Understanding and Learning Prediction Targets in Diffusion Models

Qing Jin*
Independent Researcher
jinqingking@gmail.com

Chaoyang Wang
Independent Researcher

Abstract

Recent advances in diffusion and flow matching models have highlighted a shift in the preferred prediction target—moving from noise ($\boldsymbol{\varepsilon}$) and velocity ($\boldsymbol{v}$) to direct data ($\boldsymbol{x}$) prediction—particularly in high-dimensional settings. However, a formal explanation of why the optimal target depends on the specific properties of the data remains elusive. In this work, we provide a theoretical framework based on a generalized prediction formulation $\boldsymbol{u} = k\boldsymbol{x} - (1-k)\boldsymbol{n}$, where $\boldsymbol{x}$ represents clean data and $\boldsymbol{n}$ denotes noise. This formulation accommodates arbitrary targets, of which $\boldsymbol{\varepsilon}$-, $\boldsymbol{v}$-, and $\boldsymbol{x}$-prediction are special cases. We derive the analytical relationship between data's geometry and the optimal prediction target, offering a rigorous justification for why $\boldsymbol{x}$-prediction becomes superior when the ambient dimension significantly exceeds the data's intrinsic dimension. Furthermore, our controlled analysis reveals how time conditioning and loss variations systematically modulate this optimal objective. While our theory identifies dimensionality as the governing factor for the optimal prediction target, the intrinsic dimension of manifold-bound data is typically intractable to estimate in practice. To bridge this gap, we propose ***k*-Diff**, a framework that learns the optimal parameter k directly from data, bypassing the need for explicit dimension estimation. Extensive experiments in both latent-space and pixel-space image generation demonstrate that k-Diff consistently outperforms fixed-target baselines across varying architectures and data scales, providing a principled and automated approach to enhancing generative performance.

1 Introduction

One of the key design choices of diffusion models [1] is the network output prediction. Early works on low-dimensional data primarily utilize noise prediction ($\boldsymbol{\varepsilon}$-prediction) [2, 3], a paradigm that was later largely superseded by velocity prediction ($\boldsymbol{v}$-prediction) [4, 5], especially following the rise in popularity of flow matching models [6]. Most recently, research into high-resolution generation—particularly for models operating directly in pixel space—has revealed that predicting the clean data ($\boldsymbol{x}$-prediction) often yields superior results and enhanced stability compared to other options [7, 8, 9]. This empirical shift suggests that as we scale models to higher resolutions and more complex data distributions, the "optimal" choice of prediction target is not static but evolves with the task.

Despite these empirical advancements, the theoretical landscape remains fragmented, leaving several fundamental questions unanswered. First, it is unclear whether the optimal prediction target is strictly confined to the conventional triad of $\boldsymbol{\varepsilon}$, $\boldsymbol{v}$, and $\boldsymbol{x}$, or if superior alternatives exist within a more expansive, generalized objective space. For instance, in low-dimensional settings, it remains unproven whether $\boldsymbol{v}$-prediction represents a universal optimum; conversely, for high-dimensional data, we lack a definitive answer as to whether $\boldsymbol{x}$-prediction is an indispensable requirement. Second, the field lacks a rigorous theoretical explanation for how data dimensionality fundamentally dictates the efficiency and stability of these targets. While prior literature has invoked the manifold hypothesis to qualitatively

*Project lead

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

justify $\boldsymbol{x}$-prediction for sparse, high-dimensional data [8], such conceptual frameworks are insufficient for rigorous analysis and cannot substitute for a formal quantitative description. This absence of mathematical precision leads to a further ambiguity: whether a direct, quantifiable relationship exists between the data's ambient or intrinsic dimension and its optimal prediction target, and if so, whether the latter can be analytically derived from the former. Indeed, the precise mechanism by which the optimal target shifts from $\boldsymbol{v}$-prediction toward $\boldsymbol{x}$-prediction as dimensionality increases remains unknown. Finally, from a practical standpoint, there remains no principled methodology to determine the ideal target for a given dataset. Currently, researchers and practitioners are forced to rely on computationally expensive heuristic-driven searches or trial-and-error, which hampers the efficiency of model development for novel high-dimensional domains.

In this work, we systematically address these gaps through a novel theoretical and algorithmic framework. To this end, we first introduce a generic formalism that accommodates prediction targets of arbitrary functional forms, treating standard targets as specific instances of a unified objective. To gain fundamental insights into the training process, we analyze a much simplified diffusion model, which consists of a single linear layer tasked to learn across the entire spectrum of Signal-to-Noise Ratios (SNR) of the diffusion process. Surprisingly, such an SNR-invariant linear model exhibits remarkably rich learning dynamics, offering a tractable yet powerful lens through which to examine the interplay between data dimensionality and prediction targets. Specifically, the optimization dynamics decouple into two distinct modes: a parallel mode aligned with the data manifold and a perpendicular mode orthogonal to it. Notably, we demonstrate that these two modes exhibit fundamentally different learning behaviors; while the parallel mode is governed by data recovery, the perpendicular mode is dominated by noise elimination. Since the relative contribution of these two modes is governed by the dataset's dimensionality, a fixed prediction target—chosen without regard for these geometric properties—is inherently suboptimal.

To further investigate the quantitative relationship between dimensionality and the optimal prediction objective, we introduce a generalized target formulation: $\boldsymbol{u} = k\boldsymbol{x} - (1-k)\boldsymbol{n}$ with $0 \leq k \leq 1$. While mathematically concise, this expression defines a continuous objective space that naturally encompasses conventional targets as discrete points along the spectrum; specifically, $k = 0$, $k = 0.5$, and $k = 1$ recover $\boldsymbol{\varepsilon}$-, $\boldsymbol{v}$-, and $\boldsymbol{x}$-prediction, respectively (up to a constant scaling factor). By optimizing the loss with respect to the parameter k, we derive the optimal target as a function of both the intrinsic and ambient dimension of the data. This derivation establishes a formal, quantitative bridge between the geometric properties of the data manifold and the optimal prediction objective.

Finally, our parameterization of $\boldsymbol{u} = k\boldsymbol{x} - (1-k)\boldsymbol{n}$ also allows us to move beyond the selection of isolated candidates and instead treat the search for an optimal target as a continuous optimization problem over the scalar k. Because intrinsic dimension is difficult to estimate and the learned target can also depend on the denoiser, time sampler, and loss (Section 3.4), we propose k-Diff, which learns a scalar k jointly with the denoiser through backpropagation. Extensive experiments in both latent-space and pixel-space image generation demonstrate that k-Diff consistently achieves comparable or superior results to fixed-target baselines. By adding negligible parameter overhead, k-Diff provides a principled and automated approach to optimizing generative performance across varying data scales and representation spaces.

2 Related Work

Low-Dimensional Diffusion Models The foundations of modern diffusion models [1] were established through $\boldsymbol{\varepsilon}$-prediction objectives [2, 3], which focused primarily on capturing noise patterns in relatively low-resolution or low-dimensional data regimes. Frameworks such as Denoising Diffusion Probabilistic Models (DDPM) [2] demonstrated the initial efficacy of this approach for image synthesis. However, as the field transitioned toward deterministic sampling and straighter probability paths, velocity prediction ($\boldsymbol{v}$-prediction) emerged as a dominant alternative within the Flow Matching [6] and Progressive Distillation paradigms [4]. Notably, through elaborated parameterization, the EDM framework [5] leverages $\boldsymbol{v}$-prediction to ensure the neural network maintains constant output variance across the entire SNR spectrum, thereby significantly easing training convergence. While these models improve stability and enable high-quality generation with fewer sampling steps in moderately high-dimensional spaces, the choice of target remains largely decoupled from the geometric properties of the data.

Pixel-Space High-Resolution Diffusion As generative modeling scaled to high-resolution pixel spaces, the limitations of standard noise and velocity targets became more apparent. Recent large-scale models, particularly those operating directly in pixel space, have increasingly adopted data prediction ($\boldsymbol{x}$-prediction) or heavily weighted SNR-based loss functions to maintain training stability [7, 8, 9]. While empirical evidence suggests that $\boldsymbol{x}$-prediction is superior for handling the sparse structures and high-frequency details of high-resolution images, this shift from $\boldsymbol{v}$ to $\boldsymbol{x}$ has remained largely heuristic. Our work provides the first quantitative bridge explaining why this transition is a natural consequence of the increasing dimensionality of the data manifold.

Learning and Generalization Dynamics of Diffusion Models A growing body of literature seeks to understand the mathematical underpinnings of diffusion models beyond their generative performance, focusing specifically on their ability to generalize versus their tendency to memorize the training set. By investigating the inductive biases inherent in convolutional architectures, [10] establishes a formal link between a model's creativity—its capacity to generate novel samples rather than merely memorizing training data—and the properties of locality and equivariance. They provide both theoretical and empirical evidence that these architectural constraints govern how diffusion models generalize from discrete training data to the broader manifold. Regarding the mechanics of the sampling process, [11] identifies three distinct dynamical regimes within the reverse-time generation process and characterizes two phase transitions between the regimes. The first is the speciation transition, where the evolution moves from pure noise toward the emergence of coarse structures corresponding to a specific class, in a manner analogous to a symmetry-breaking phase transition. This is followed by the collapse transition into a third regime where the diffusion model displays memorization of individual training samples, a phenomenon similar to condensation in a glass phase. In parallel, to understand the acquisition of generation and memorization during training, [12] leverages an analysis of diffusion models using random feature networks and elucidates a transition from effective generation at early training stages to rote memorization in later phases. This training evolution is characterized analytically as a two-timescale process, identifying a specific training window—which is proportional to the dataset size—where the model generates effectively before the onset of memorization. Despite these foundational studies, existing literature primarily addresses the global properties of diffusion training and sampling; the specific impact of the choice of prediction target on these dynamics remains largely unexplored.

Learning and Generalization Dynamics of Neural Networks The study of learning and generalization in high-dimensional neural networks is deeply rooted in the analysis of their training dynamics and the geometric properties of their loss landscapes. Seminal work on weight initialization [13, 14] demonstrated that preserving signal variance across layers is essential for the stability of deep architectures. Furthermore, the analysis of exact solutions for deep linear networks [15] has provided a fundamental understanding of how the spectral properties of the data determine the speed and order of feature learning. [16] characterize the high-dimensional dynamics of generalization error by leveraging random matrix theory, revealing how the ratio of parameters to data and the signal-to-noise ratio govern the emergence of over-fitting and the trade-off between bias and variance. Our work builds upon this tradition by extending these analytical frameworks to the generative setting, utilizing a simplified linear diffusion model to map the relationship between data dimensionality and the optimal prediction objective.

3 Theoretical Analysis of the Optimal Prediction

The general form of the diffusion process involves the noisy input $\boldsymbol{z}$ from the diffusion process and the output or target $\boldsymbol{u}$ to be modeled by neural networks, which are defined as

$$\boldsymbol{z} = \alpha \boldsymbol{x} + \sigma \boldsymbol{n}, \quad \boldsymbol{u} = \varphi \boldsymbol{x} + \psi \boldsymbol{n} \tag{1}$$

Here, $\boldsymbol{x} \in \mathbb{R}^D$ represents the clean data, and $\boldsymbol{n} \in \mathbb{R}^D$ is a Gaussian noise. α, σ, φ, ψ are process parameters that characterize the diffusion process. Typically, α and σ are pre-defined, and φ and ψ are left to be designed based on the needs. For example, for the widely-used flow matching model with $\boldsymbol{v}$-prediction, we have $\alpha = t$ and $\sigma = 1 - t$, where t denotes the forward and backward diffusion time, and $\varphi = 1$, $\psi = -1$. On the other hand, for $\boldsymbol{x}$-prediction, we would have $\varphi = 1$ and $\psi = 0$. Without loss of generality, we choose α as the independent variable, and the other three parameters are functions of α. Meanwhile, for simplicity, we omit the subscript α in $\boldsymbol{z}$ or $\boldsymbol{u}$ although they are functions of α. Note that $\boldsymbol{x}$ and $\boldsymbol{n}$ are constants that do not depend on α.

3.1 Problem Definition

To derive the optimal prediction target, we begin by investigating the learning dynamics of a single-layer linear diffusion model trained on intrinsically low-dimensional data embedded in a high-dimensional space. Following prior work [8], we consider a simplified setting where the data $\boldsymbol{x} \in \mathbb{R}^D$ is projected from an intrinsic latent $\widetilde{\boldsymbol{x}} \in \mathbb{R}^d$ via an orthogonal mapping $\boldsymbol{x} = P\widetilde{\boldsymbol{x}}$. Here, $P \in \mathbb{R}^{D\times d}$ is an arbitrary sub-matrix of a $D \times D$ orthogonal matrix, such that $P^\top P = I_{d\times d}$. In this formulation, d and D denote the intrinsic and ambient dimensions of the data, respectively, while $I_{d\times d}$ represents the $d \times d$ identity matrix.

Consider a diffusion model consisting of a single linear layer $W \in \mathbb{R}^{D\times D}$. The model estimates the target $\boldsymbol{u}$ from the noisy input $\boldsymbol{z}$ via:

$$\hat{\boldsymbol{u}} = W\boldsymbol{z} \tag{2}$$

In practice, rather than directly minimizing the Mean Squared Error (MSE) between $\boldsymbol{u}$ and $\hat{\boldsymbol{u}}$, training often involves an alternative variable $\boldsymbol{w}$—such as the velocity $\boldsymbol{v}$—which is itself a linear combination of data $\boldsymbol{x}$ and noise $\boldsymbol{n}$. As detailed in Appendix A, the errors associated with these two variables are related by a scaling factor κ [17]:

$$\hat{\boldsymbol{w}} - \boldsymbol{w} = \kappa \cdot (\hat{\boldsymbol{u}} - \boldsymbol{u}) \tag{3}$$

Consequently, when training the model using the MSE loss of $\boldsymbol{w}$, the objective function becomes:

$$\mathcal{L} = \frac{1}{2}\int_{\mathcal{I}} \widetilde{\mathcal{D}}\alpha \mathbb{E}\left[\|W\boldsymbol{z} - \boldsymbol{u}\|^2\right] \tag{4}$$

where $\mathbb{E}$ denotes expectation over $\boldsymbol{x}$ and/or $\boldsymbol{n}$, and $\mathcal{I}$ is the interval of integration (typically $[0, 1]$). Here, we define the effective integration measure $\widetilde{\mathcal{D}}\alpha$ as

$$\widetilde{\mathcal{D}}\alpha \coloneqq \mathcal{D}\alpha \cdot \kappa^2 \tag{5}$$

where $\mathcal{D}\alpha$ is the probability measure of the sampling distribution for α (e.g., $\mathcal{U}[0, 1]$ or logit-normal [18]). The term κ^2 thus acts as a loss weighting function dependent on α. For brevity, we hereafter omit the integral interval $\mathcal{I}$ unless explicitly required.

To facilitate a tractable analysis, we assume the intrinsic data $\widetilde{\boldsymbol{x}}$ is whitened:

$$\Sigma_{\widetilde{x}\widetilde{x}} = \mathbb{E}\left[\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^\top\right] = I_{d\times d} \tag{6}$$

This is a widely-adopted simplification in the study of neural network learning dynamics [15, 19], and we provide the analysis for general Σ in Appendix G. Similarly, for Gaussian white noise $\boldsymbol{n}$, we have:

$$\Sigma_{nn} = \mathbb{E}\left[\boldsymbol{n}\boldsymbol{n}^\top\right] = I \tag{7}$$

where I denotes the $D \times D$ identity matrix. Under these assumptions, we now analyze the learning dynamics of the linear diffusion model.

3.2 Learning Dynamics

As derived in Appendix B, given the diffusion process and the linear layer configuration defined above, the network weights W evolve during training according to:

$$\tau\frac{\mathrm{d}W}{\mathrm{d}\widetilde{t}} = -\int \widetilde{\mathcal{D}}\alpha\left(\alpha^2 WPP^\top + \sigma^2 W - \varphi\alpha PP^\top - \psi\sigma I\right) \tag{8}$$

where τ represents the inverse learning rate and $\widetilde{t}$ denotes the training time. Notably, since $PP^\top$ is a projection operator, the dynamics of W can be decoupled into two orthogonal modes: one **parallel** to the data manifold and one **perpendicular** to it.

$$\tau\frac{\mathrm{d}W_\parallel}{\mathrm{d}\widetilde{t}} = -\int \widetilde{\mathcal{D}}\alpha\Big((\alpha^2 + \sigma^2)W_\parallel - (\varphi\alpha + \psi\sigma)PP^\top\Big) \tag{9a}$$

$$\tau\frac{\mathrm{d}W_\perp}{\mathrm{d}\widetilde{t}} = -\int \widetilde{\mathcal{D}}\alpha\Big(\sigma^2 W_\perp - \psi\sigma(I - PP^\top)\Big) \tag{9b}$$

Specifically, we define:

$$W_\parallel = WPP^\top, \quad W_\perp = W(I - PP^\top) \tag{10}$$

which represent the parallel and perpendicular components of W, respectively.

Remarkably, the two modes exhibit distinct dynamics. While the parallel mode depends on data-related parameters such as α and φ, these factors are largely irrelevant to the perpendicular mode. Physically, the parallel mode is driven by the underlying data distribution, resembling a process of **data recovery**. In contrast, the perpendicular mode is not data-driven; its dynamics are dominated by **noise elimination** within the ambient space.

Furthermore, these diverging dynamics impose different requirements on the parameters φ and ψ. An optimal design must therefore account for the competing impacts of both modes. Based on these dynamics, we arrive at the following theorem regarding the equilibrium weights and the resulting loss.

Theorem 3.1. *Under the loss and learning dynamics defined in Eq. (4) and Eq. (9), the optimal weight W^* is achieved at the equilibrium of both modes, given by:*

$$W^* = W_{\parallel}^* + W_{\perp}^* = \frac{\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)} PP^\top + \frac{\int \widetilde{\mathcal{D}}\alpha\psi\sigma}{\int \widetilde{\mathcal{D}}\alpha\sigma^2}(I - PP^\top) \tag{11}$$

The corresponding optimal loss is obtained by substituting W^* into Eq. (4):

$$\Delta^* = \underbrace{\frac{1}{2}d\Big(\int \widetilde{\mathcal{D}}\alpha(\varphi^2 + \psi^2) - \frac{\big(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)\big)^2}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)}\Big)}_{\text{Parallel Contribution}} + \underbrace{\frac{1}{2}(D-d)\Big(\int \widetilde{\mathcal{D}}\alpha\psi^2 - \frac{\big(\int \widetilde{\mathcal{D}}\alpha\psi\sigma\big)^2}{\int \widetilde{\mathcal{D}}\alpha\sigma^2}\Big)}_{\text{Perpendicular Contribution}} \tag{12}$$

Remark. Based on the optimal loss derived in Eq. (12), we can make several key observations. First, the optimal loss is composed of two distinct contributions: the **intra-manifold loss**, arising from the data manifold, and the **residual loss**, originating from the perpendicular ambient space. The intra-manifold loss is proportional to the **intrinsic dimension** d of the manifold on which the data reside. Conversely, the residual loss from the perpendicular ambient space is proportional to the **codimension** $(D - d)$. Second, consistent with the learning dynamics and the optimal weights, the intra-manifold loss depends on the data-dependent parameters α and φ. However, the contribution from the perpendicular ambient space is entirely independent of these factors. Third, in low-dimensional regimes where $D \approx d$, the intra-manifold contribution dominates the total loss. In contrast, for high-dimensional data where $D \gg d$, the ambient contribution becomes the primary contributor. Fourth, by setting $\psi = 0$—which corresponds to $\boldsymbol{x}$-prediction—the perpendicular contribution of the loss vanishes, leaving only the intra-manifold contribution to determine the total loss.

Up to this point, our analysis has addressed the general case where the parameters $(\alpha, \sigma, \varphi, \psi)$ may take arbitrary values. To further elucidate the impact of the dimensions d and D, one could theoretically minimize the optimal loss in Eq. (12) with respect to φ and ψ using variational methods (assuming α and σ are predefined). However, such a procedure is highly complex, as the results depend not only on the dimensions d and D but also on several auxiliary factors. These include the choice of the optimization objective (which determines the scaling factor κ), the sampling distribution of α used during training, and the specific integration interval $\mathcal{I}$.

Therefore, in the following section, we introduce further simplifications to gain deeper insights into how the optimal prediction targets are fundamentally determined by the relationship between the dimensions d and D.

3.3 k-Parameterized Diffusion Prediction Target

To further elucidate the influence of the intrinsic and ambient dimensions on the optimal prediction target, we introduce a simplification where $\varphi = k$ and $\psi = -(1 - k)$. Here, $0 \leq k \leq 1$ is a constant parameter independent of α. Under this parameterization, the network prediction target takes the form:

$$\boldsymbol{u} = k\boldsymbol{x} - (1 - k)\boldsymbol{n} \tag{13}$$

We further specialize our analysis to the flow matching framework, setting $\alpha = t$ and $\sigma = 1 - t$. Notably, up to a constant scaling factor, the conventional $\boldsymbol{\varepsilon}$-, $\boldsymbol{v}$-, $\boldsymbol{x}$-predictions are recovered as special cases where $k = 0$, $k = 0.5$, and $k = 1$, respectively. This leads to the following central result:

Theorem 3.2. *Consider a diffusion process defined by $\alpha = t$ and $\sigma = 1 - t$. Let the prediction target be parameterized by $\varphi = k$ and $\psi = -(1 - k)$, where $k \in [0, 1]$ is independent of t. For the objective defined in Eq. (4), assuming $t \sim \mathcal{U}[0, 1]$ and a loss weighting $\kappa = 1$ (i.e., $\boldsymbol{w} = \boldsymbol{u}$), the optimal loss in Eq. (12) is minimized when:*

$$k^* = \frac{D}{D + d} \tag{14}$$

Theorem 3.2 provides an explicit analytical relationship between the optimal prediction target and the dimensions d and D. Specifically, Eq. (14) reveals three distinct regimes:

- **Low-dimensional / Dense data** ($D \approx d$): Here, $k^* \approx 0.5$, confirming that $\boldsymbol{v}$-prediction is optimal when the data fills the ambient space.
- **High-dimensional / Sparse data** ($D \gg d$): In this regime, where data resides on a low-dimensional manifold, $k^* \to 1$, establishing $\boldsymbol{x}$-prediction as the superior choice.
- **Intermediate regimes**: For general data distributions, neither conventional target is optimal; instead, the ideal k^* lies between 0.5 and 1.

Our analysis provides, to our knowledge, the first quantitative derivation of the optimal k relative to data geometry. Furthermore, it demonstrates that for the vast majority of real-world datasets, the optimal prediction target lies in a continuous space between the discrete points of $\boldsymbol{\varepsilon}$-, $\boldsymbol{v}$-, and $\boldsymbol{x}$-prediction.

3.4 Efficacy of the Closed-Form Prediction

Theorem 3.2 predicts $k^* = D/(D + d)$ for a time-shared linear denoiser with uniform time sampling and $\boldsymbol{u}$-MSE loss. To test its efficacy and scope, we jointly learn $k = \mathrm{sigmoid}(w_k)$ and the denoiser on isotropic Gaussian data in d-dimensional subspaces of $\mathbb{R}^D$ ($D = 128$ with eleven verified d values). For each d, we reuse the dataset and learn one k shared across t.

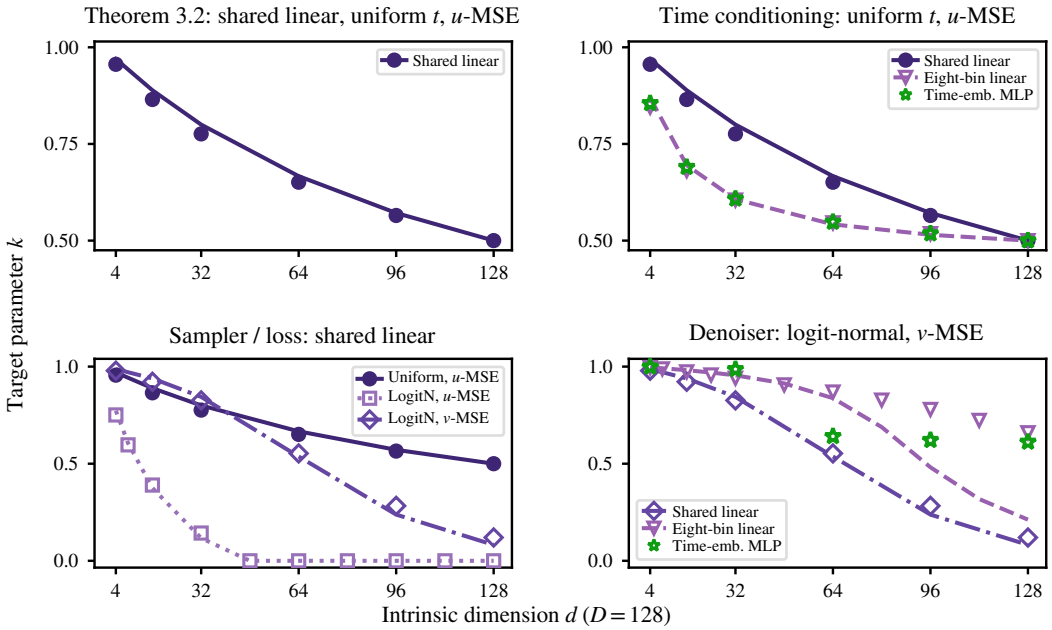


Figure 1: Controlled dimension sweep ($D = 128$) across model architectures, time samplers, and loss functions. Curves represent analytical or numerical linear-model optima; markers denote empirically learned k values (MLP shown with markers only). **Top left**: Exact Theorem 3.2 baseline (shared linear model, uniform sampling, $\boldsymbol{u}$-MSE). **Top right**: Impact of time conditioning under the baseline sampler and loss. **Bottom left**: Effect of sampler and loss variations on the shared linear denoiser. **Bottom right**: Combined effect of logit-normal sampling and $\boldsymbol{v}$-MSE across architectures. Vertical axes on the top row are magnified for visual clarity.

As illustrated in Fig. 1, our empirical findings validate these theoretical predictions while systematically demonstrating how capacity, sampling, and loss functions shape the learned target. Under the

exact assumptions of Theorem 3.2 (top left), the learned k closely tracks the closed-form prediction $D/(D+d)$ within 0.027 across the dimension sweep. Keeping uniform sampling and $\boldsymbol{u}$-MSE, adding time conditioning (top right) shifts the learned k away from $D/(D+d)$. We use independent linear maps in eight equal-width time bins and minimize their summed population loss over one k shared across bins. This binwise prediction matches the learned eight-bin k within 0.025, while the time-embedded MLP matches the learned eight-bin model within 0.012 (mean difference 0.004). Thus time conditioning explains the departure from the time-shared formula in this setting, and the binwise analysis remains predictive. An idealized model with an independent linear map at each noise level eliminates the perpendicular residual and attains lower minimum loss than the time-shared linear model for the same fixed k (Appendix H).

Modifying the objective while retaining the shared-linear denoiser (bottom left) reveals further theoretical agreement: switching to logit-normal sampling drives the optimum to $k = 0$ for $d \geq 48$ (predicted within 0.026 by Theorem 3.1), whereas additionally adopting $\boldsymbol{v}$-MSE pulls the optimum away from the boundary at high d (predicted within 0.045). Thus changing the sampler and loss moves the optimum, but the general time-shared linear analysis remains predictive. Finally, combining time conditioning with logit-normal sampling and $\boldsymbol{v}$-MSE (bottom right), the binwise population optimum no longer quantitatively predicts its trained model, especially for large values of d. For example, at $d = 128$, the eight-bin population optimum is 0.212, whereas the trained eight-bin model learns 0.659, and the time-embedded MLP learns 0.612. These results show that knowing d and D alone does not prescribe the k learned across denoisers and training objectives. This motivates learning k under the actual training setup.

4 Learnable Diffusion Prediction Target

Theorem 3.2 yields $k^* = D/(D+d)$ for a time-shared linear denoiser under uniform sampling and $\boldsymbol{u}$-MSE. However, estimating the intrinsic dimension of high-dimensional, non-linear manifolds is often intractable in practice. Meanwhile, Section 3.4 shows that time conditioning, sampling, and loss choice also alter the theoretical prediction. We therefore optimize a scalar $k = \mathrm{sigmoid}(w_k)$ jointly with the denoiser instead of assigning k from the closed form. The parameters θ and w_k are updated through the following training objective:

$$\mathcal{L}(\theta, w_k) = \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha \mathbb{E}\left[\|\boldsymbol{u}_\theta(\boldsymbol{z}) - \boldsymbol{u}\|^2\right] \tag{15}$$

Note that here $\boldsymbol{u}$ depends on w_k via k as in Eq. 13. This end-to-end, data-driven formulation enables the model to automatically discover the optimal balance between noise elimination and data recovery.

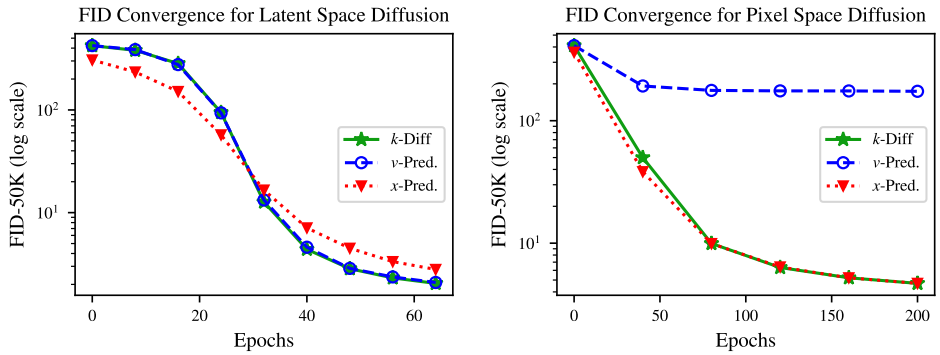
Beyond the simplified case of a constant k (time-independent), we also explore a more flexible time-dependent formulation, $k(t)$. For this purpose, we employ a piecewise linear approximation: the interval $[0, 1]$ is partitioned into N bins (e.g., $N = 128$), where each bin endpoint t_i is associated with a learnable parameter $w_k(t_i)$. For any $t \in [t_i, t_{i+1}]$, the value of $k(t)$ is obtained via linear interpolation of the sigmoid-transformed endpoints (see Appendix F for more details).

We term our method ***k*-Diff** and evaluate its performance in both latent-space and pixel-space diffusion regimes. By adaptively determining the optimal prediction target, k-Diff eliminates the need for manual hyper-parameter tuning across different data domains. Furthermore, our approach is orthogonal to existing diffusion enhancements—such as feature alignment, advanced sampling schemes, or improved autoencoders—and can be seamlessly integrated into existing pipelines.

The pseudo-code for the training and sampling procedures of k-Diff is provided in Alg. 1 and Alg. 2 in Appendix E. For numerical stability, and following the practice in [8], we clamp the denominator in the velocity calculation, $k(1-t) + (1-k)t$, to a minimum value of 0.05 to prevent division-by-zero errors when necessary (see Table 3 for more details).

5 Experiments

In this section, we provide empirical results to verify the effectiveness of our proposed method of k-Diff to learn the optimal prediction. Our experiments focus on ImageNet of resolutions 256×256 and 512×512. For ImageNet [20] 256×256, we verify with LightningDiT-XL/1 from [21] for latent



(a) Latent space diffusion.

(b) Pixel space diffusion.

Figure 2: **FID-50k convergence dynamics of k-Diff in latent and pixel spaces.** (a) Performance comparison in latent space using a LightningDiT-XL/1 architecture on ImageNet-256 over 64 epochs. k-Diff maintains a convergence profile nearly identical to the original Flow Matching ($\boldsymbol{v}$-prediction) framework. (b) Convergence comparison in pixel space using a JiT-B/16 architecture over 200 epochs. While $\boldsymbol{x}$-prediction exhibits a slightly faster initial descent, k-Diff recovers this gap as the learnable target stabilizes, indicating that the dynamic optimization of k does not compromise training efficiency or final generative performance across disparate architectures.

space generation and JiT-B/16 from [8] for pixel space generation. For ImageNet 512×512, we only verify with JiT-B/32 from [8] for pixel space generation.

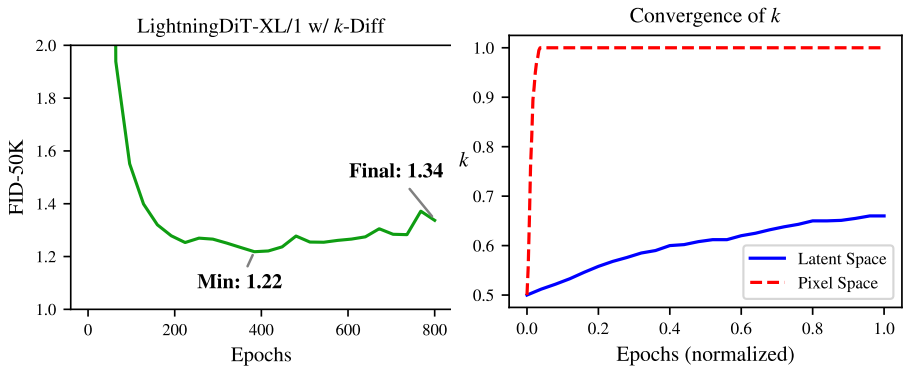
5.1 Latent Space Diffusion

We first verify our method within the latent space diffusion regime, primarily adopting the architecture and training protocol established by [21]. Specifically, we train the model for 64 epochs and perform sampling using a Classifier-Free Guidance (CFG) [22] scale of 1.5 with the Heun solver [23] for 50 sampling steps (NFE=99). While these hyperparameters deviate slightly from the original report [21]—which utilized CFG=10.0 with an Euler solver and 250 steps—we justify this configuration by noting that the baseline Flow Matching model ($\boldsymbol{v}$-prediction) yields an FID of 2.08 under our settings, which is highly consistent with the reported value of 2.11. In comparison, our model achieves a superior FID of 2.05 after the same 64 epochs of training. This is further illustrated in Fig. 2a, where the k-Diff method exhibits a convergence profile highly similar to the original $\boldsymbol{v}$-prediction framework, as measured by the FID-50k metric across the training process.

To assess the long-term convergence of our method, we extended the training process to 800 epochs, ultimately reaching a final FID of 1.34. As illustrated in the convergence plot of FID-50k in Fig. 3a, we observe a noticeable overfitting trend in the latter stages of training; while the final model remains highly performant, the peak generative quality is reached around epoch 384 with a minimum FID of 1.22. Despite this intermediate peak, we maintain a consistent reporting standard: all results tabulated in Table 1 and the visual samples presented in Fig. 4 are produced by the final 800-epoch checkpoint to provide a clear view of the model's stability over the entire training run. Note that even with this slight degradation, the final FID of 1.34 remains comparable to that from the original work [21], demonstrating the robustness of the k-Diff framework even when trained far beyond the point of initial convergence.

It is also instructive to examine the optimization trajectory of the learnable parameter k and its final steady-state value to evaluate how the empirical optimum compares with the standard $\boldsymbol{v}$-prediction objective ($k = 0.5$). As shown in Fig. 3b, the k value monotonically increases from its initial value of 0.5 to a final converged value of 0.66. This shift indicates that while the latent space is lower-dimensional than the pixel space, it still possesses significant ambient dimensions. The dynamics within this orthogonal subspace pull the optimal k toward a higher value, suggesting that a target closer to x-prediction is more effective at resolving the data manifold in this specific latent representation.

We further investigate the impact of the time-dependence of k on generative quality. As detailed in the ablation study in Appendix F, we observe that a constant k is sufficient for achieving optimal performance, whereas a time-dependent parameterization using 128 bins fails to yield further im-



(a) Long-term FID-50k dynamics. (b) Evolution of k.

Figure 3: **Long-term FID-50k dynamics and k-evolution.** (a) Latent space training (LightningDiT-XL/1) reaches a minimum FID of 1.22 at epoch 384, slightly degrading to 1.34 by epoch 800, indicating a shift in the generalization-memorization balance. (b) Trajectories of k reveal distinct optimization dynamics: in pixel space (JiT-B/16), k quickly converges to 1.0 ($\boldsymbol{x}$-prediction), whereas in latent space, it stabilizes at 0.66. This confirms k-Diff autonomously adapts to representation space, favoring intermediate targets on latent space versus $\boldsymbol{x}$-prediction in high-dimensional pixel space.

provements. This empirical finding aligns with our theoretical analysis, suggesting that the benefits of a stable, consistent prediction objective outweigh the potential flexibility of a time-varying target.



Figure 4: **Qualitative Results.** Selected examples on ImageNet 256×256 from LightningDiT-XL/1 trained with k-Diff.

In Fig. 4, we present a selection of generated samples produced by the model trained with the k-Diff objective within the latent space. These visual results demonstrate that k-Diff effectively captures the complex, multi-modal distribution of the ImageNet-256 dataset. Notably, the samples exhibit high levels of structural integrity and fine-grained textures, ranging from the intricate patterns in biological species to the sharp geometric features of architectural structures. The quality of these images suggests that the learned target $k = 0.66$ strikes an optimal balance between denoising stability and the preservation of manifold-specific details.

5.2 Pixel Space Diffusion

In addition to the latent space experiments, we evaluate the efficacy of our method within the pixel space regime. As shown in Fig. 2b, we compare the convergence trajectory of k-Diff against the $\boldsymbol{x}$-prediction baseline utilizing the JiT framework [8] for ImageNet-256 with the model architecture JiT-B/16. Our empirical results indicate that our method achieves a final convergence profile comparable to $\boldsymbol{x}$-prediction (FID=4.70 for both cases after training 200 epochs), demonstrating that k-Diff can adaptively recover the performance of specialized targets in high-dimensional pixel environments.

This adaptive behavior is further elucidated in Fig. 3b, which tracks the evolution of the learnable parameter k during the pixel-space training. Unlike the latent space case, we observe that k rapidly ascends from its initialization at 0.5 and converges to a value near 1.0 (corresponding to x-prediction) in fewer than 30 epochs.

Table 1: **Comparison results on ImageNet 256×256 and 512×512.** We evaluate our k-Diff objective across both latent space (using LightningDiT-XL/1) and pixel space (using JiT-B/16 and JiT-B/32) configurations. FID [24] and IS [25] are computed using 50K samples. Following [8], reported parameters include the generator and tokenizer decoder required at inference time, while excluding auxiliary pre-trained modules. GFLOPs are calculated for a single forward pass—excluding the tokenizer—serving as a proxy for the computational cost per iteration during both training and inference. For multi-scale methods [26], we measure the finest resolution level.

ImgNet 256×256	**params**	**Gflops**	**FID↓**	**IS↑**		**params**	**Gflops**	**FID↓**	**IS↑**
Latent-space Diffusion					*Pixel-space Diffusion*				
DiT-XL/2 [27]	675+49M	119	2.27	278.2	ADM-G [28]	554M	1120	4.59	186.7
SiT-XL/2 [29]	675+49M	119	2.06	277.5	RIN [30]	410M	334	3.42	182.0
REPA [31], SiT-XL/2	675+49M	119	1.42	305.7	SiD2 [7], UViT/2	N/A	137	**1.73**	-
DDT-XL/2 [32]	675+49M	119	1.26	310.6	PixelFlow [26], XL/4	677M	2909	1.98	282.1
RAE [33], DiT^{DH}-XL/2	839+415M	146	**1.13**	262.6	PixNerd [34], XL/16	700M	134	2.15	297
LightningDiT-XL/1 [21]	675+49M	119	1.35	295.3	JiT-B/16 [8]	131M	25	3.66	275.1
k-Diff (LightningDiT-XL/1)	675+49M	119	1.34	301.7	**k-Diff** (JiT-B/16)	131M	25	3.64	274.9

ImgNet 512×512	**params**	**Gflops**	**FID↓**	**IS↑**		**params**	**Gflops**	**FID↓**	**IS↑**
Latent-space Diffusion					*Pixel-space Diffusion*				
DiT-XL/2 [27]	675+49M	525	3.04	240.8	RIN [30]	320M	415	3.95	216.0
SiT-XL/2 [29]	675+49M	525	2.62	252.2	SiD2 [7], UViT/4	N/A	137	**2.19**	-
REPA [31], SiT-XL/2	675+49M	525	2.08	274.6	PixNerd [34], XL/16	700 M	583	2.84	245.6
DDT-XL/2 [32]	675+49M	525	1.28	305.1	JiT-B/32 [8]	133M	26	4.02	271.0
RAE [33], DiT^{DH}-XL/2	839+415M	642	**1.13**	259.6	**k-Diff** (JiT-B/32)	133M	26	4.03	277.1

This accelerated convergence suggests that the high dimensionality of the pixel space creates a strong signal for the model to prioritize data recovery over noise estimation. The fact that k-Diff autonomously identifies x-prediction as the optimal objective confirms previous hypothesis [7, 8] and the results derived in Theorem 3.2: in spaces where the ambient dimension D is much larger than the manifold dimension d, the optimal target naturally shifts toward x. By reaching this state so early in the training process, k-Diff eliminates the need for manual objective selection while ensuring the model is optimized for the correct geometric regime from the outset.

We evaluate the performance of k-Diff against established baselines across multiple resolutions and generation spaces. The results for ImageNet-256, spanning both latent and pixel-space diffusion regimes, are summarized in Table 1. Furthermore, we extend our comparative analysis to pixel-space diffusion at the higher resolution of ImageNet-512. Across these benchmarks, our method consistently matches or outperforms existing fixed-target approaches, validating its effectiveness as a general-purpose objective for high-fidelity generative modeling.

6 Conclusion and Limitation

In this work, we addressed how to determine the optimal prediction target for diffusion and flow matching processes. By analyzing the learning dynamics of a linear diffusion model, we provided a rigorous theoretical framework that identifies the **intrinsic and ambient dimensions** of data as the governing factors of training efficiency. Our analytical relationship, $k^* = D/(D + d)$, justifies the empirical shift toward $\boldsymbol{x}$-prediction while revealing why conventional targets are often suboptimal for real-world manifolds. Furthermore, our controlled sweeps demonstrate that time conditioning, samplers, and loss functions jointly shape the effective optimal target.

To bridge theory and practice, we introduced k-Diff, a framework that learns the optimal target through standard backpropagation. By introducing **only a single learnable scalar**, k-Diff bypasses explicit dimension estimation. Experiments show k-Diff matches or exceeds manually tuned baselines, offering a robust, model-agnostic, and theoretically grounded path to optimize generative processes with negligible overhead.

Limitations. The closed-form result $k^* = D/(D + d)$ assumes a whitened subspace distribution, a time-shared linear denoiser, uniform time sampling, and $\boldsymbol{u}$-MSE. Our controlled experiments show that the learned target also changes with the denoiser, sampler, and loss, and that the binwise population optimum need not match the trained value. Extending this analysis to nonlinear models and larger-scale text-to-image systems, and clarifying how target choice affects sample quality, remain open questions.

Acknowledgement

QJ gratefully acknowledges GPUHub for the generous gift and the partial support of computing resources used in this study.

References

[1] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

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

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[4] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[5] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

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

[7] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024.

[8] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

[9] Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025.

[10] Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.

[11] Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

[12] Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. *arXiv preprint arXiv:2505.17638*, 2025.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[14] Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

[15] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[16] Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

[17] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36:65484–65516, 2023.

[18] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[19] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[21] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.

[22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshops*, 2021.

[23] Karl Heun. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. *Z. Math. Phys*, 45:23–38, 1900.

[24] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[25] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[26] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow. *arXiv preprint arXiv:2504.07963*, 2025.

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[28] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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

[30] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

[31] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[32] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. Ddt: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025.

[33] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[34] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. *arXiv preprint arXiv:2507.23268*, 2025.

[35] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv:1706.02677*, 2017.

[36] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[37] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

A Definition of the Scaling Factor κ

Here we provide more detailed description of the scaling factor κ defined in Eq. 3. Our analysis is aligned with the derivations presented in [17]. Generally, for the diffusion process we have $\boldsymbol{z}$ and $\boldsymbol{u}$ defined in Eq. 1, and the estimated value from the neural network $\hat{\boldsymbol{u}}$. Now, we could solve for the estimated $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{n}}$ from the requirement

$$\boldsymbol{z} = \alpha\hat{\boldsymbol{x}} + \sigma\hat{\boldsymbol{n}} \tag{16a}$$

$$\hat{\boldsymbol{u}} = \varphi\hat{\boldsymbol{x}} + \psi\hat{\boldsymbol{n}} \tag{16b}$$

which is similar in form to the equation of the original diffusion process given by Eq. 1. Indeed, from Eq. 1 and Eq. 16, we have

$$\boldsymbol{x} = \frac{-\psi\boldsymbol{z} + \sigma\boldsymbol{u}}{\varphi\sigma - \psi\alpha} \tag{17a}$$

$$\boldsymbol{n} = \frac{\varphi\boldsymbol{z} - \alpha\boldsymbol{u}}{\varphi\sigma - \psi\alpha} \tag{17b}$$

and

$$\hat{\boldsymbol{x}} = \frac{-\psi\boldsymbol{z} + \sigma\hat{\boldsymbol{u}}}{\varphi\sigma - \psi\alpha} \tag{18a}$$

$$\hat{\boldsymbol{n}} = \frac{\varphi\boldsymbol{z} - \alpha\hat{\boldsymbol{u}}}{\varphi\sigma - \psi\alpha} \tag{18b}$$

respectively.

Now, for any target variable for MSE minimization, which is a linear combination of $\boldsymbol{x}$ and $\boldsymbol{n}$ and defined as

$$\boldsymbol{w} = \xi\boldsymbol{x} + \eta\boldsymbol{n} \tag{19}$$

where ξ and η are some parameters and can be functions of α, we have its estimated value defined as

$$\hat{\boldsymbol{w}} = \xi\hat{\boldsymbol{x}} + \eta\hat{\boldsymbol{n}} \tag{20}$$

So the loss defined for $\boldsymbol{w}$ and that for $\boldsymbol{u}$ is related by

$$\hat{\boldsymbol{w}} - \boldsymbol{w} = \xi\left(\hat{\boldsymbol{x}} - \boldsymbol{x}\right) + \eta\left(\hat{\boldsymbol{n}} - \boldsymbol{n}\right) \tag{21a}$$

$$= \xi\left(\frac{-\psi\boldsymbol{z} + \sigma\hat{\boldsymbol{u}}}{\varphi\sigma - \psi\alpha} - \frac{-\psi\boldsymbol{z} + \sigma\boldsymbol{u}}{\varphi\sigma - \psi\alpha}\right) + \eta\left(\frac{\varphi\boldsymbol{z} - \alpha\hat{\boldsymbol{u}}}{\varphi\sigma - \psi\alpha} - \frac{\varphi\boldsymbol{z} - \alpha\boldsymbol{u}}{\varphi\sigma - \psi\alpha}\right) \tag{21b}$$

$$= \xi\frac{\sigma}{\varphi\sigma - \psi\alpha}\left(\hat{\boldsymbol{u}} - \boldsymbol{u}\right) + \eta\frac{-\alpha}{\varphi\sigma - \psi\alpha}\left(\hat{\boldsymbol{u}} - \boldsymbol{u}\right) \tag{21c}$$

$$= \frac{\xi\sigma - \eta\alpha}{\varphi\sigma - \psi\alpha}\left(\hat{\boldsymbol{u}} - \boldsymbol{u}\right) \tag{21d}$$

$$= \kappa\left(\hat{\boldsymbol{u}} - \boldsymbol{u}\right) \tag{21e}$$

which is Eq. 3. Here, we have defined

$$\kappa \coloneqq \frac{\xi\sigma - \eta\alpha}{\varphi\sigma - \psi\alpha} \tag{22}$$

Now we give some specific examples to illustrate the usage of this relationship. For example, for the original $\boldsymbol{u}$-loss where $\boldsymbol{w} = \boldsymbol{u}$, we have

$$\kappa = 1 \tag{23}$$

For $\boldsymbol{x}$-loss, we have $\boldsymbol{w} = \boldsymbol{x}, \xi = 1, \eta = 0$, and

$$\kappa = \frac{\sigma}{\varphi\sigma - \psi\alpha} \tag{24}$$

For $\boldsymbol{\varepsilon}$-loss, we have $\boldsymbol{w} = \boldsymbol{n}, \xi = 0, \eta = 1$, and

$$\kappa = -\frac{\alpha}{\varphi\sigma - \psi\alpha} \tag{25}$$

For $\boldsymbol{v}$-loss, we have $\boldsymbol{w} = \boldsymbol{x} - \boldsymbol{n}, \xi = 1, \eta = -1$, and

$$\kappa = \frac{\alpha + \sigma}{\varphi\sigma - \psi\alpha} \tag{26}$$

B Derivation of the Learning Dynamics in Eq. 9

Here we derive the learning dynamics given in Eq. 9. The learning dynamics for the weight W is generally given by the gradient descent

$$\tau \frac{\mathrm{d}W}{\mathrm{d}\widetilde{t}} = -\partial_W \mathcal{L} \tag{27}$$

Here, as before, τ represents the inverse of the learning rate, and $\widetilde{t}$ denotes the training progress of the network. From the definition of the MSE loss in Eq. 4, we have

$$\begin{aligned}
\tau \frac{\mathrm{d}W}{\mathrm{d}\widetilde{t}} &= -\partial_W \mathcal{L} && \text{(28a)}\\
&= -\partial_W \left[\frac{1}{2} \int \widetilde{\mathcal{D}} \alpha \mathbb{E} \left[\|W\boldsymbol{z} - \boldsymbol{u}\|^2 \right] \right] && \text{(28b)}\\
&= -\frac{1}{2} \int \widetilde{\mathcal{D}} \alpha \mathbb{E} \left[\partial_W \|W\boldsymbol{z} - \boldsymbol{u}\|^2 \right] && \text{(28c)}\\
&= -\frac{1}{2} \int \widetilde{\mathcal{D}} \alpha \mathbb{E} \left[2 \left(W\boldsymbol{z} - \boldsymbol{u} \right) \boldsymbol{z}^\top \right] && \text{(28d)}\\
&= -\int \widetilde{\mathcal{D}} \alpha \Big(W \mathbb{E} \left[\boldsymbol{z}\boldsymbol{z}^\top \right] - \mathbb{E} \left[\boldsymbol{u}\boldsymbol{z}^\top \right] \Big) && \text{(28e)}
\end{aligned}$$

where for simplicity we have omitted the integral interval $\mathcal{I}$ without ambiguity.

Now from the definition in Eq. 1, we have

$$\begin{aligned}
\mathbb{E} \left[\boldsymbol{z}\boldsymbol{z}^\top \right] &= \mathbb{E} \left[(\alpha\boldsymbol{x} + \sigma\boldsymbol{n})(\alpha\boldsymbol{x} + \sigma\boldsymbol{n})^\top \right] && \text{(29a)}\\
&= \alpha^2 \mathbb{E} \left[\boldsymbol{x}\boldsymbol{x}^\top \right] + \alpha\sigma \mathbb{E} \left[\boldsymbol{x}\boldsymbol{n}^\top \right] + \alpha\sigma \mathbb{E} \left[\boldsymbol{n}\boldsymbol{x}^\top \right] + \sigma^2 \mathbb{E} \left[\boldsymbol{n}\boldsymbol{n}^\top \right] && \text{(29b)}\\
\mathbb{E} \left[\boldsymbol{u}\boldsymbol{z}^\top \right] &= \mathbb{E} \left[(\varphi\boldsymbol{x} + \psi\boldsymbol{n})(\alpha\boldsymbol{x} + \sigma\boldsymbol{n})^\top \right] && \text{(29c)}\\
&= \varphi\alpha \mathbb{E} \left[\boldsymbol{x}\boldsymbol{x}^\top \right] + \varphi\sigma \mathbb{E} \left[\boldsymbol{x}\boldsymbol{n}^\top \right] + \psi\alpha \mathbb{E} \left[\boldsymbol{n}\boldsymbol{x}^\top \right] + \psi\sigma \mathbb{E} \left[\boldsymbol{n}\boldsymbol{n}^\top \right] && \text{(29d)}
\end{aligned}$$

Since $\boldsymbol{x}$ and $\boldsymbol{n}$ are independent random variables, and $\mathbb{E}[\boldsymbol{n}] = \boldsymbol{0}$, we have

$$\begin{aligned}
\mathbb{E} \left[\boldsymbol{x}\boldsymbol{n}^\top \right] &= \mathbb{E}[\boldsymbol{x}] \, \mathbb{E}[\boldsymbol{n}]^\top = 0 && \text{(30a)}\\
\mathbb{E} \left[\boldsymbol{n}\boldsymbol{x}^\top \right] &= \mathbb{E}[\boldsymbol{n}] \, \mathbb{E}[\boldsymbol{x}]^\top = 0 && \text{(30b)}
\end{aligned}$$

From the definition $\boldsymbol{x} = P\widetilde{\boldsymbol{x}}$ and the assumptions in Eq. 6, we have

$$\begin{aligned}
\mathbb{E} \left[\boldsymbol{x}\boldsymbol{x}^\top \right] &= \mathbb{E} \left[P\widetilde{\boldsymbol{x}} \left(P\widetilde{\boldsymbol{x}} \right)^\top \right] && \text{(31a)}\\
&= P \mathbb{E} \left[\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^\top \right] P^\top && \text{(31b)}\\
&= PP^\top && \text{(31c)}
\end{aligned}$$

Now using the white noise assumption in Eq. 7, we have

$$\begin{aligned}
\mathbb{E} \left[\boldsymbol{z}\boldsymbol{z}^\top \right] &= \alpha^2 PP^\top + \sigma^2 I && \text{(32a)}\\
\mathbb{E} \left[\boldsymbol{u}\boldsymbol{z}^\top \right] &= \varphi\alpha PP^\top + \psi\sigma I && \text{(32b)}
\end{aligned}$$

The learning dynamics is thus given by

$$\begin{aligned}
\tau \frac{\mathrm{d}W}{\mathrm{d}\widetilde{t}} &= -\int \widetilde{\mathcal{D}} \alpha \Big(W \mathbb{E} \left[\boldsymbol{z}\boldsymbol{z}^\top \right] - \mathbb{E} \left[\boldsymbol{u}\boldsymbol{z}^\top \right] \Big) && \text{(33a)}\\
&= -\int \widetilde{\mathcal{D}} \alpha \Big(W \left(\alpha^2 PP^\top + \sigma^2 I \right) - \left(\varphi\alpha PP^\top + \psi\sigma I \right) \Big) && \text{(33b)}\\
&= -\int \widetilde{\mathcal{D}} \alpha \left(\alpha^2 WPP^\top + \sigma^2 W - \varphi\alpha PP^\top - \psi\sigma I \right) && \text{(33c)}
\end{aligned}$$

which is exactly Eq. 8.

Now we multiply both sides of the equation on the right by $PP^\top$ and $I - PP^\top$, respectively. Notice that $PP^\top$ is a projection operation, and

$$(PP^\top)(PP^\top) = PP^\top \tag{34a}$$
$$(I - PP^\top)(I - PP^\top) = I - PP^\top \tag{34b}$$
$$PP^\top(I - PP^\top) = (I - PP^\top)PP^\top = 0 \tag{34c}$$

we have

$$\tau \frac{\mathrm{d}(WPP^\top)}{\mathrm{d}\widetilde{t}} = \tau \frac{\mathrm{d}W}{\mathrm{d}\widetilde{t}} PP^\top \tag{35a}$$
$$= -\int \widetilde{\mathcal{D}}\alpha \left(\alpha^2 WPP^\top + \sigma^2 W - \varphi\alpha PP^\top - \psi\sigma I\right) PP^\top \tag{35b}$$
$$= -\int \widetilde{\mathcal{D}}\alpha \left(\alpha^2 WPP^\top PP^\top + \sigma^2 WPP^\top - \varphi\alpha PP^\top PP^\top - \psi\sigma PP^\top\right) \tag{35c}$$
$$= -\int \widetilde{\mathcal{D}}\alpha \left(\alpha^2 WPP^\top + \sigma^2 WPP^\top - \varphi\alpha PP^\top - \psi\sigma PP^\top\right) \tag{35d}$$
$$= -\int \widetilde{\mathcal{D}}\alpha \left((\alpha^2 + \sigma^2)WPP^\top - (\varphi\alpha + \psi\sigma)PP^\top\right) \tag{35e}$$

and

$$\tau \frac{\mathrm{d}(W(I - PP^\top))}{\mathrm{d}\widetilde{t}} = \tau \frac{\mathrm{d}W}{\mathrm{d}\widetilde{t}} (I - PP^\top) \tag{36a}$$
$$= -\int \widetilde{\mathcal{D}}\alpha \left(\alpha^2 WPP^\top + \sigma^2 W - \varphi\alpha PP^\top - \psi\sigma I\right) (I - PP^\top) \tag{36b}$$
$$= -\int \widetilde{\mathcal{D}}\alpha \big(\alpha^2 WPP^\top (I - PP^\top) + \sigma^2 W(I - PP^\top) - \varphi\alpha PP^\top (I - PP^\top) - \psi\sigma(I - PP^\top)\big) \tag{36c}$$
$$= -\int \widetilde{\mathcal{D}}\alpha \left(\sigma^2 W(I - PP^\top) - \psi\sigma(I - PP^\top)\right) \tag{36d}$$

which are exactly Eq. 9 by the definition in Eq. 10.

C Derivation of the Optimal Loss Δ^* in Theorem 3.1

Now the optimal value of the two modes $W_\parallel$ and $W_\perp$ are given by the condition that the time derivatives in Eq. 9a and Eq. 9b vanish, respectively. Notice that W is not dependent on α and so are $W_\parallel$ and $W_\perp$, so the equilibrium conditions given by

$$\int \widetilde{\mathcal{D}}\alpha \Big((\alpha^2 + \sigma^2)W_\parallel^* - (\varphi\alpha + \psi\sigma)PP^\top\Big) = 0 \tag{37a}$$
$$\int \widetilde{\mathcal{D}}\alpha \Big(\sigma^2 W_\perp^* - \psi\sigma(I - PP^\top)\Big) = 0 \tag{37b}$$

are equivalent to

$$W_\parallel^* \int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2) - PP^\top \int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma) = 0 \tag{38a}$$
$$W_\perp^* \int \widetilde{\mathcal{D}}\alpha\sigma^2 - (I - PP^\top) \int \widetilde{\mathcal{D}}\alpha\psi\sigma = 0 \tag{38b}$$

which immediately gives Eq. 11.

The optimal loss is given by substituting Eq. 11 into Eq. 4, which is

$$\Delta^* = \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha \mathbb{E}\left[\|W^*\boldsymbol{z} - \boldsymbol{u}\|^2\right] \tag{39a}$$

$$= \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha \mathbb{E}\left[\mathrm{Tr}\left((W^*\boldsymbol{z} - \boldsymbol{u})(W^*\boldsymbol{z} - \boldsymbol{u})^\top\right)\right] \tag{39b}$$

$$= \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha \,\mathrm{Tr}\left(\mathbb{E}\left[(W^*\boldsymbol{z} - \boldsymbol{u})(W^*\boldsymbol{z} - \boldsymbol{u})^\top\right]\right) \tag{39c}$$

$$= \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha \,\mathrm{Tr}\left(\mathbb{E}\left[W^*\boldsymbol{z}\boldsymbol{z}^\top {W^*}^\top - \boldsymbol{u}\boldsymbol{z}^\top {W^*}^\top - W^*\boldsymbol{z}\boldsymbol{u}^\top + \boldsymbol{u}\boldsymbol{u}^\top\right]\right) \tag{39d}$$

$$= \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha \,\mathrm{Tr}\left(W^*\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^\top\right]{W^*}^\top - \mathbb{E}\left[\boldsymbol{u}\boldsymbol{z}^\top\right]{W^*}^\top - W^*\mathbb{E}\left[\boldsymbol{z}\boldsymbol{u}^\top\right] + \mathbb{E}\left[\boldsymbol{u}\boldsymbol{u}^\top\right]\right) \tag{39e}$$

Now as in Eq. 32, we have

$$\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^\top\right] = \alpha^2 PP^\top + \sigma^2 I \tag{40a}$$

$$\mathbb{E}\left[\boldsymbol{u}\boldsymbol{z}^\top\right] = \varphi\alpha PP^\top + \psi\sigma I \tag{40b}$$

$$\mathbb{E}\left[\boldsymbol{z}\boldsymbol{u}^\top\right] = \mathbb{E}\left[\boldsymbol{u}\boldsymbol{z}^\top\right]^\top \tag{40c}$$

$$= \varphi\alpha PP^\top + \psi\sigma I \tag{40d}$$

$$\mathbb{E}\left[\boldsymbol{u}\boldsymbol{u}^\top\right] = \mathbb{E}\left[(\varphi\boldsymbol{x} + \psi\boldsymbol{n})(\varphi\boldsymbol{x} + \psi\boldsymbol{n})^\top\right] \tag{40e}$$

$$= \varphi^2\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\top\right] + \varphi\psi\mathbb{E}\left[\boldsymbol{x}\boldsymbol{n}^\top\right] + \varphi\psi\mathbb{E}\left[\boldsymbol{n}\boldsymbol{x}^\top\right] + \psi^2\mathbb{E}\left[\boldsymbol{n}\boldsymbol{n}^\top\right] \tag{40f}$$

$$= \varphi^2 PP^\top + \psi^2 I \tag{40g}$$

we have

$$\begin{aligned}\Delta^* =& \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha \,\mathrm{Tr}\Big(W^*(\alpha^2 PP^\top + \sigma^2 I){W^*}^\top - (\varphi\alpha PP^\top + \psi\sigma I){W^*}^\top \\ & - W^*(\varphi\alpha PP^\top + \psi\sigma I) + (\varphi^2 PP^\top + \psi^2 I)\Big)\end{aligned} \tag{41}$$

From Eq. 11, and notice that $W_\parallel^*$ and $W_\perp^*$ are directly proportional to $PP^\top$ and $I - PP^\top$, respectively, we have

$$W_\parallel^* PP^\top = PP^\top W_\parallel^* = W_\parallel^* \tag{42a}$$

$$W_\perp^*(I - PP^\top) = (I - PP^\top)W_\perp^* = W_\perp^* \tag{42b}$$

$$W_\parallel^*(I - PP^\top) = (I - PP^\top)W_\parallel^* = 0 \tag{42c}$$

$$W_\perp^* PP^\top = PP^\top W_\perp^* = 0 \tag{42d}$$

and thus

$$\mathrm{Tr}\left(W^* PP^\top {W^*}^\top\right) = \mathrm{Tr}\left((W_\parallel^* + W_\perp^*)PP^\top(W_\parallel^* + W_\perp^*)\right) \tag{43a}$$

$$= \mathrm{Tr}\left(W_\parallel^* PP^\top {W_\parallel^*}^\top\right) \tag{43b}$$

$$= \mathrm{Tr}\left(W_\parallel^* {W_\parallel^*}^\top\right) \tag{43c}$$

$$= \mathrm{Tr}\left(\frac{\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)} PP^\top \frac{\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)} PP^\top\right) \tag{43d}$$

$$= \frac{\left(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)\right)^2}{\left(\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)\right)^2}\,\mathrm{Tr}(PP^\top) \tag{43e}$$

$$\operatorname{Tr}\left(W^*(I-PP^\top){W^*}^\top\right) = \operatorname{Tr}\left((W_\parallel^*+W_\perp^*)(I-PP^\top)(W_\parallel^*+W_\perp^*)\right) \tag{44a}$$
$$= \operatorname{Tr}\left(W_\perp^*(I-PP^\top){W_\perp^*}^\top\right) \tag{44b}$$
$$= \operatorname{Tr}\left(W_\perp^*{W_\perp^*}^\top\right) \tag{44c}$$
$$= \operatorname{Tr}\left(\frac{\int\widetilde{\mathcal{D}}\alpha\psi\sigma}{\int\widetilde{\mathcal{D}}\alpha\sigma^2}(I-PP^\top)\frac{\int\widetilde{\mathcal{D}}\alpha\psi\sigma}{\int\widetilde{\mathcal{D}}\alpha\sigma^2}(I-PP^\top)\right) \tag{44d}$$
$$= \frac{\left(\int\widetilde{\mathcal{D}}\alpha\psi\sigma\right)^2}{\left(\int\widetilde{\mathcal{D}}\alpha\sigma^2\right)^2}\operatorname{Tr}(I-PP^\top) \tag{44e}$$

$$\operatorname{Tr}\left(W^*PP^\top\right) = \operatorname{Tr}\left(PP^\top{W^*}^\top\right) \tag{45a}$$
$$= \operatorname{Tr}\left(W_\parallel^*\right) \tag{45b}$$
$$= \frac{\int\widetilde{\mathcal{D}}\alpha(\varphi\alpha+\psi\sigma)}{\int\widetilde{\mathcal{D}}\alpha(\alpha^2+\sigma^2)}\operatorname{Tr}\left(PP^\top\right) \tag{45c}$$

and

$$\operatorname{Tr}\left(W^*(I-PP^\top)\right) = \operatorname{Tr}\left((I-PP^\top){W^*}^\top\right) \tag{46a}$$
$$= \operatorname{Tr}\left(W_\perp^*\right) \tag{46b}$$
$$= \frac{\int\widetilde{\mathcal{D}}\alpha\psi\sigma}{\int\widetilde{\mathcal{D}}\alpha\sigma^2}\operatorname{Tr}\left(I-PP^\top\right) \tag{46c}$$

With these results, we have

$$\begin{aligned}\Delta^* =& \frac{1}{2}\int\widetilde{\mathcal{D}}\alpha\operatorname{Tr}\Big(W^*(\alpha^2PP^\top+\sigma^2I){W^*}^\top-(\varphi\alpha PP^\top+\psi\sigma I){W^*}^\top\\ &-W^*(\varphi\alpha PP^\top+\psi\sigma I)+(\varphi^2PP^\top+\psi^2I)\Big)\end{aligned} \tag{47a}$$
$$\begin{aligned}=& \frac{1}{2}\int\widetilde{\mathcal{D}}\alpha\operatorname{Tr}\Big(W^*(\alpha^2PP^\top+\sigma^2I){W^*}^\top-2W^*(\varphi\alpha PP^\top+\psi\sigma I)\\ &+(\varphi^2PP^\top+\psi^2I)\Big)\end{aligned} \tag{47b}$$
$$\begin{aligned}=& \frac{1}{2}\int\widetilde{\mathcal{D}}\alpha\operatorname{Tr}\Big(W^*\big((\alpha^2+\sigma^2)PP^\top+\sigma^2(I-PP^\top)\big){W^*}^\top\\ &-2W^*\big((\varphi\alpha+\psi\sigma)PP^\top+\psi\sigma(I-PP^\top)\big)\\ &+\big((\varphi^2+\psi^2)PP^\top+\psi^2(I-PP^\top)\big)\Big)\end{aligned} \tag{47c}$$
$$\begin{aligned}=& \frac{1}{2}\int\widetilde{\mathcal{D}}\alpha\Big((\alpha^2+\sigma^2)\operatorname{Tr}(W^*PP^\top{W^*}^\top)+\sigma^2\operatorname{Tr}\left(W^*(I-PP^\top){W^*}^\top\right)\\ &-2(\varphi\alpha+\psi\sigma)\operatorname{Tr}(W^*PP^\top)\\ &-2\psi\sigma\operatorname{Tr}\left(W^*(I-PP^\top)\right)+(\varphi^2+\psi^2)\operatorname{Tr}(PP^\top)+\psi^2\operatorname{Tr}(I-PP^\top)\Big)\end{aligned} \tag{47d}$$
$$\begin{aligned}=& \frac{1}{2}\int\widetilde{\mathcal{D}}\alpha\Big((\alpha^2+\sigma^2)\frac{\left(\int\widetilde{\mathcal{D}}\bar{\alpha}(\bar{\varphi}\bar{\alpha}+\bar{\psi}\bar{\sigma})\right)^2}{\left(\int\widetilde{\mathcal{D}}\bar{\alpha}(\bar{\alpha}^2+\bar{\sigma}^2)\right)^2}\operatorname{Tr}(PP^\top)+\sigma^2\frac{\left(\int\widetilde{\mathcal{D}}\bar{\alpha}\bar{\psi}\bar{\sigma}\right)^2}{\left(\int\widetilde{\mathcal{D}}\bar{\alpha}\bar{\sigma}^2\right)^2}\operatorname{Tr}(I-PP^\top)\\ &-2(\varphi\alpha+\psi\sigma)\frac{\int\widetilde{\mathcal{D}}\bar{\alpha}(\bar{\varphi}\bar{\alpha}+\bar{\psi}\bar{\sigma})}{\int\widetilde{\mathcal{D}}\bar{\alpha}(\bar{\alpha}^2+\bar{\sigma}^2)}\operatorname{Tr}\left(PP^\top\right)-2\psi\sigma\frac{\int\widetilde{\mathcal{D}}\bar{\alpha}\bar{\psi}\bar{\sigma}}{\int\widetilde{\mathcal{D}}\bar{\alpha}\bar{\sigma}^2}\operatorname{Tr}\left(I-PP^\top\right)\end{aligned}$$

$$+ (\varphi^2 + \psi^2) \operatorname{Tr}(PP^\top) + \psi^2 \operatorname{Tr}(I - PP^\top)\Big) \tag{47e}$$

$$= \frac{1}{2} \frac{\left(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)\right)^2}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)} \operatorname{Tr}(PP^\top) + \frac{1}{2} \frac{\left(\int \widetilde{\mathcal{D}}\alpha\psi\sigma\right)^2}{\int \widetilde{\mathcal{D}}\alpha\sigma^2} \operatorname{Tr}(I - PP^\top)$$
$$- \frac{1}{2} \cdot \frac{2\left(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)\right)^2}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)} \operatorname{Tr}(PP^\top) - \frac{1}{2} \cdot \frac{2\left(\int \widetilde{\mathcal{D}}\alpha\psi\sigma\right)^2}{\int \widetilde{\mathcal{D}}\alpha\sigma^2} \operatorname{Tr}(I - PP^\top)$$
$$+ \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha(\varphi^2 + \psi^2) \operatorname{Tr}(PP^\top) + \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha\psi^2 \operatorname{Tr}(I - PP^\top) \tag{47f}$$

$$= \frac{1}{2} \left(\int \widetilde{\mathcal{D}}\alpha(\varphi^2 + \psi^2) - \frac{\left(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)\right)^2}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)} \right) \operatorname{Tr}(PP^\top)$$
$$+ \frac{1}{2} \left(\int \widetilde{\mathcal{D}}\alpha\psi^2 - \frac{\left(\int \widetilde{\mathcal{D}}\alpha\psi\sigma\right)^2}{\int \widetilde{\mathcal{D}}\alpha\sigma^2} \right) \operatorname{Tr}(I - PP^\top) \tag{47g}$$

where we have introduced dummy variables $\bar{\alpha}, \bar{\sigma}, \bar{\varphi}, \bar{\psi}$ to avoid symbolic ambiguity.

Notice that

$$\operatorname{Tr}(PP^\top) = \operatorname{Tr}(P^\top P) \tag{48a}$$
$$= \operatorname{Tr}(I_{d\times d}) \tag{48b}$$
$$= d \tag{48c}$$

and

$$\operatorname{Tr}(I - PP^\top) = \operatorname{Tr}(I) - \operatorname{Tr}(P^\top P) \tag{49a}$$
$$= D - d \tag{49b}$$

we have

$$\Delta^* = \frac{1}{2} d \left(\int \widetilde{\mathcal{D}}\alpha(\varphi^2 + \psi^2) - \frac{\left(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha + \psi\sigma)\right)^2}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2 + \sigma^2)} \right)$$
$$+ \frac{1}{2}(D - d) \left(\int \widetilde{\mathcal{D}}\alpha\psi^2 - \frac{\left(\int \widetilde{\mathcal{D}}\alpha\psi\sigma\right)^2}{\int \widetilde{\mathcal{D}}\alpha\sigma^2} \right) \tag{50}$$

which is Eq. 12.

D Derivation of the Optimal Value of k in Theorem 3.2

Now for the simplified case where $\alpha = t, \sigma = 1 - t, \varphi = k, \psi = -(1 - k)$ with $0 \leq k \leq 1$ independent of t, we have the optimal loss in Eq. 12 simplified as

$$\Delta^* = \frac{1}{2} d \left(\left(k^2 + (1-k)^2\right) \int \widetilde{\mathcal{D}}\alpha - \frac{\left(k \int \widetilde{\mathcal{D}}\alpha\alpha - (1-k) \int \widetilde{\mathcal{D}}\alpha(1-\alpha)\right)^2}{\int \widetilde{\mathcal{D}}\alpha \left(\alpha^2 + (1-\alpha)^2\right)} \right)$$
$$+ \frac{1}{2}(D - d) \left((1-k)^2 \int \widetilde{\mathcal{D}}\alpha - \frac{(1-k)^2 \left(\int \widetilde{\mathcal{D}}\alpha(1-\alpha)\right)^2}{\int \widetilde{\mathcal{D}}\alpha(1-\alpha)^2} \right) \tag{51a}$$

where for uniform sampling of α on the interval $[0, 1]$

$$\int \widetilde{\mathcal{D}}\alpha = \int_0^1 \mathrm{d}\alpha \cdot \kappa^2 \tag{52}$$

If we use the original MSE loss of $\boldsymbol{u}$, we have $\kappa = 1$, and

$$\int \widetilde{\mathcal{D}}\alpha = \int_0^1 \mathrm{d}\alpha = 1 \tag{53a}$$

$$\int \widetilde{\mathcal{D}}\alpha\alpha = \int_0^1 \mathrm{d}\alpha\alpha = \frac{1}{2} \tag{53b}$$

$$\int \widetilde{\mathcal{D}}\alpha(1-\alpha) = \int_0^1 \mathrm{d}\alpha(1-\alpha) = \frac{1}{2} \tag{53c}$$

$$\int \widetilde{\mathcal{D}}\alpha\alpha^2 = \int_0^1 \mathrm{d}\alpha\alpha^2 = \frac{1}{3} \tag{53d}$$

$$\int \widetilde{\mathcal{D}}\alpha(1-\alpha)^2 = \int_0^1 \mathrm{d}\alpha(1-\alpha)^2 = \frac{1}{3} \tag{53e}$$

so we have

$$\Delta^* = \frac{1}{2}d\left(\left(k^2 + (1-k)^2\right) - \frac{\left(\frac{1}{2}k - \frac{1}{2}(1-k)\right)^2}{\frac{2}{3}}\right) + \frac{1}{2}(D-d)\left((1-k)^2 - \frac{(1-k)^2 \cdot \left(\frac{1}{2}\right)^2}{\frac{1}{3}}\right) \tag{54a}$$

$$= \frac{1}{2}d\left(\frac{1}{2}k^2 - \frac{1}{2}k + \frac{5}{8}\right) + \frac{1}{8}(D-d)(1-k)^2 \tag{54b}$$

$$= \frac{1}{4}d\left(k^2 - k + \frac{5}{4}\right) + \frac{1}{8}(D-d)(1-k)^2 \tag{54c}$$

$$= \frac{1}{16}\left[2(D+d)k^2 - 4Dk + (2D+3d)\right] \tag{54d}$$

which is minimized by

$$k^* = \frac{D}{D+d} \tag{55}$$

This is the condition given in Eq. 14.

E Training and Generation Algorithms

The algorithms for training and generation with the proposed k-Diff method is provided in Alg. 1 and Alg. 2, respectively.

Algorithm 1 Training step

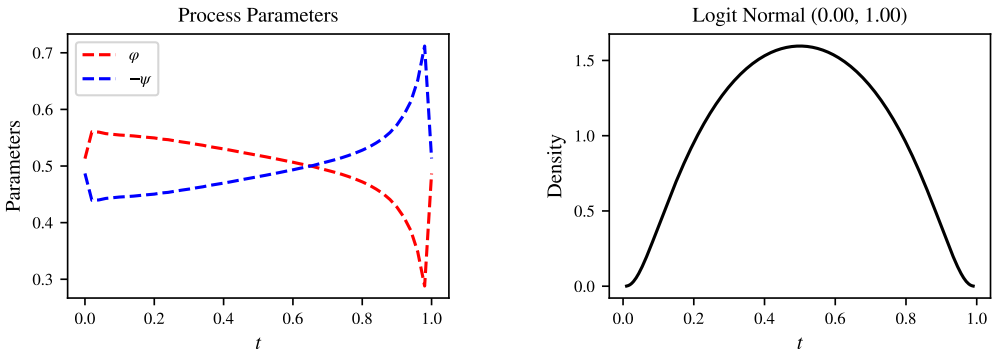
```
# net(z, t): Diffusion Model
# w_k: trainable parameter for k
# x: training batch

t = sample_t()
e = randn_like(x)
k = sigmoid(w_k)

z = t * x + (1 - t) * e
u = k * x - (1 - k) * e
v = ((1 - 2 * k) * z + u) / (k * (1 - t) +
     (1 - k) * t)

u_pred = net(z, t)
v_pred = ((1 - 2 * k) * z + u_pred) / (k *
     (1 - t) + (1 - k) * t)

loss = l2_loss(v - v_pred)
```

Algorithm 2 Sampling step (Euler)

```
# z: current samples at t

k = sigmoid(w_k)

u_pred = net(z, t)
v_pred = ((1 - 2 * k) * z + u_pred) / (k *
     (1 - t) + (1 - k) * t)

z_next = z + (t_next - t) * v_pred
```

F Ablation Study

In this section, we provide ablation study to justify our design choices regarding the parameterization of k and the impact of the time-/α-dependence of k.



(a) (b)

Figure 5: (a) **The learned time-dependent $k(t)$ for 128 time bins.** (b) **Probability distribution of the logit normal distribution with $\mu = 0.0, \sigma = 1.0$.**

Number of independent parameters. We first investigate whether decoupling the coefficients for $\boldsymbol{x}$ and $\boldsymbol{\varepsilon}$ improves performance by introducing more degrees of freedom. In this setup, the target is defined as $k_0\boldsymbol{x} - k_1\boldsymbol{\varepsilon}$, where k_0 and k_1 are two independent trainable parameters both in the range of $[0, 1]$. As shown in Table 2, this two-parameter approach achieves an FID of 2.04, which is nearly identical to the 2.05 achieved by our single-parameter k-Diff ($k\boldsymbol{x} - (1-k)\boldsymbol{\varepsilon}$). This suggests that the relationship between the data and noise components is captured sufficiently by a single interpolation constant, and the additional complexity of a second independent parameter yields no significant generative gain.

Constant vs. Time-Dependent k. We also examine whether k should remain constant across the diffusion process or vary with respect to the timestep t (or equivalently, α). We implement a time-dependent version of k-Diff using 128 bins, where $k(t)$ is determined via linear interpolation across the range of t. This configuration resulted in a degraded FID of 2.17. We hypothesize that a constant k provides a more stable optimization target, whereas a time-dependent k forces the model to adapt to a shifting prediction objective at different noise levels. This lack of target consistency may increase the difficulty of the learning task and reduce the model's robustness to estimation errors, whereas fixed-target objectives (such as $\boldsymbol{x}$, $\boldsymbol{\varepsilon}$, or $\boldsymbol{v}$ in flow matching) benefit from a consistent output distribution across all timesteps.

Table 2: **Ablation study on the parameterization and time-dependency of k.** All models were trained for 64 epochs using the LightningDiT-XL/1 architecture on ImageNet 256×256 in latent space. We compare the performance of single vs. dual independent k values and the effect of using a constant k versus a time-dependent $k(t)$ interpolated across 128 bins.

# of k	Target Type	# of Time Bins	Learned Value(s)†	FID
1 (k-Diff)	Constant	1	0.522	2.05
2 (k_0, k_1)	Constant	1	(0.556, 0.509)	2.04
1 (k-Diff)	Time-dependent	128	See Fig. 5a	2.17

† Learn value(s) are from Epoch 64.

In Fig. 5a, we plot the trained parameters $\varphi = k(t)$ and $-\psi = 1 - k(t)$ for the time-dependent configuration, observing that the converged values deviate significantly from their initial state of 0.5. Excluding the boundary regions near $t = 0$ and $t = 1$, we find that $k(t)$ monotonically decreases as t increases. Specifically, $k(t)$ originates at 0.56 for $t \approx 0$ and exhibits a gradual decline in the small-t regime, followed by a accelerated decreasing toward a value near 0.3 as t approaches 1. Crucially, $k(t)$ remains essentially stagnant at its initial value of 0.5 in the immediate vicinity of the boundaries ($t \in \{0, 1\}$). We attribute this lack of optimization at the end-points to the density of the time-sampling distribution adopted during training. As illustrated by the probability density function of the logit-normal distribution ($\mu = 0, \sigma = 1$) in Fig. 5b, the sampling density vanishes at the boundaries. Consequently, the gradient signal at these points is insufficient to drive the parameters away from their initialization.

G Generalization to Colored Data in Ambient Space

In the main text, we focused on the whitened case where the data covariance was defined by an orthogonal projection $PP^\top$. Here, we generalize the k-Diff framework to the full ambient space $\mathbb{R}^D$, where the second moment of the data is given by $\Sigma \in \mathbb{R}^{D\times D}$, i.e. $\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\top\right] = \Sigma$, which is a general symmetric positive semi-definite matrix. The case of whitened low-dimensional data analyzed in the main text is a special case where $\Sigma = PP^\top$.

In this general case, Eq. (32) is changed to

$$\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^\top\right] = \alpha^2\Sigma + \sigma^2 I \tag{56a}$$

$$\mathbb{E}\left[\boldsymbol{u}\boldsymbol{z}^\top\right] = \varphi\alpha\Sigma + \psi\sigma I \tag{56b}$$

The learning dynamics is thus given by

$$\tau\frac{\mathrm{d}W}{\mathrm{d}\widetilde{t}} = -\int \widetilde{\mathcal{D}}\alpha\left(\alpha^2 W\Sigma + \sigma^2 W - \varphi\alpha\Sigma - \psi\sigma I\right) \tag{57}$$

Now since Σ is a symmetric semi-positive matrix, it can be decomposed into orthogonal components, each corresponding to one of the eigenvalues of Σ, i.e.

$$\Sigma = \sum_i \lambda_i \boldsymbol{q}_i \boldsymbol{q}_i^\top \tag{58}$$

where

$$\boldsymbol{q}_j^\top \boldsymbol{q}_i = \delta_{ij} \tag{59}$$

and

$$\sum_i \boldsymbol{q}_i \boldsymbol{q}_i^\top = I \tag{60}$$

we have the learning dynamics for each projected mode $\boldsymbol{w}_{\boldsymbol{q}_i} \coloneqq W\boldsymbol{q}_i$ given by

$$\tau\frac{\mathrm{d}\boldsymbol{w}_{\boldsymbol{q}_i}}{\mathrm{d}\widetilde{t}} = \tau\frac{\mathrm{d}W\boldsymbol{q}_i}{\mathrm{d}\widetilde{t}} \tag{61a}$$

$$= -\int \widetilde{\mathcal{D}}\alpha\left(\alpha^2 W\Sigma\boldsymbol{q}_i + \sigma^2 W\boldsymbol{q}_i - \varphi\alpha\Sigma\boldsymbol{q}_i - \psi\sigma I\boldsymbol{q}_i\right) \tag{61b}$$

$$= -\int \widetilde{\mathcal{D}}\alpha\left(\alpha^2 W\lambda_i\boldsymbol{q}_i + \sigma^2 W\boldsymbol{q}_i - \varphi\alpha\lambda_i\boldsymbol{q}_i - \psi\sigma\boldsymbol{q}_i\right) \tag{61c}$$

$$= -\int \widetilde{\mathcal{D}}\alpha\left(\lambda_i\alpha^2\boldsymbol{w}_{\boldsymbol{q}_i} + \sigma^2\boldsymbol{w}_{\boldsymbol{q}_i} - \lambda_i\varphi\alpha\boldsymbol{q}_i - \psi\sigma\boldsymbol{q}_i\right) \tag{61d}$$

which is determined by the eigenvalue λ_i. For each mode, the final optimal value at equilibrium is given by

$$\boldsymbol{w}^*_{\boldsymbol{q}_i} = \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)}\boldsymbol{q}_i \tag{62}$$

Thus the final optimal weight matrix W^* is given by

$$W^* = \sum_i \boldsymbol{w}^*_{\boldsymbol{q}_i}\boldsymbol{q}_i^\top \tag{63a}$$

$$= \sum_i \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)}\boldsymbol{q}_i\boldsymbol{q}_i^\top \tag{63b}$$

The optimal loss is given by

$$\Delta^* = \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha\mathbb{E}\left[\|W^*\boldsymbol{z} - \boldsymbol{u}\|^2\right] \tag{64a}$$

$$= \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha\,\mathrm{Tr}\left(W^*\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^\top\right]W^{*\top} - \mathbb{E}\left[\boldsymbol{u}\boldsymbol{z}^\top\right]W^{*\top} - W^*\mathbb{E}\left[\boldsymbol{z}\boldsymbol{u}^\top\right] + \mathbb{E}\left[\boldsymbol{u}\boldsymbol{u}^\top\right]\right) \tag{64b}$$

$$= \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha\,\mathrm{Tr}\Big(W^*(\alpha^2\Sigma + \sigma^2 I)W^{*\top} - (\varphi\alpha\Sigma + \psi\sigma I)W^{*\top} - W^*(\varphi\alpha\Sigma + \psi\sigma I) + (\varphi^2\Sigma + \psi^2 I)\Big) \tag{64c}$$

Note that

$$
\begin{aligned}
W^{*\top} =& W^* && \text{(65a)}\\
W^*\Sigma =& \sum_i \boldsymbol{w}^*_{\boldsymbol{q}_i} \boldsymbol{q}_i^\top \Sigma && \text{(65b)}\\
=& \sum_i \boldsymbol{w}^*_{\boldsymbol{q}_i} (\Sigma \boldsymbol{q}_i)^\top && \text{(65c)}\\
=& \sum_i \boldsymbol{w}^*_{\boldsymbol{q}_i} (\lambda_i \boldsymbol{q}_i)^\top && \text{(65d)}\\
=& \sum_i \lambda_i \boldsymbol{w}^*_{\boldsymbol{q}_i} \boldsymbol{q}_i^\top && \text{(65e)}\\
=& \sum_i \lambda_i \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \boldsymbol{q}_i \boldsymbol{q}_i^\top && \text{(65f)}\\
\Sigma W^{*\top} =& (W^*\Sigma)^\top && \text{(65g)}\\
=& W^*\Sigma && \text{(65h)}\\
W^* W^{*\top} =& \sum_i \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \boldsymbol{q}_i \boldsymbol{q}_i^\top \sum_j \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\alpha^2 + \sigma^2)} \boldsymbol{q}_j \boldsymbol{q}_j^\top && \text{(65i)}\\
=& \sum_{ij} \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\alpha^2 + \sigma^2)} \boldsymbol{q}_i \boldsymbol{q}_i^\top \boldsymbol{q}_j \boldsymbol{q}_j^\top && \text{(65j)}\\
=& \sum_{ij} \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\alpha^2 + \sigma^2)} \boldsymbol{q}_i \delta_{ij} \boldsymbol{q}_j^\top && \text{(65k)}\\
=& \sum_i \left(\frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \right)^2 \boldsymbol{q}_i \boldsymbol{q}_i^\top && \text{(65l)}\\
W^*\Sigma W^{*\top} =& \sum_i \lambda_i \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \boldsymbol{q}_i \boldsymbol{q}_i^\top \sum_j \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_j\alpha^2 + \sigma^2)} \boldsymbol{q}_j \boldsymbol{q}_j^\top && \text{(65m)}\\
=& \sum_i \lambda_i \left(\frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \right)^2 \boldsymbol{q}_i \boldsymbol{q}_i^\top && \text{(65n)}
\end{aligned}
$$

so we have

$$
\begin{aligned}
\Delta^* =& \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha \operatorname{Tr} \Big(W^*(\alpha^2\Sigma + \sigma^2 I) W^{*\top} - (\varphi\alpha\Sigma + \psi\sigma I) W^{*\top} \\
& - W^*(\varphi\alpha\Sigma + \psi\sigma I) + (\varphi^2\Sigma + \psi^2 I) \Big) && \text{(66a)}\\
=& \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha \operatorname{Tr} \Big(W^*(\alpha^2\Sigma + \sigma^2 I) W^{*\top} - 2(\varphi\alpha\Sigma + \psi\sigma I) W^{*\top} + (\varphi^2\Sigma + \psi^2 I) \Big) && \text{(66b)}\\
=& \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha \operatorname{Tr} \Bigg(\alpha^2 \sum_i \lambda_i \left(\frac{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\varphi}\widetilde{\alpha} + \widetilde{\psi}\widetilde{\sigma})}{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\alpha}^2 + \widetilde{\sigma}^2)} \right)^2 \boldsymbol{q}_i \boldsymbol{q}_i^\top + \sigma^2 \sum_i \left(\frac{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\varphi}\widetilde{\alpha} + \widetilde{\psi}\widetilde{\sigma})}{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\alpha}^2 + \widetilde{\sigma}^2)} \right)^2 \boldsymbol{q}_i \boldsymbol{q}_i^\top \\
& - 2\varphi\alpha \sum_i \lambda_i \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \boldsymbol{q}_i \boldsymbol{q}_i^\top - 2\psi\sigma \sum_i \frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha + \psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2 + \sigma^2)} \boldsymbol{q}_i \boldsymbol{q}_i^\top \\
& + (\varphi^2\Sigma + \psi^2 I) \Bigg) && \text{(66c)}
\end{aligned}
$$

Since $\operatorname{Tr}(\boldsymbol{q}_i\boldsymbol{q}_i^\top) = \operatorname{Tr}(\boldsymbol{q}_i^\top\boldsymbol{q}_i) = 1$, we have

$$\begin{aligned}\Delta^* =& \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha\bigg(\alpha^2\sum_i \lambda_i\left(\frac{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\varphi}\widetilde{\alpha}+\widetilde{\psi}\widetilde{\sigma})}{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\alpha}^2+\widetilde{\sigma}^2)}\right)^2 + \sigma^2\sum_i\left(\frac{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\varphi}\widetilde{\alpha}+\widetilde{\psi}\widetilde{\sigma})}{\int \widetilde{\mathcal{D}}\widetilde{\alpha}(\lambda_i\widetilde{\alpha}^2+\widetilde{\sigma}^2)}\right)^2 \\ &- 2\varphi\alpha\sum_i\lambda_i\frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha+\psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2+\sigma^2)} - 2\psi\sigma\sum_i\frac{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha+\psi\sigma)}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2+\sigma^2)} + (\varphi^2\operatorname{Tr}\Sigma + \psi^2\operatorname{Tr} I)\bigg) \end{aligned} \tag{67a}$$

$$= \frac{1}{2}\sum_i\bigg(\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi^2+\psi^2) - \frac{\Big(\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha+\psi\sigma)\Big)^2}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2+\sigma^2)}\bigg) \tag{67b}$$

$$= \frac{1}{2}\operatorname{Tr}\bigg[\int \widetilde{\mathcal{D}}\alpha(\varphi^2\Sigma+\psi^2 I) - \Big(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha\Sigma+\psi\sigma I)\Big)^2\Big(\int \widetilde{\mathcal{D}}\alpha(\alpha^2\Sigma+\sigma^2 I)\Big)^{-1}\bigg] \tag{67c}$$

Note that the optimal loss can be decomposed into contribution from each mode, and for different modes, the loss and thus the corresponding optimal prediction can be different. Specifically, for the mode with eigenvalue λ_i, the contribution to the loss is given by

$$\Delta_i^* = \frac{1}{2}\bigg(\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi^2+\psi^2) - \frac{\Big(\int \widetilde{\mathcal{D}}\alpha(\lambda_i\varphi\alpha+\psi\sigma)\Big)^2}{\int \widetilde{\mathcal{D}}\alpha(\lambda_i\alpha^2+\sigma^2)}\bigg) \tag{68}$$

For larger λ_i, i.e., larger signal-to-noise ratio (SNR), the contribution of φ will dominate, and thus we should expect smaller φ is better. On the other hand, for smaller λ_i, the contribution of ψ dominates, and ψ should be preferred. Indeed, for sufficiently large eigenvalue $\lambda_i \gg 1$, the optimal loss can be well approximated by

$$\Delta_i^* \approx \frac{1}{2}\bigg(\lambda_i\int \widetilde{\mathcal{D}}\alpha\varphi^2 - \frac{(\lambda_i\int \widetilde{\mathcal{D}}\alpha\varphi\alpha)^2}{\lambda_i\int \widetilde{\mathcal{D}}\alpha\alpha^2}\bigg) \tag{69a}$$

$$= \frac{1}{2}\lambda_i\bigg(\int \widetilde{\mathcal{D}}\alpha\varphi^2 - \frac{(\int \widetilde{\mathcal{D}}\alpha\varphi\alpha)^2}{\int \widetilde{\mathcal{D}}\alpha\alpha^2}\bigg) \tag{69b}$$

which is minimized by $\varphi = 0$, i.e., $\boldsymbol{\varepsilon}$-prediction is optimal. On the other hand, for sufficiently small eigenvalue where $\lambda_i \ll 1$, we have

$$\Delta_i^* \approx \frac{1}{2}\bigg(\int \widetilde{\mathcal{D}}\alpha\psi^2 - \frac{(\int \widetilde{\mathcal{D}}\alpha\psi\sigma)^2}{\int \widetilde{\mathcal{D}}\alpha\sigma^2}\bigg) \tag{70a}$$

which is minimized by $\psi = 0$, i.e., $\boldsymbol{x}$-prediction is preferred.

Under the assumptions given by Theorem 3.2, we have

$$\Delta^* = \frac{1}{2}\sum_i\bigg((k^2\lambda_i + (1-k)^2) - \frac{\Big(\frac{1}{2}k\lambda_i - \frac{1}{2}(1-k)\Big)^2}{\frac{1}{3}\lambda_i + \frac{1}{3}}\bigg) \tag{71a}$$

$$= \frac{1}{2}\sum_i\bigg((1+\lambda_i)k^2 - 2k + 1 - \frac{3}{4}\frac{\big((1+\lambda_i)k-1\big)^2}{1+\lambda_i}\bigg) \tag{71b}$$

$$= \frac{1}{2}\sum_i\bigg(\frac{1}{4}(1+\lambda_i)k^2 - 2k + 1 + \frac{3}{2}k - \frac{3}{4}\frac{1}{1+\lambda_i}\bigg) \tag{71c}$$

$$= \frac{1}{8}\sum_i\bigg((1+\lambda_i)k^2 - 2k + \frac{1+4\lambda_i}{1+\lambda_i}\bigg) \tag{71d}$$

$$= \frac{1}{8}\bigg((D+\operatorname{Tr}\Sigma)k^2 - 2Dk + \operatorname{Tr}\big((I+4\Sigma)(I+\Sigma)^{-1}\big)\bigg) \tag{71e}$$

The optimal prediction target is thus given by

$$k^* = \frac{D}{D + \operatorname{Tr}\Sigma} \tag{72}$$

For the special case where $\Sigma = PP^T$ as that in the main text, we have $\operatorname{Tr}\Sigma = d$, which agrees with the result derived in Theorem 3.2.

H Noise-Level-Dependent Linear Analysis

To understand the theoretical benefits of model capacity across noise levels, we extend the analysis of Theorem 3.1 by allowing the linear denoiser to dynamically adapt to the noise level, i.e., $\widehat{\boldsymbol{u}} = W(\alpha)\boldsymbol{z}$. Under this noise-level-dependent setting, the learning dynamics of the weight matrix $W(\alpha)$ are governed by the functional derivative with respect to each α:

$$\tau \frac{\mathrm{d}W(\alpha)}{\mathrm{d}\widetilde{t}} = -\frac{\delta \mathcal{L}}{\delta W(\alpha)} \tag{73}$$

By the definition of the MSE loss in Eq. 4, we have

$$\begin{aligned}
\tau \frac{\mathrm{d}W(\alpha)}{\mathrm{d}\widetilde{t}} &= -\frac{\delta \mathcal{L}}{\delta W(\alpha)} && \text{(74a)}\\
&= -\frac{\delta}{\delta W(\alpha)} \left[\frac{1}{2} \int \widetilde{\mathcal{D}}\alpha \mathbb{E}\left[\|W(\alpha)\boldsymbol{z} - \boldsymbol{u}\|^2 \right] \right] && \text{(74b)}\\
&= -\frac{1}{2} \frac{\widetilde{\mathcal{D}}\alpha}{\mathrm{d}\alpha} \cdot \mathbb{E}\left[\partial_{W(\alpha)} \|W(\alpha)\boldsymbol{z} - \boldsymbol{u}\|^2 \right] && \text{(74c)}\\
&= -\frac{1}{2} \frac{\widetilde{\mathcal{D}}\alpha}{\mathrm{d}\alpha} \cdot \mathbb{E}\left[2\left(W(\alpha)\boldsymbol{z} - \boldsymbol{u}\right)\boldsymbol{z}^\top \right] && \text{(74d)}\\
&= -\frac{\widetilde{\mathcal{D}}\alpha}{\mathrm{d}\alpha} \cdot \left(W(\alpha)\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^\top \right] - \mathbb{E}\left[\boldsymbol{u}\boldsymbol{z}^\top \right] \right) && \text{(74e)}
\end{aligned}$$

where $\frac{\widetilde{\mathcal{D}}\alpha}{d\alpha}$ acts as the local density measure over the diffusion trajectory. Substituting the statistical expectations from Eq. 32, the learning dynamics is simplified as

$$\begin{aligned}
\tau \frac{\mathrm{d}W(\alpha)}{\mathrm{d}\widetilde{t}} &= -\frac{\widetilde{\mathcal{D}}\alpha}{\mathrm{d}\alpha} \cdot \left(W(\alpha)\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^\top \right] - \mathbb{E}\left[\boldsymbol{u}\boldsymbol{z}^\top \right] \right) && \text{(75a)}\\
&= -\frac{\widetilde{\mathcal{D}}\alpha}{\mathrm{d}\alpha} \cdot \left(W(\alpha)\left(\alpha^2 PP^\top + \sigma^2 I \right) - \left(\varphi\alpha PP^\top + \psi\sigma I \right) \right) && \text{(75b)}\\
&= -\frac{\widetilde{\mathcal{D}}\alpha}{\mathrm{d}\alpha} \cdot \left(\alpha^2 W(\alpha)PP^\top + \sigma^2 W(\alpha) - \varphi\alpha PP^\top - \psi\sigma I \right) && \text{(75c)}
\end{aligned}$$

Following the same derivation as for the time-shared linear model, the optimal weights under the equilibrium condition is given by

$$\begin{aligned}
W^*(\alpha) &= W_{\|}^*(\alpha) + W_{\perp}^*(\alpha) && \text{(76a)}\\
&= \frac{\varphi\alpha + \psi\sigma}{\alpha^2 + \sigma^2} PP^T + \frac{\psi}{\sigma}(I - PP^T) && \text{(76b)}
\end{aligned}$$

As in Eq. 41, the optimal loss is

$$\begin{aligned}
\Delta_{\mathtt{dep}}^* =& \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha \operatorname{Tr}\Big(W^*(\alpha)(\alpha^2 PP^\top + \sigma^2 I)W^*(\alpha)^\top - (\varphi\alpha PP^\top + \psi\sigma I)W^*(\alpha)^\top \\
& - W^*(\alpha)(\varphi\alpha PP^\top + \psi\sigma I) + (\varphi^2 PP^\top + \psi^2 I) \Big) && \text{(77a)}\\
=& \frac{1}{2} \int \widetilde{\mathcal{D}}\alpha \Big((\varphi^2 + \psi^2) - \frac{(\varphi\alpha + \psi\sigma)^2}{\alpha^2 + \sigma^2} \Big) \operatorname{Tr}\left(PP^T \right) && \text{(77b)}\\
=& \frac{1}{2} d \int \widetilde{\mathcal{D}}\alpha \frac{(\varphi\sigma - \psi\alpha)^2}{\alpha^2 + \sigma^2} && \text{(77c)}
\end{aligned}$$

Comparison with the Time-Shared Model Comparing Δ^*_{dep} with the optimal loss of the time-shared model (Δ^* in Eq. 12), we have several observations. First, unlike the time-shared model, whose optimal loss contains a persistent ambient penalty proportional to $(D-d)$, the noise-level-dependent loss Δ^*_{dep} collapses entirely into the intra-manifold term. This demonstrates that granting the linear model noise-level dependence completely eliminates the perpendicular residual penalty. Additionally, we can establish a direct analytical relationship using integral inequalities. Specifically, by applying Cauchy-Schwarz-type inequalities:

$$\int \widetilde{\mathcal{D}}\alpha \frac{(\varphi\alpha+\psi\sigma)^2}{\alpha^2+\sigma^2} \int \widetilde{\mathcal{D}}\alpha(\alpha^2+\sigma^2) \geq \Big(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha+\psi\sigma)\Big)^2 \tag{78a}$$

$$\int \widetilde{\mathcal{D}}\alpha\psi^2 \int \widetilde{\mathcal{D}}\alpha\sigma^2 \geq \Big(\int \widetilde{\mathcal{D}}\alpha\psi\sigma\Big)^2 \tag{78b}$$

we can see that

$$\begin{aligned}
\Delta^*_{\text{dep}} =& \frac{1}{2}\int \widetilde{\mathcal{D}}\alpha\Big((\varphi^2+\psi^2) - \frac{(\varphi\alpha+\psi\sigma)^2}{\alpha^2+\sigma^2}\Big)\,\mathrm{Tr}\,(PP^T) && \text{(79a)}\\
\leq& \frac{1}{2}\left(\int \widetilde{\mathcal{D}}\alpha(\varphi^2+\psi^2) - \frac{\Big(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha+\psi\sigma)\Big)^2}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2+\sigma^2)}\right)\mathrm{Tr}(PP^\top) && \text{(79b)}\\
\leq& \frac{1}{2}\left(\int \widetilde{\mathcal{D}}\alpha(\varphi^2+\psi^2) - \frac{\Big(\int \widetilde{\mathcal{D}}\alpha(\varphi\alpha+\psi\sigma)\Big)^2}{\int \widetilde{\mathcal{D}}\alpha(\alpha^2+\sigma^2)}\right)\mathrm{Tr}(PP^\top)\\
&+ \frac{1}{2}\left(\int \widetilde{\mathcal{D}}\alpha\psi^2 - \frac{\Big(\int \widetilde{\mathcal{D}}\alpha\psi\sigma\Big)^2}{\int \widetilde{\mathcal{D}}\alpha\sigma^2}\right)\mathrm{Tr}(I-PP^\top) && \text{(79c)}\\
=& \Delta^* && \text{(79d)}
\end{aligned}$$

This inequality formally proves that allowing the linear denoiser to be noise-level-dependent attains a minimum training loss no greater than that of the time-shared model at the same fixed k. Intuitively, a time-shared model is constrained to find a single global compromise matrix W across all noise levels, whereas the noise-level-dependent formulation ($W(\alpha)$) tailors its capacity to every individual noise scale α, effectively eliminating the sub-optimality introduced by temporal averaging.

I Implementation Details

For latent diffusion, we adopt the training configuration of LightningDiT [21], with the exception of the sampling procedure. We utilize the Heun solver with 50 sampling steps (NFE=99) and a Classifier-Free Guidance (CFG) [22] scale of 1.5. Notably, we observe that our method, when applied to the original flow matching setting of LightningDiT, achieves comparable or superior performance for 64 training epochs.

For pixel-space diffusion, we strictly follow the experimental protocol from JiT [8] for both training and inference, employing the Heun solver with 50 steps and a CFG scale of 2.9.

A comprehensive summary of all architectural and sampling hyperparameters is provided in Table 3. Each experiment is conducted with 8 RTX PRO 6000 GPUs.

J Broader Impacts

By automating the selection of optimal prediction targets, our work enhances training efficiency and reduces the computational energy required to develop high-quality generative models. However, as with any advancement in generative AI, these improvements in model performance also lower the barrier for the potential misuse of diffusion models in creating harmful or deceptive content.

Table 3: **Configurations of experiments.**

	Latent-space Diffusion	Pixel-space Diffusion
	LightningDiT-XL	**JiT-B**
architecture		
tokenizer	VA-VAE [21]	-
depth	28	12
hidden dim	1152	768
heads	16	12
image size	256	256 (other settings: 512)
patch size	1	`image_size` / 16
bottleneck	-	128
dropout	0	0
in-context class tokens	-	32
in-context start block	-	4
training		
loss	$\boldsymbol{v}$-loss	
epochs	64 (ablation), 800	200 (ablation), 600
warmup epochs [35]	0	5
optimizer	Adam [36], $\beta_1, \beta_2 = 0.9, 0.95$	
batch size	1024	
learning rate	2e-4	
learning rate schedule	constant	
weight decay	0	
ema decay	{0.9996, 0.9998, 0.9999}	
time sampler	$\text{logit}(t) \sim \mathcal{N}(\mu, \sigma^2)$	
	$\mu = 0.0, \sigma = 1.0$	$\mu = -0.8, \sigma = 0.8$
noise scale	1.0	$1.0 \times$ `image_size` / 256
clip of $(1-t)$ in division	0.05 for $\boldsymbol{x}$-prediction and 0 for others	0.05
class token drop (for CFG)	0.1	
sampling		
ODE solver	Heun [23]	
ODE steps	50	
time steps	linear in [0.0, 1.0]	
CFG scale [22]	1.5	2.9
CFG interval [37]	[0.1, 1]	

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: This paper introduces a theoretical framework for understanding the relationship between data dimensionality and optimal prediction targets in diffusion models. The authors provide a theoretical analysis alongside a practical, data-driven solution to automatically determine these targets, validating their approach with empirical results on ImageNet at 256×256 and 512×512 resolutions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

Answer: [Yes]

Justification: In the final section the authors talk about the limitation and potential future work of this paper.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: All assumptions in the paper are claimed clearly in the theorem description and further comments, with necessary references of previous work provided. The theorems are proved in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

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: All details, including details of the hyper-parameters and datasets to run the experiments, are provided in the main paper and the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [No]

Justification: The datasets used in this work are all public; the code is being prepared for release and will be made available to the research community shortly.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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: All details for training and verification are provided in both the main text and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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: [No]

   Justification: The paper is mainly about theoretical analysis and the experiments part are relatively easy. Also, running experiments multiple times is very costly and thus not very suitable. However, the authors provided ablation study and results in the appendix for analyzing the impact of the most important hyper-parameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: This part is provided in the appendix for the implementation details.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: This work conforms with the code of ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

Answer: [Yes]

Justification: This part is provided in the appendix.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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: [No]

Justification: This paper is mainly about theory, and the authors currently have no plan to provide checkpoints of the trained models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: [Yes]

Justification: This paper mainly used ImageNet as the training data and cited the original work properly.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A]

Justification: This paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.