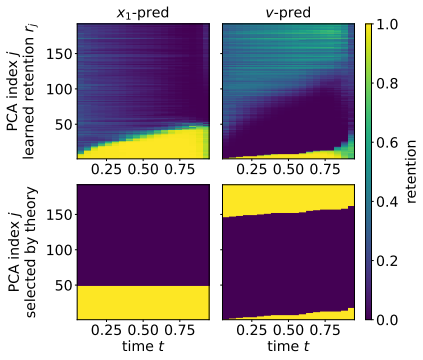


Figure 4: Theoretical and learned spectral directions under a rank-r bottleneck. *Top*: retention measured from the learned bottleneck, $r_j^p(t) = \|P_{W_t^p} u_j\|^2$, where P_{W_t} projects onto the subspace preserved by W_t. $r_j^p(t) \simeq 1$ indicates that the PCA direction u_j is preserved. *Bottom*: The r PCA directions with largest residual criterion $\mathcal{R}_{t,j}^p$.

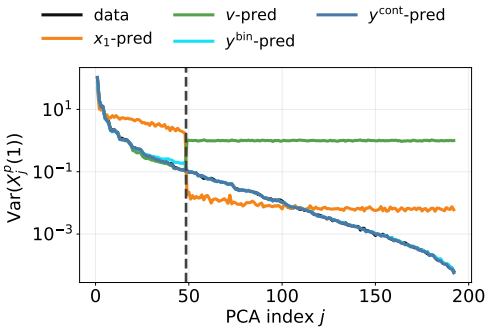


Figure 5: Coordinatewise variance of generated samples using the same fixed rank-48 top-PCA bottleneck for all parameterizations. Beyond the cutoff, x_1-prediction suppresses variability and velocity prediction retains the source variance, whereas the spectral hybrids closely recover the target variance spectrum, in agreement with (17).

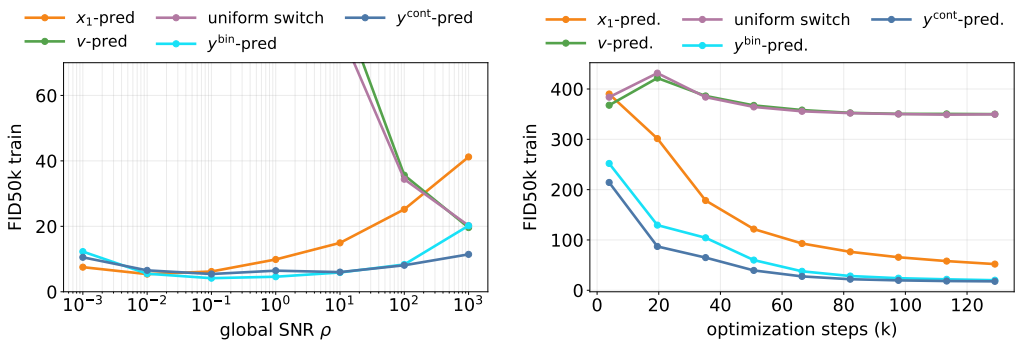
Remark 2. The analysis above is a first step toward understanding compression in Vision Transformers. Here, the image is treated as a single patch, so the bottleneck and the spectral hybrid act in the same covariance basis. In a standard ViT, the bottleneck acts in local patch space whereas the hybrid uses global image statistics, so the exact variance-matching argument no longer applies directly. Nevertheless, the analysis highlights mechanisms that may also contribute to compression effects in practical ViTs.

5 Experiments

We now evaluate the proposed spectral parameterizations on standard image-generation settings. Figure 6a illustrates the impact of the global SNR on the model's final performance on CelebA (Liu et al., 2015), and Figures 6b and 7 show faster training on standard image datasets Bedroom-64 (Yu et al., 2015) and AFHQ-256 (Choi et al., 2020). We use alternatively U-Net architectures (Ronneberger et al., 2015) and ViTs (Dosovitskiy et al., 2021) from the implementation of Li and He (2025), and use the Adam optimizer (Kingma and Ba, 2015).

Robustness to source scale and architecture On CelebA-64, we vary only the source scale to obtain $\rho \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ and compare the fixed prediction targets with the binary and continuous spectral models. We also include a parametrization with an x_1/x_0 switch at $t = 1/2$ *uniformly* in every spectral direction. This baseline tests whether the gain of the proposed models comes from avoiding the ill-conditioned x_0-prediction regime near $t = 0$, or from their spectral adaptation, which selects a different parameterization across directions according to their variance. As can be seen in Figure 6a; the spectral models remain close to the best fixed target across the SNR range, whereas the uniform switch does not provide the same robustness. This shows that adapting the parameterization across spectral directions is crucial.

Optimization speed-ups We next evaluate FID throughout training on a subset of Bedroom-64 (100,000 training images) at $\sigma_0 = 1$, for the same U-Net and ViT architectures. As shown in Figure 6b (full results with UNet in Figure 15), the hybrid predictions y^{cont} and y^{bin} consistently converge faster than the best fixed target. For the U-Net, it reaches the low-FID regime noticeably earlier than v-prediction. For the compressed ViT, the gain is even larger: y^{cont} reaches FID below 60 after about 35k optimization steps, whereas x_1-prediction needs three times as many steps to reach a similar level. These results are



(a) Final FID versus SNR (CelebA-64) (b) FID throughout training (Bedrooms-64)

Figure 6: Generative performance (FID) for a ViT with patch size 4 and compression factor 2. *Left:* On CelebA-64, we compare x_1-pred, v-pred, a fixed x_1/x_0 switch at $t = 1/2$ (uniform switch), and the binary and continuous spectral parameterizations. The two hybrid parameterizations Y^{bin} and Y^{cont} are robust to SNR variations and architectural compression, and in the worst case nearly match the best fixed prediction target. *Right*: FID throughout training on Bedroom-64. Our spectral hybrid parameterizations converge substantially faster.

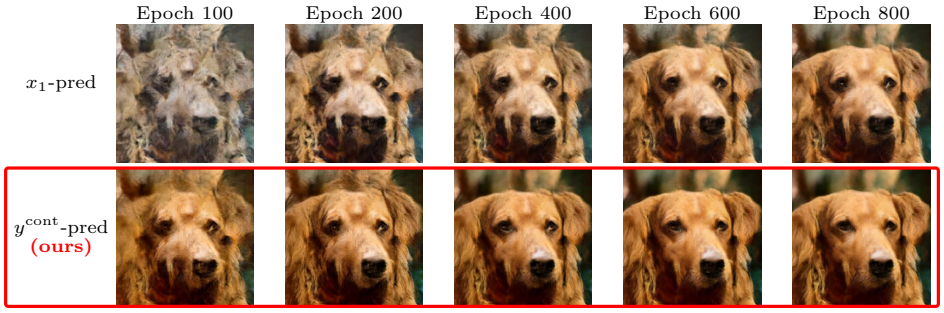


Figure 7: AFHQ-256 samples generated from the same fixed source sample throughout training, using a ViT-16 without bottleneck. The best fixed prediction target here is x_1-pred.

consistent with the residual construction: the analytical spectral term reduces what remains to be learned by the network, which can translate into faster optimization.

We finally complement this study with a higher-resolution experiment on AFHQ-256, using a ViT-16 without bottleneck and token size 768. In this setting, x_1-prediction is the strongest fixed target, and we compare it directly with the continuous spectral model, implemented in a DCT basis. The results, reported in Figure 7, show that the hybrid parametrization allows to recover better images and finer details, as suggested by our theory in Section 4.

6 Conclusion

We showed that prediction target preference depends on both SNR and architecture, motivating new spectral parameterizations that adapt across time and directions. These are robust across regimes and can substantially accelerate optimization, with no additional training cost beyond computing the data PCA once beforehand. While PCA computation becomes difficult in large scale settings, our AFHQ experiments show that a fixed DCT basis can provide an effective alternative. Several questions remain open. In particular, the relative advantage of the binary and continuous models may depend on how accurately a Gaussian second-order approximation describes the data, as suggested by the spiral experiments in Figure 17. We also deliberately decoupled parameterization from loss weighting, although the two choices may interact.

References

Albergo, M. S. and Vanden-Eijnden, E. (2023). Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations (ICLR)*.

Benny, Y. and Wolf, L. (2022). Dynamic dual-output diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press.

Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., and Yoon, S. (2022). Perception prioritized training of diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). StarGAN v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

De Bortoli, V. (2022). Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research (TMLR)*.

Degeorge, L., Couairon, P., Ghosh, A., Efros, A. A., Picard, D., and Kalogeiton, V. (2026). Balancing frequencies and pixels in flow matching. *arXiv preprint arXiv:2609.02748*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Esteves, C. and Makadia, A. (2026). Spectrally-guided diffusion noise schedules. *arXiv preprint arXiv:2603.19222*.

Falck, F., Pandeva, T., Zahirnia, K., Lawrence, R., Turner, R., Meeds, E., Zazo, J., and Karmalkar, S. (2025). A Fourier space perspective on diffusion models. *arXiv preprint arXiv:2505.11278*.

Finn, E., Wang, B., Keller, T. A., and Ba, D. E. (2026). Where the score lives: A wavelet view of diffusion. *arXiv preprint arXiv:2606.08309*.

Fu, F., Wang, T., Zhou, G., Cen, J., and Zhu, Q. (2026). JLT: Clean-latent prediction in latent diffusion transformers. *arXiv preprint arXiv:2605.27102*.

Gagneux, A., Martin, S. T., Gribonval, R., and Massias, M. (2026). Training flow matching: The role of weighting and parameterization. In *International Conference on Learning Representations (ICLR) 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

Geng, Z., Lu, Y., Wu, Z., Shechtman, E., Kolter, J. Z., and He, K. (2026). Improved mean flows: On the challenges of fastforward generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guo, S., Ma, L., Tong, H., He, Y., Guo, Z., Liu, X., Yan, F., Guo, Y., Wang, F., Huang, J., et al. (2026). Pixel-space diffusion via observation operators. *arXiv preprint arXiv:2608.21885*.

Han, X., Hu, J., and Liu, L. (2026). Self-consistent flow: Unifying velocity and endpoint prediction for rectified flow models. *Transactions on Machine Learning Research (TMLR)*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hoogeboom, E. and Salimans, T. (2023). Blurring diffusion models. In *International Conference on Learning Representations (ICLR)*.

Jin, Q. and Wang, C. (2026). Revisiting diffusion model predictions through dimensionality. *arXiv preprint arXiv:2601.21419*.

Kamb, M. and Ganguli, S. (2025). An analytic theory of creativity in convolutional diffusion models. In *International Conference on Machine Learning (ICML)*.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kim, J., Park, J., Jeon, S., and Kim, S. (2026). Better source, better flow: Learning condition-dependent source distribution for flow matching. *arXiv preprint arXiv:2602.05951*.

Kingma, D. and Gao, R. (2023). Understanding diffusion objectives as the ELBO with simple data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324.

Lee, J., Kim, K., and Lee, J. (2026). Is there a better source distribution than Gaussian? exploring source distributions for image flow matching. *Transactions on Machine Learning Research*.

Li, T. and He, K. (2025). Back to basics: Let denoising generative models denoise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*.

Liu, X., Gong, C., and Liu, Q. (2023). Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision (ICCV)*.

Ning, M., Li, M., Su, J., Haozhe, J., Liu, L., Benes, M., Chen, W., Salah, A. A., and Ertugrul, I. O. (2025). DCTdiff: Intriguing properties of image generative modeling in the DCT space. In *International Conference on Machine Learning (ICML)*.

Peebles, W. and Xie, S. (2023). Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision (ICCV)*.

Rissanen, S., Heinonen, M., and Solin, A. (2023). Generative modelling with inverse heat dissipation. *International Conference on Learning Representations (ICLR)*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer.

Salimans, T. and Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Tian, K., Jiang, Y., Yuan, Z., Peng, B., and Wang, L. (2024). Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

van der Schaaf, A. and van Hateren, J. (1996). Modelling the power spectra of natural images: statistics and information. *Vision research*, 36(17):2759–2770.

Wang, B. and Vastola, J. (2024). The unreasonable effectiveness of Gaussian score approximation for diffusion models and its applications. *Transactions on Machine Learning Research (TMLR)*.

Yao, J., Wang, C., Liu, W., and Wang, X. (2024). FasterDiT: Towards faster diffusion transformers training without architecture modification. *Advances in Neural Information Processing Systems (Advances in Neural Information Processing Systems (NeurIPS))*, 37:56166–56189.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

AI USE STATEMENT

In this work, we used generative AI tools for research ideation and execution, including brainstorming on prediction parameterizations, source–data SNR, spectral switching, and architectural compression; for checking and improving mathematical derivations, including directional switching rules and residual formulations; and for discussing the design and interpretation of controlled experiments, including the spiral, PCA-bottleneck, and spectral-hybrid experiments. We also used generative AI to prototype code for some synthetic diagnostic experiments and to assist with drafting and restructuring parts of the manuscript. Additionally, we used generative AI for literature discovery and comparison with related work, and for polishing the writing. No other required-disclosure uses are applicable to this work.

We reviewed all AI-assisted work. In particular, we compiled and checked the final bibliography manually, verified the mathematical claims and derivations, and checked the implementation and experimental results. We take responsibility for the final content of this work.

A BACKGROUND ON PARAMETRIZATION AND LOSS

A.1 EQUIVALENCE BETWEEN PARAMETRIZATIONS AND LOSSES

A fundamental observation is that for $X_t = (1-t)X_0 + tX_1$, the ideal velocity $\mathbb{E}[X_1 - X_0 \mid X_t = x]$, the ideal denoiser $\mathbb{E}[X_1 \mid X_t = x]$ and the ideal noise estimator $\mathbb{E}[X_0 \mid X_t = x]$ are all algebraically related to each other. For example, the ideal velocity is linked to the ideal denoiser by:

$$\mathbb{E}[X_1 - X_0 \mid X_t = x] = \mathbb{E}[\tfrac{X_1 - X_t}{1-t} \mid X_t = x] = \frac{\mathbb{E}[X_1 \mid X_t = x] - x}{1-t} \tag{18}$$

Therefore, from a network $N_\theta^{x_1}$ outputting the clean image X_1, one can obtain a prediction for the velocity as $\frac{N_\theta^{x_1}(x,t)-x}{1-t}$. As a consequence, this network can be trained by regressing against the velocity, by using as a loss:

$$\mathbb{E}_{x_0,x_1,t}[\|\tfrac{N_\theta^{x_1}(x,t)-x}{1-t} - (x_1 - x_0)\|^2] \tag{19}$$

Interestingly, this loss rewrites $\mathbb{E}_{x_0,x_1,t}[\frac{1}{(1-t)^2}\|N_\theta^{x_1}(x,t) - x_1\|^2]$: regressing against the velocity is actually regressing against the clean image under a time dependent weighting of $(1-t)^{-2}$. The same argument applies to any pair of prediction and regression target: each reduces to a weighted regression onto the network's output, with a weight specific to the pair.

The 3×3 possible combination of loss and prediction are summarized in Table 1 (included for completeness, it also be found e.g. in Li and He (2025, Table 1)).

Derivation of the denoiser for each parametrization in Eq. (3)): Using $X_t = (1-t)X_0 + tX_1$, one can easily check that:

$$\mathbb{E}[X_1 \mid X_t = x] = x + (1-t)\mathbb{E}[X_1 - X_0 \mid X_t = x] \tag{20}$$

$$= \frac{x - (1-t)\mathbb{E}[X_0 \mid X_t = x]}{t} \tag{21}$$

Hence, the natural denoiser for v-pred is $\mathrm{Id} + (1-t)N_\theta^v$, while for x_0-pred it is $\frac{\mathrm{Id}-(1-t)N_\theta}{t}$

A.2 ERROR AMPLIFICATION FOR DIFFUSION PATHS

The amplification discussed in Remark 1 depends on the interpolation path. Consider the general Gaussian interpolation

$$X_t = \alpha_t X_1 + \sigma_t X_0, \qquad X_0 \sim \mathcal{N}(0, I), \tag{22}$$

Table 1: Combinations of prediction and loss. The network N_θ^p outputs $p \in \{x_0, v, x_1\}$ (row); the loss regresses against another (column), possibly after algebraic conversion. Every cell reduces to a weighted regression onto the network's own output, $\mathbb{E}_{x_0,x_1,t}\big[w(t)\,\|N_\theta^p(x_t,t)-p\|^2\big]$

	x_1-loss	v-loss	x_0-loss
x_1-pred	1	$\frac{1}{(1-t)^2}$	$\frac{t^2}{(1-t)^2}$
v-pred	$(1-t)^2$	1	t^2
x_0-pred	$\frac{(1-t)^2}{t^2}$	$\frac{1}{t^2}$	1

where, as in the main text, $t = 0$ corresponds to the source distribution and $t = 1$ to the data distribution. Differentiating gives

$$\dot{X}_t = \dot{\alpha}_t X_1 + \dot{\sigma}_t X_0. \tag{23}$$

The optimal velocity writes

$$V^\star(x,t) = \mathbb{E}[\dot{X}_t \mid X_t = x] = \frac{\dot{\alpha}_t}{\alpha_t} X_t + \left(\dot{\sigma}_t - \sigma_t \frac{\dot{\alpha}_t}{\alpha_t}\right) \mathbb{E}[X_0 \mid X_t = x]. \tag{24}$$

For x_0-prediction, we therefore have

$$V_\theta^{x_0}(X_t,t) = \frac{\dot{\alpha}_t}{\alpha_t} X_t + \left(\dot{\sigma}_t - \sigma_t \frac{\dot{\alpha}_t}{\alpha_t}\right) N_\theta^{x_0}(X_t,t). \tag{25}$$

Similarly, for x_1-prediction,

$$V_\theta^{x_1}(X_t,t) = \frac{\dot{\sigma}_t}{\sigma_t} X_t + \left(\dot{\alpha}_t - \alpha_t \frac{\dot{\sigma}_t}{\sigma_t}\right) N_\theta^{x_1}(X_t,t). \tag{26}$$

Hence errors due to training are multiplied respectively by

$$c_{x_0}(t) = \dot{\sigma}_t - \sigma_t \frac{\dot{\alpha}_t}{\alpha_t}, \qquad c_{x_1}(t) = \dot{\alpha}_t - \alpha_t \frac{\dot{\sigma}_t}{\sigma_t}. \tag{27}$$

Consider a variance-preserving diffusion. Let $s = 1 - t$ denote the usual forward diffusion time and

$$\bar{\alpha}_s = \exp\left(-\frac{1}{2}\int_0^s \beta_r\, dr\right), \qquad \bar{\sigma}_s^2 = 1 - \bar{\alpha}_s^2. \tag{28}$$

where β denotes a positive, continuous, and bounded noise-rate schedule. Define $\alpha_t = \bar{\alpha}_{1-t}$ and $\sigma_t = \bar{\sigma}_{1-t}$. A direct differentiation gives

$$\dot{\alpha}_t = \frac{\beta_{1-t}}{2}\alpha_t, \qquad \dot{\sigma}_t = -\frac{\beta_{1-t}\alpha_t^2}{2\sigma_t}, \tag{29}$$

and therefore (using that $\sigma_t^2 + \alpha_t^2 = 1$)

$$|c_{x_0}(t)| = \frac{\beta_{1-t}}{2\sigma_t}, \qquad |c_{x_1}(t)| = \frac{\beta_{1-t}\alpha_t}{2\sigma_t^2}. \tag{30}$$

Near $t = 0$, $\sigma_t \simeq 1$, and thus x_0-prediction does not exhibit the $1/t$ singularity of the interpolation used in flow matching, provided that β_{1-t} remains bounded (e.g., the standard linear β schedule).. By contrast, near $t = 1$, if $\beta_0 > 0$,

$$\int_0^s \beta_r\, dr = \beta_0 s + o(s). \tag{31}$$

Hence

$$\begin{aligned}\bar{\alpha}_s^2 &= \exp\left(-\beta_0 s + o(s)\right) \\ &= 1 - \beta_0 s + o(s),\end{aligned} \tag{32}$$

which implies that

$$\sigma_t^2 \simeq \beta_0(1-t). \tag{33}$$

Therefore

$$|c_{x_1}(t)| \sim \frac{1}{2(1-t)}. \tag{34}$$

Thus x_1-prediction remains ill-conditioned close to $t = 1$, similarly to the $(1-t)x_0 + tx_1$ interpolation.

B Additional related work

Prediction parameterization and weighting. DDPMs popularized noise prediction (Ho et al., 2020), while Salimans and Ho (2022) introduced v-prediction as a better conditioned parameterization for progressive distillation and few-step sampling. Velocity prediction is also the standard regression target in flow matching (Lipman et al., 2023). EDM (Karras et al., 2022) reorganized diffusion models into a modular design space, separating choices related to network preconditioning, training, and sampling.

A separate line of work has studied how the contribution of different noise levels should be weighted during training. Choi et al. (2022) show that different SNR ranges correspond to qualitatively different denoising tasks and propose to emphasize intermediate noise levels that are most relevant to perceptual content. Kingma and Gao (2023) show that many diffusion training objectives can be written as weighted integrals of a denoising loss over noise levels. Different objectives therefore mainly differ in how much weight they assign to each SNR regime.

More recently, Gagneux et al. (2026) found that the weighting $1/(1-t)^2$, corresponding to the standard flow-matching velocity loss written in the denoising representation, provides a robust choice across several architectures and prediction targets. In our experiments, we therefore use the same denoising representation and time weighting for all parameterizations, so that their comparison isolates the effect of the regression target rather than differences induced by loss weighting.

Manifold and intrinsic-dimension explanations. Li and He (2025) motivate direct data prediction through the manifold hypothesis: natural images are concentrated near a low-dimensional set, whereas source noise and velocity contain full-dimensional components. Jin and Wang (2026) formalize a related idea in a linear model and derive a dimension-dependent optimum over a continuous family of prediction targets. Specifically, they consider $U_k = kX_1 - (1-k)X_0$, which encompasses noise-, v-, and data-prediction up to scaling and sign. Their theoretical analysis considers the optimal linear predictor of U_k from X_t and selects k by minimizing the resulting regression loss. In their main result, the data are assumed to lie in a d-dimensional subspace of $\mathbb{R}^D$ and to have unit variance along each direction of this subspace and zero variance in the orthogonal directions. Under these assumptions, they obtain $k^\star = D/(D+d)$ and interpret its dependence on d/D in terms of intrinsic versus ambient dimension.

Global SNR and source distribution To the best of our knowledge, previous works have not considered SNR at the level of individual covariance directions, and we believe this perspective is new. Several works, however, have highlighted the importance of the global SNR ρ for training flow-matching and diffusion models.

FasterDiT (Yao et al., 2024) highlights the importance of the signal-to-noise ratio during training, showing that rescaling the data variance can substantially affect generation quality. Their analysis focuses primarily on the distribution of SNR values induced by the interpolation schedule rather than on the prediction target. They nevertheless observe that v-prediction in flow matching is more robust to scale changes than noise prediction. In contrast, we show that varying the source–data SNR alone can reverse the performance ordering between x_0-, x_1-, and v-prediction, identifying it as a key factor governing the relative effectiveness of prediction parameterizations.

Lee et al. (2026); Kim et al. (2026) study the role of the source distribution in flow matching. Lee et al. (2026) identify source–data norm mismatch as a factor increasing the learning cost and propose rescaling the Gaussian source to match the norm of the data distribution, which is closely related to controlling the relative source–data scale captured by our SNR. Going further, Kim et al. (2026) treat the source distribution itself as a design variable and learn a condition-dependent source, emphasizing the importance of source–data directional alignment. Together, these works show that the conventional Gaussian source is not always the best choice. Their focus, however, is on the design of the source distribution under a

fixed flow-matching prediction objective, rather than on how the source–data configuration affects the relative performance of different prediction parameterizations.

The source–data SNR also appears in the appendix of Jin and Wang (2026), where their optimal target parameter satisfies

$$k^\star = \frac{D}{D + \mathrm{Tr}(\Sigma_1)} = \frac{1}{1+\rho}, \qquad \rho = \frac{\mathrm{Tr}(\Sigma_1)}{D}. \tag{35}$$

This generalizes their dimension-dependent result and makes the dependence on the global SNR explicit. However, this dependence is partly entangled with the k-dependent energy of the regression target itself, since $\mathbb{E}\|U_k\|^2$ is minimized at the same value of k. In contrast, our comparisons map all parameterizations to a common denoising space and use the same time weighting, so that differences cannot be attributed simply to changes in target magnitude.

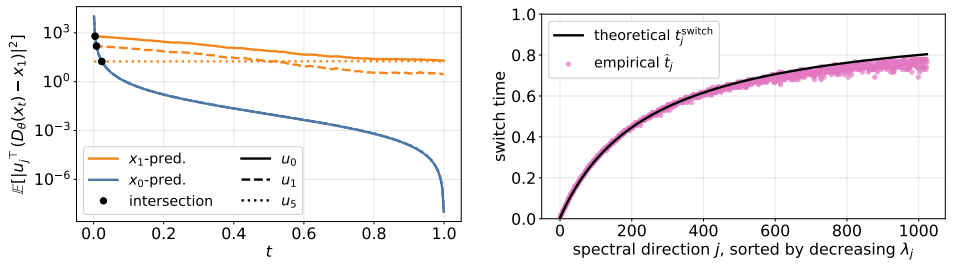
Covariance and spectral explanations. Several recent works emphasize that prediction parameterizations behave differently across covariance directions. Fu et al. (2026) provide a local Gaussian analysis showing that, in low-data-variance directions, the statistics of x_1- and v-prediction differ substantially because the velocity target retains a source-noise component. Importantly, this does not imply that x_1-prediction is intrinsically easier simply because its raw target variance is smaller. In our analysis, all parameterizations are first mapped to a common denoising representation, and we compare the residuals that each parameterization requires the network to learn. This reveals that the relative difficulty is governed by the directional source–data SNR rather than by the raw variance of the regression target alone.

Related works exploit spectral representations or frequency-dependent training to better match the anisotropic structure of image data. Finn et al. (2026) derive an analytically tractable approximation of the score in a wavelet basis, relating denoising behavior to data moments and architectural inductive biases. DCTdiff (Ning et al., 2025) performs diffusion directly in the DCT domain and interprets generation as a coarse-to-fine spectral process, while Esteves and Makadia (2026) use the image power spectrum to design noise schedules. Similarly, Guo et al. (2026) progressively reveal finer image information through time-dependent transformed targets, and Degeorge et al. (2026) rebalances the flow-matching objective across frequencies. Together, these works show that spectral components can have very different denoising and optimization behavior, although they do not study how this heterogeneity changes the relative performance of prediction parameterizations.

A Gaussian approximation closely related to the one underlying our continuous hybrid parameterization was studied by Wang and Vastola (2024). They show that such an approximation captures a substantial part of the score dynamics at moderate-to-high noise levels and use the resulting closed-form dynamics to accelerate sampling. In contrast, in our case, the Gaussian analysis is instead used to derive a covariance-dependent prediction parameterization that is built directly into training.

Hybrid prediction strategies. Benny and Wolf (2022) jointly predict clean data and noise and learn a time-dependent combination of the corresponding denoising updates, motivated by their complementary behavior at different stages of the diffusion trajectory. The combination is global across image directions, whereas our hybrid rule can vary both with time and across covariance directions. More recently, Han et al. (2026) train a single model to predict x_0, x_1, and v simultaneously in flow matching, while enforcing consistency between the three predictions. At sampling time, they use the velocity reconstructed from the x_0- and x_1-predictions during one part of the trajectory, and switch to the directly predicted velocity after a fixed time. Our hybrid parameterizations instead derive this complementarity from source–data statistics.

Architecture and representation bottlenecks. Prediction target also interacts with architectural inductive bias. Gagneux et al. (2026) observed that replacing a U-Net by a patch-based Transformer can reverse the ordering between x_1- and v-prediction. This is consistent with broader analyses showing that neural denoisers are constrained by locality, equivariance, receptive fields, and representation bottlenecks (Kamb and Ganguli, 2025). Section 4 focuses specifically on the information removed by a rank-constrained first representation.



(a) Directional MSE as a function of time (b) t_{switch}: practice confirms theory.

Figure 8: Validation of the theoretical switching time on the anisotropic Gaussian model, for $\rho = 100$ (i.e., $\sigma_0^2 = 0.01$). *Left:* directional MSEs of x_0- and x_1-prediction for three selected directions u_j. The crossing (black dot) occurs earlier in high-variance directions and later in low-variance directions. *Right:* For each spectral direction j, we compare the theoretical switching time t_j^{switch} with the empirical transition time at which, in this direction, the denoising error of the x_0-prediction model becomes smaller than that of the x_1-prediction model (black dot in left figure). The theory agrees strongly with empirical observation.

C Additional Gaussian analysis for the spectral switch

C.1 Directional MSE curves and switching time

For this experiment, we use the synthetic anisotropic Gaussian setup described in the main text. The data endpoint is sampled as $x_1 \sim \mathcal{N}(0, \Sigma_1)$, with eigenvalues $\lambda_j \propto j^{-2}$ normalized so that $\sum_{j=1}^{D} \lambda_j = D$, and with eigenvectors given by a 2D-DCT basis. We use dimension $D = 1024$ and train MLP denoisers with four hidden layers of width 512 for 20,000 Adam steps, batch size 512, and learning rate 10^{-3}.

Figure 8 complements Figure 2 by showing the same plots in the case of a higher global SNR. We see that the same conclusions hold.

It is worth noting that the x_0-prediction curves vary less across directions than the x_1-prediction curves. This is expected: the source distribution is isotropic, so x_0-prediction is trained to recover a target with the same variance in every direction, whereas x_1-prediction directly inherits the anisotropic spectrum of Σ_1. More generally, the dependence of x_0-prediction on the source covariance mirrors that of x_1-prediction on the data covariance.

C.2 Residual comparison including velocity prediction

For completeness, we extend the residual criterion of Section 3.2 to velocity prediction. Under the Gaussian model with isotropic source,

$$\sigma_{t,j}^2 = (1-t)^2\sigma_0^2 + t^2\lambda_j,$$

and the residual quantities are

$$\mathcal{R}_{t,j}^{x_1} = \frac{t^2\lambda_j^2}{\sigma_{t,j}^2}, \qquad \mathcal{R}_{t,j}^{v} = \frac{(1-t)^2\big((1-t)\sigma_0^2 - t\lambda_j\big)^2}{\sigma_{t,j}^2}, \tag{36}$$

$$\mathcal{R}_{t,j}^{x_0} = \frac{(1-t)^4\sigma_0^4}{t^2\sigma_{t,j}^2}. \tag{37}$$

Comparing these three expressions shows that velocity prediction is preferred between the regimes dominated by x_1- and x_0-prediction. More precisely,

$$\operatorname*{argmin}_{p\in\{x_1,v,x_0\}} \mathcal{R}_{t,j}^{p} = \begin{cases} x_1, & t < t_{-,j}, \\ v, & t_{-,j} \le t \le t_{+,j}, \\ x_0, & t > t_{+,j}, \end{cases} \tag{38}$$

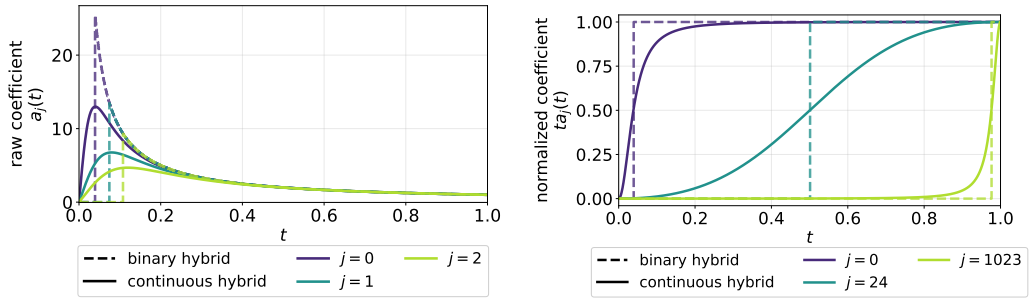


Figure 9: Binary versus continuous spectral parameterizations for selected directions. Left: analytical coefficient $a_j(t)$. Right: normalized coefficient $\alpha_j(t) = ta_j(t)$, highlighting the hard binary switch and its smooth continuous relaxation.

where

$$t_{-,j} = 1 - \sqrt{\frac{\lambda_j}{\sigma_0^2 + \lambda_j}}, \qquad t_{+,j} = \sqrt{\frac{\sigma_0^2}{\sigma_0^2 + \lambda_j}}. \tag{39}$$

Thus x_1-prediction is favored near the source, x_0-prediction near the data, and velocity prediction provides an intermediate regime.

C.3 Comparison of the binary and continuous parameterizations

The continuous model can be viewed as a smooth relaxation of the binary one. As can be seen in Figure 9, for each spectral direction, the binary model switches abruptly between x_1- and x_0-prediction, while the continuous coefficient varies smoothly between the same two regimes.

C.4 Details on the MMSE estimator in the Gaussian case

We derive here the MMSE estimator used in the Gaussian analysis. We first consider the one-dimensional case. Let

$$X_1 \sim \mathcal{N}(m_j, \lambda_j), \qquad X_0 \sim \mathcal{N}(0, \sigma_0^2), \qquad X_0 \perp X_1,$$

Conditionally on $X_1 = x_1$,

$$X_t \mid X_1 = x_1 \sim \mathcal{N}\big(tx_1, (1-t)^2\sigma_0^2\big)\,.$$

Let $s_t^2 = (1-t)^2\sigma_0^2$. By Bayes' rule, $p(x_1 \mid x_t) \propto p(x_t \mid x_1)p(x_1)$, so that, up to terms independent of x_1,

$$\begin{aligned} -\log p(x_1 \mid x_t) &= \frac{(x_1 - m_j)^2}{2\lambda_j} + \frac{(x_t - tx_1)^2}{2s_t^2} + \text{cst} \\ &= \frac{1}{2}\left(\frac{1}{\lambda_j} + \frac{t^2}{s_t^2}\right) x_1^2 - \left(\frac{m_j}{\lambda_j} + \frac{t}{s_t^2} x_t\right) x_1 + \text{cst}. \end{aligned} \tag{40}$$

Completing the square therefore gives

$$\mathbb{E}[X_1 \mid X_t = x_t] = \frac{\frac{m_j}{\lambda_j} + \frac{t}{s_t^2} x_t}{\frac{1}{\lambda_j} + \frac{t^2}{s_t^2}}.$$

After rearranging,

$$\mathbb{E}[X_1 \mid X_t = x_t] = m_j + \frac{t\lambda_j}{(1-t)^2\sigma_0^2 + t^2\lambda_j}(x_t - tm_j). \tag{41}$$

Introducing the centered variables $\bar{X}_t = X_t - tm_j$, and defining $\sigma_{t,j}^2 = (1-t)^2\sigma_0^2 + t^2\lambda_j$, we obtain

$$D_j^\star(\bar{x}, t) = \mathbb{E}[\bar{X}_1 \mid \bar{X}_t = \bar{x}] = \frac{t\lambda_j}{\sigma_{t,j}^2}\bar{x}. \tag{42}$$

We now return to the D-dimensional setting. Let

$$X_1 \sim \mathcal{N}(m, \Sigma_1), \qquad X_0 \sim \mathcal{N}(0, \sigma_0^2 I),$$

and diagonalize $\Sigma_1 = U\Lambda U^\top$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_D)$. In the centered eigenbasis, $\bar{X}_t = U^\top(X_t - tm)$, the coordinates are independent and

$$\bar{X}_{1,j} \sim \mathcal{N}(0, \lambda_j), \qquad \bar{X}_{t,j} = (1-t)\bar{X}_{0,j} + t\bar{X}_{1,j}.$$

Applying (42) coordinatewise yields

$$D_j^\star(\bar{x}_j, t) = \frac{t\lambda_j}{\sigma_{t,j}^2}\bar{x}_j \tag{43}$$

Equivalently,

$$D^\star(\bar{x}, t) = t\Lambda\left((1-t)^2\sigma_0^2 I + t^2\Lambda\right)^{-1}\bar{x}. \tag{44}$$

Transforming back to the original coordinates recovers

$$\mathbb{E}[X_1 \mid X_t = x_t] = m + t\Sigma_1\left((1-t)^2\sigma_0^2 I + t^2\Sigma_1\right)^{-1}(x_t - tm). \tag{45}$$

D Additional details and results on architectural compression

D.1 CIFAR-10 compression and SNR sweep

For the CIFAR-10 sweep in Figure 3, we vary both the compression ratio β and the source variance σ_0^2, and hence the global source–data SNR ρ. We consider patch bottleneck dimensions $d_b \in \{48, 43, 34, 24\}$, corresponding to $\beta = 48/d_b \approx \{1.0, 1.1, 1.4, 2.0\}$, and $\rho \in \{100, 1, 0.01\}$. For each configuration, x_1- and v-prediction models use the same architecture and optimization procedure. All models are trained for 1000 epochs using a ViT with 4×4 patches, a token dimension of 768, 12 Transformer blocks, 6 attention heads, and an MLP expansion ratio of 4. We use Adam with a batch size of 128 and a learning rate of 2×10^{-4}, preceded by a linear warm-up over 5000 optimization steps and held constant thereafter. Gradients are clipped to a maximum norm of 1. We use exponential moving average (EMA) of the model parameters with decay 0.9999, starting after 1000 optimization steps.

D.2 Learned and theoretical spectral allocation

We provide additional details for the spectral-allocation analysis of Section 4. We estimate the covariance of 8×8 RGB CIFAR-10 patches ($D = 192$) and construct the Gaussian distribution

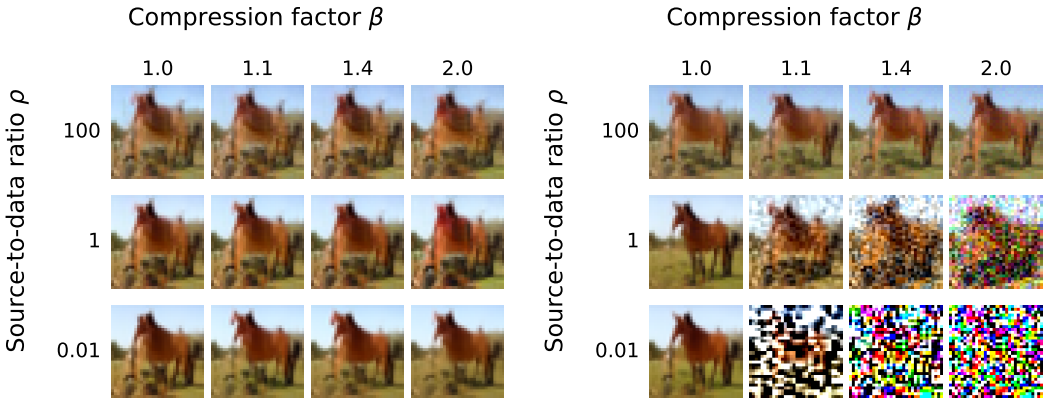
$$X_1^{\mathrm{G}} \sim \mathcal{N}(0, \Sigma_1), \qquad \Sigma_1 = U\,\mathrm{diag}(\lambda_1, \ldots, \lambda_D)U^\top, \tag{46}$$

with isotropic source $X_0 \sim \mathcal{N}(0, \sigma_0^2 I)$.

Learned bottlenecks at fixed time. For several fixed values of t, we independently train an MLP whose first layer is a learned rank-48 bottleneck $W_t^p \in \mathbb{R}^{48\times 192}$. These separate bottlenecks are used only as a diagnostic: they reveal which subspace each parameterization would prefer to preserve at a given time.

This learned allocation can be compared directly with the residual criterion $\mathcal{R}_{t,j}^p$ from (12). For a PCA-aligned bottleneck retaining a set of directions $\{u_j, j \in S\}$ indexed by the set S, the residual that cannot be corrected by the network is

$$\sum_{j \notin S} \mathcal{R}_{t,j}^p. \tag{47}$$



(a) x_1-prediction (b) v-prediction

Figure 10: Representative CIFAR-10 samples generated by a ViT with a patch bottleneck, for different source-to-data ratios ρ and compression factors β. Each panel shows the same generated sample index across the sweep, comparing x_1-prediction and v-prediction.

Thus, under a rank constraint, the optimal bottleneck retains the directions with largest $\mathcal{R}^p_{t,j}$. In particular, $\mathcal{R}^{x_1}_{t,j}$ is increasing in λ_j, so x_1-prediction favors the leading PCA directions, whereas the ordering for velocity prediction changes with time.

To measure which PCA directions are retained by the learned bottleneck W, we define

$$r^p_j(t) := \left\| P_{W^p_t} u_j \right\|^2, \tag{48}$$

where P_W is the orthogonal projector onto the input space preserved by W, i.e., the projection onto $\ker(W)^\perp$. Values $r^p_j(t) \simeq 1$ indicate that u_j is preserved, whereas values close to zero indicate that it is discarded. As shown in Figure 11, the learned bottlenecks closely follow the theoretical allocation predicted by the residual criterion. The agreement is less sharp for some eigenvalues because the residual curves can exhibit nearly flat maxima, making the ranking between neighboring directions ambiguous.

A bottleneck shared across time. A standard time-conditioned model uses a single encoder for all t. We therefore repeat the experiment with one rank-48 bottleneck per parameterization shared across the complete trajectory, and compare its retention with the directions selected by the residual complexity averaged over time. As shown in Figure 12, x_1-prediction remains concentrated on the leading PCA directions, whereas velocity prediction allocates more capacity deeper in the spectrum.

Validation on trained image models. Finally, we verify that this target-dependent allocation is also visible in ViTs trained directly on CIFAR-10. We project their learned patch bottlenecks onto the patch-PCA basis. Despite using the same bottleneck rank, x_1- and velocity prediction retain different spectral directions: x_1 emphasizes the leading PCA directions, whereas velocity allocates more of its representation to later directions.

D.3 Discarded directions

Gaussian analysis. We provide the derivation of the discarded-direction result in (17). For completeness, we consider here the more general case of a potentially non-centered data distribution, i.e., $X_1 \sim \mathcal{N}(m, \Sigma_1)$. Let $X^p(t)$ denote the inference-time trajectory generated by the trained model under parameterization p, and define its centered coordinate along u_j by

$$\bar{X}^p_j(t) := u_j^\top \big(X^p(t) - tm\big). \tag{49}$$

Consider a PCA direction u_j that is completely unavailable to the learned residual. Under the Gaussian model, the PCA coordinates are independent, so the retained representation

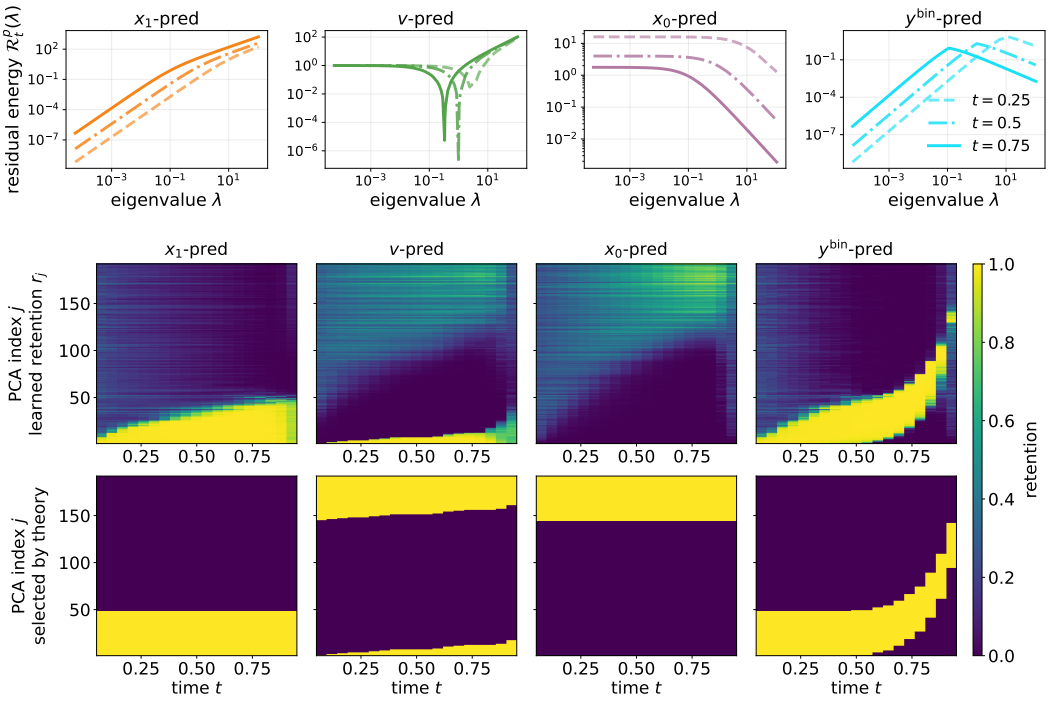


Figure 11: Target-dependent spectral allocation under a rank-48 bottleneck. *Top:* directional residual $\mathcal{R}^p_{t,j}$ across the data spectrum. *Bottom:* theoretical and learned retention of the corresponding PCA directions, measured by $r^p_j(t)$ (48). The learned bottlenecks closely follow the allocation predicted by minimizing the residual (47) : x_1 consistently favors the leading PCA directions, whereas velocity and the binary hybrid exhibit more time-dependent allocations.

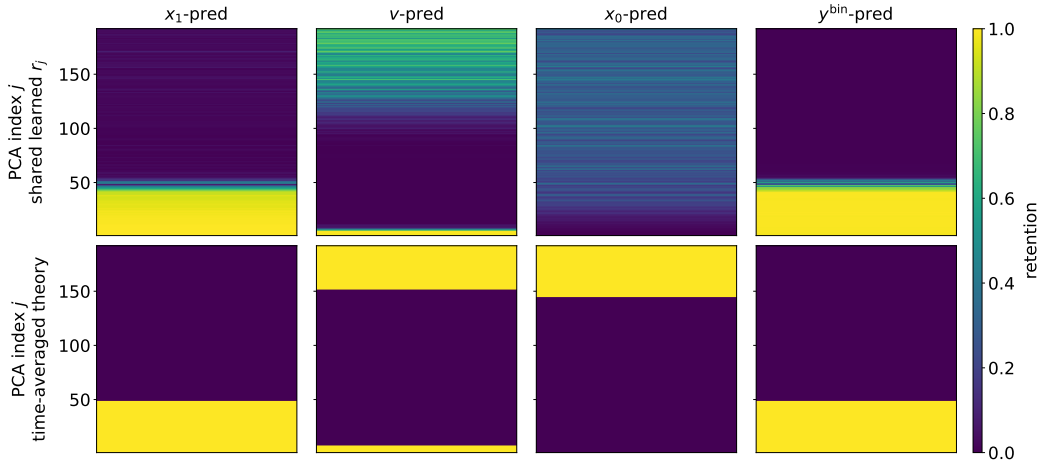


Figure 12: Spectral allocation of a rank-48 bottleneck shared across time. The time-averaged theoretical allocation is compared with the retention measured from the learned bottleneck.

contains no information about the centered target in direction u_j. The optimal squared-error network correction in this direction is therefore zero.

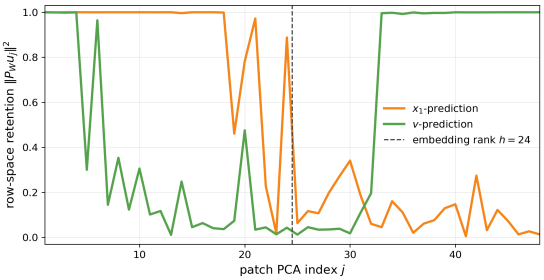


Figure 13: Patch-PCA retention of learned rank-24 bottlenecks in CIFAR-10 ViTs. The two prediction parameterizations learn different retained subspaces.

In centered coordinates, the inference-time dynamics satisfy

$$\frac{\mathrm{d}\bar{X}_j^p(t)}{\mathrm{d}t} = \frac{\bar{D}_{\theta,j}^p(X^p(t),t) - \bar{X}_j^p(t)}{1-t}, \tag{50}$$

where $\bar{D}_{\theta,j}^p$ denotes the centered denoiser coordinate.

For x_1-prediction, the learned output in the discarded direction is zero, so $\bar{D}_{\theta,j}^{x_1} = 0$. Hence

$$\dot{\bar{X}}_j^{x_1}(t) = -\frac{\bar{X}_j^{x_1}(t)}{1-t}, \qquad \bar{X}_j^{x_1}(t) = (1-t)X_{0,j}, \tag{51}$$

and therefore $\bar{X}_j^{x_1}(1) = 0$.

For velocity prediction, a zero learned output gives $\bar{D}_{\theta,j}^v = \bar{X}_j^v(t)$. Consequently, $\dot{\bar{X}}_j^v(t) = 0$ and

$$\bar{X}_j^v(1) = X_{0,j}. \tag{52}$$

For x_0-prediction, a zero learned output gives $\bar{D}_{\theta,j}^{x_0} = \bar{X}_j^{x_0}(t)/t$, and therefore

$$\dot{\bar{X}}_j^{x_0}(t) = \frac{\bar{X}_j^{x_0}(t)}{t}. \tag{53}$$

Starting from any time $\tau > 0$, these dynamics can be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\bar{X}_j^{x_0}(t)}{t}\right) = 0.$$

Hence $\bar{X}_j^{x_0}(t)/t$ is constant along the trajectory, and

$$\bar{X}_j^{x_0}(t) = \frac{t}{\tau}\bar{X}_j^{x_0}(\tau), \qquad t \geq \tau > 0. \tag{54}$$

In particular, $\bar{X}_j^{x_0}(1) = \bar{X}_j^{x_0}(\tau)/\tau$. Pure x_0-prediction cannot be initialized directly at $t = 0$ because these dynamics are singular there.

The binary hybrid avoids this singularity by using x_1-prediction before the directional switching time $\tau_j = t_j^{\text{switch}}$ and x_0-prediction afterwards. The first phase gives

$$\bar{X}_j^{\text{bin}}(\tau_j) = (1-\tau_j)X_{0,j}.$$

Applying (54) from τ_j to 1 then yields

$$\bar{X}_j^{\text{bin}}(1) = \frac{1-\tau_j}{\tau_j}X_{0,j} = \frac{\sqrt{\lambda_j}}{\sigma_0}X_{0,j}, \tag{55}$$

where the second equality follows from $\tau_j = \sigma_0/(\sigma_0 + \sqrt{\lambda_j})$.

Since $X_{0,j} \sim \mathcal{N}(0, \sigma_0^2)$, we obtain $\mathrm{Var}(\bar{X}_j^{x_1}(1)) = 0$, $\mathrm{Var}(\bar{X}_j^{v}(1)) = \sigma_0^2$, and $\mathrm{Var}(\bar{X}_j^{\mathrm{bin}}(1)) = \lambda_j$.

For the continuous hybrid, the learned residual is again zero in the discarded direction. The remaining analytical term is

$$\bar{D}_{\theta,j}^{\mathrm{cont}} = a_j^\star(t)\bar{X}_j^{\mathrm{cont}}(t), \qquad a_j^\star(t) = \frac{t\lambda_j}{(1-t)^2\sigma_0^2 + t^2\lambda_j}. \tag{56}$$

Writing $\sigma_{t,j}^2 = (1-t)^2\sigma_0^2 + t^2\lambda_j$, the corresponding sampling dynamics are

$$\dot{\bar{X}}_j^{\mathrm{cont}}(t) = \frac{a_j^\star(t) - 1}{1-t}\bar{X}_j^{\mathrm{cont}}(t) \tag{57}$$

$$= \frac{t\lambda_j - (1-t)\sigma_0^2}{\sigma_{t,j}^2}\bar{X}_j^{\mathrm{cont}}(t) \tag{58}$$

$$= \left(\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\log\sigma_{t,j}^2\right)\bar{X}_j^{\mathrm{cont}}(t). \tag{59}$$

Integrating from $t = 0$, where $\bar{X}_j^{\mathrm{cont}}(0) = X_{0,j}$, gives

$$\bar{X}_j^{\mathrm{cont}}(t) = \frac{\sigma_{t,j}}{\sigma_0}X_{0,j}. \tag{60}$$

Hence

$$\bar{X}_j^{\mathrm{cont}}(1) = \frac{\sqrt{\lambda_j}}{\sigma_0}X_{0,j}, \qquad \mathrm{Var}\big(\bar{X}_j^{\mathrm{cont}}(1)\big) = \lambda_j. \tag{61}$$

Remarkably, the continuous hybrid recovers the correct Gaussian marginal variance at every time t, not only at $t = 1$, even though the corresponding direction is unavailable to the learned residual.

Why the switch must be direction dependent. The same calculation also shows why a uniform switching time cannot in general recover the covariance of discarded directions. If all directions switch at a common time τ, then

$$\mathrm{Var}\big(\bar{X}_j^{\mathrm{bin}}(1)\big) = \left(\frac{1-\tau}{\tau}\right)^2\sigma_0^2, \tag{62}$$

which is identical for every discarded direction j. It therefore cannot match an anisotropic target spectrum $\{\lambda_j\}_j$. In contrast, the direction-dependent choice $\tau_j = \sigma_0/(\sigma_0 + \sqrt{\lambda_j})$ yields $\mathrm{Var}(\bar{X}_j^{\mathrm{bin}}(1)) = \lambda_j$ independently in every direction.

Non-Gaussian CIFAR-10 patch experiment. For Figure 5, we use the original CIFAR-10 patch distribution rather than its Gaussian approximation. After centering and globally rescaling the patches such that $D^{-1}\,\mathrm{Tr}(\Sigma_1) = 1$, we impose the same fixed rank-48 top-PCA bottleneck

$$W_{\mathrm{PCA}} = U_{1:48}^\top \tag{63}$$

for all prediction parameterizations. Thus, all compressed models receive exactly the same retained coordinates, so differences in the discarded directions arise from the prediction parameterization rather than from the bottleneck itself.

We repeat the experiment with a bottleneck chosen to favor v-prediction. Specifically, we average the velocity residual criterion over time,

$$\overline{\mathcal{R}}_j^v := \int_0^1 \mathcal{R}_{t,j}^v \,\mathrm{d}t, \tag{64}$$

and retain the 48 directions with largest $\overline{\mathcal{R}}_j^v$. Despite this deliberately favorable choice, v-prediction still fails to reproduce the target variance in directions that are unavailable to the network, substantially overestimating a large part of the spectrum. Conversely, x_1-prediction strongly underestimates much of the discarded spectrum. In contrast, both spectral hybrids almost exactly recover the target variance across PCA directions. This confirms that their robustness to discarded information is not specific to the top-PCA bottleneck used in Figure 5.

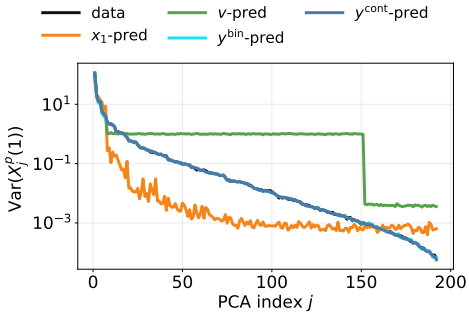
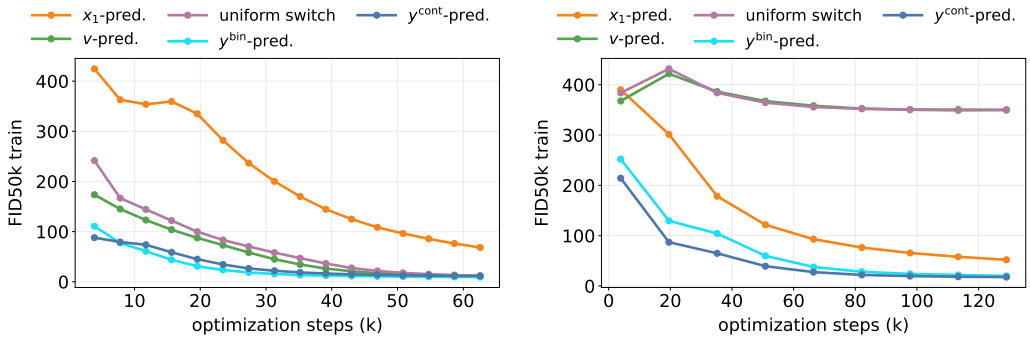


Figure 14: Generation with a fixed rank-48 time-averaged velocity-optimal bottleneck (i.e., the one in Figure 12) on non-Gaussian CIFAR-10 patches. The spectral hybrids again track the target covariance and reduce marginal error across discarded directions, showing that the effect is not specific to a top-PCA bottleneck.

Figure 15: FID on Bedroom-64 throughout training, for a U-Net (left) and a ViT with patch size 4 and compression factor 2 (right). The two spectral hybrid parameterizations, Y^{bin} and Y^{cont}, converge substantially faster toward their final FID than the fixed prediction targets.



(a) With UNet (b) With ViT-4

E Full experimental results

We report here the complete evaluation for both U-Net and compressed ViT architectures. These results complement the ViT results shown in the main paper and confirm that the proposed spectral parameterizations behave consistently across architectures.

On Bedroom-64, Figure 15 shows that the U-Net reaches a similar final FID with v-prediction and with the two spectral parameterizations. However, the spectral models start from substantially better generative performance and converge much faster, reaching their final FID after far fewer optimization steps.

Finally, in Figure 16, we show additional AFHQ samples generated throughout training from a fixed noise realization, for x_1-, v-, and y^{cont}-prediction.

F Evaluating the hybrid parametrizations on the spiral dataset

Figure 17 reveals a marked difference between the two spectral parameterizations. The binary hybrid remains robust across all considered dimensions and SNRs, whereas the continuous hybrid often generates an approximately Gaussian cloud instead of recovering the spiral. We attribute this difference to their distinct dependence on the Gaussian model. The binary parameterization uses the covariance only to select, in each direction and at each time, between the two endpoint parameterizations. In contrast, the continuous parameterization



Figure 16: AFHQ-256 samples generated from two fixed source samples throughout training using a ViT with patch size 16, token dimension 768, and no bottleneck. The same source samples are used at every checkpoint, allowing the evolution of individual generated images to be compared directly.

explicitly incorporates the Gaussian conditional mean through the linear term $A_t^\star X_t$. For the strongly non-Gaussian spiral distribution, the learned residual must therefore correct this very bad linear approximation.

This interpretation is also consistent with the dependence on dimension and SNR. In direction j, the Gaussian contribution is $a_j^\star(t) = t\lambda_j/((1-t)^2\sigma_0^2 + t^2\lambda_j)$ and is therefore weaker when σ_0 is larger (ρ is small). This occurs both at smaller SNR and, at fixed global SNR, at smaller ambient dimension since $\sigma_0^2 = \mathrm{Tr}(\Sigma_1)/(D\rho)$.

G Proposed parameterizations for non-centered data

We now extend the proposed parameterizations to the case

$$X_1 \sim \mathcal{N}(m, \Sigma_1), \qquad X_0 \sim \mathcal{N}(0, \sigma_0^2 I), \tag{65}$$

with X_0 and X_1 independent. Introducing the centered variables $\bar{X}_1 = X_1 - m$ and $\bar{X}_t = X_t - tm$, we have

$$\bar{X}_t = (1-t)X_0 + t\bar{X}_1. \tag{66}$$

Hence, all covariance-based derivations from the centered setting apply unchanged to $(X_0, \bar{X}_1, \bar{X}_t)$. In particular, the switching matrix M_t and the linear coefficient $A^\star(t)$ depend only on the source and data covariances and are therefore unaffected by the data mean.

Binary parameterization. The binary model is trained to predict X_1 directly in the directions selected by M_t and X_0 in the complementary directions, i.e.,

$$N_\theta^{\text{bin}}(X_t, t) \approx M_t X_1 + (I - M_t)X_0. \tag{67}$$

Since

$$X_1 = \frac{X_t - (1-t)X_0}{t}, \tag{68}$$

the conversion to a common X_1-denoising representation is identical to the centered case:

$$D_\theta^{\text{bin}}(x,t) = M_t N_\theta^{\text{bin}}(x,t) + (I - M_t)\frac{x - (1-t)N_\theta^{\text{bin}}(x,t)}{t}. \tag{69}$$

Thus, no explicit mean correction is required for the binary parameterization when the network predicts the original, non-centered variable X_1.

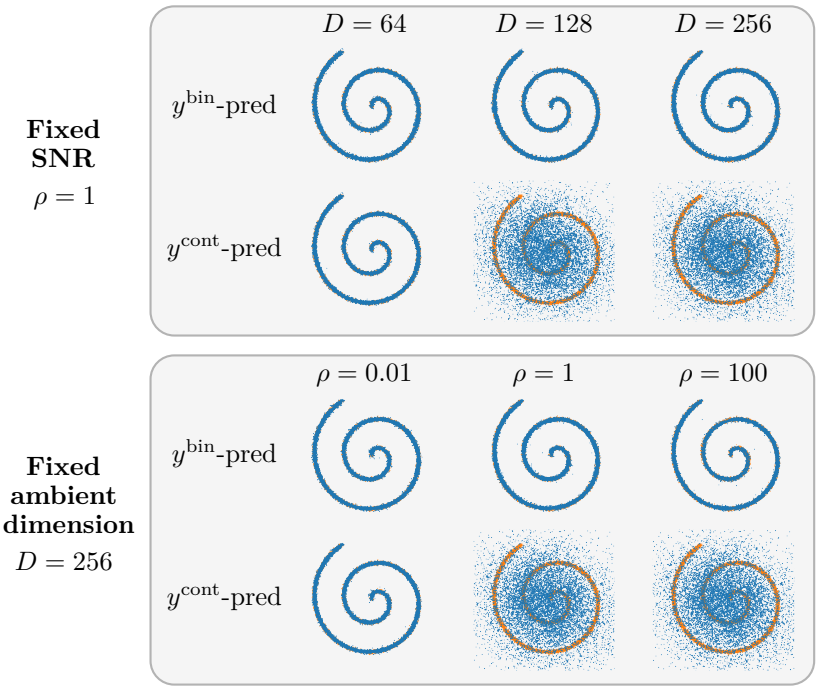


Figure 17: True data are shown in orange and generated samples in blue. *Top:* the SNR is fixed to $\rho = 1$ while the ambient dimension varies. *Bottom:* the ambient dimension is fixed to $D = 256$ while the source–data SNR varies; $\rho \in \{0.01, 1, 100\}$ corresponds respectively to $\sigma_0 \in \{0.6, 0.06, 0.006\}$. We observe that the y^{bin} parametrization achieves good performance in every scenario, whereas the y^{cont} parametrization fails most of the time.

Continuous parameterization. For the continuous model, the linear component derived in the centered setting is applied to $\bar{X}_t = x - tm$. The denoiser therefore becomes

$$D_\theta^{\text{cont}}(x, t) = m + A^\star(t)(x - tm) + (1 - t)N_\theta^{\text{cont}}(x, t). \tag{70}$$

Equivalently, the network is trained to predict the residual

$$Y_t^{\text{cont}} = \frac{I - tA^\star(t)}{1 - t}(X_1 - m) - A^\star(t)X_0. \tag{71}$$

H Implementation details

The hybrid targets are implemented in a spectral basis. For CelebA-64, we use PCA directions computed once from the training set: we collect up to $50,000$ training images, flatten them, subtract the empirical mean, and compute the eigendecomposition of the empirical covariance matrix. These statistics are cached and reused across runs sharing the same dataset and source-noise configuration. For AFHQ-Cat, we instead use an orthonormal two-dimensional DCT applied independently to each RGB channel.

For the binary hybrid Y^{bin}, each spectral direction switches from the x_1 target to the x_0 target at its analytical switching time t_j^{switch}. In all image experiments, we lower-bound the switching times as $t_j^{\text{switch}} \leftarrow \max\left(t_j^{\text{switch}}, 0.1\right)$, to avoid the instability of x_0-prediction near $t = 0$. For the continuous hybrid Y^{cont}, we use the analytical direction-wise coefficient $a_j(t)$ and apply only an upper cap, $a_j(t) \leftarrow \min(a_j(t), a_{\max})$, with $a_{\max} = 10$ for CelebA-64 and $a_{\max} = 100$ for AFHQ-Cat.

For the spiral experiments, we use a MLP with 5 hidden layers, hidden width 512, ReLU activations, and sinusoidal time embeddings of dimension 128. Models are trained for 3000 epochs with batch size 1024, Adam learning rate $2 \cdot 10^{-4}$, default Adam betas $(0.9, 0.999)$, and EMA decay 0.9999.

All other models on image datasets are trained with Adam using a learning rate of 2×10^{-4}, default Adam coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$, and weight decay 10^{-4}. The learning rate is linearly warmed up over the first 5000 optimization steps and then kept constant. We maintain exponential moving average (EMA) weights with decay 0.9999, starting after 1000 optimization steps.

For the CelebA-64 and Bedroom-64 ViT experiments, the ViT models are trained with patch size 4, token dimension 384, patch bottleneck dimension 24, depth 12, MLP ratio 4, and 12 attention heads. Models are trained for 400 epochs with batch size 128. For AFHQ-Cat, we use 256×256 images and a ViT with patch size 16. The corresponding uncompressed patch dimension is $3 \times 16^2 = 768$, and the reported AFHQ-Cat runs use token dimension of 768. The AFHQ-Cat transformer has depth 12, MLP ratio 4, and 6 attention heads. Models are trained for 1000 epochs with batch size 32 and source noise $\sigma_0 = 1$.