ELUCIDATING REPRESENTATION DEGRADATION PROBLEM IN DIFFUSION MODEL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have achieved remarkable success, yet their training remains inefficient due to a severe optimization bottleneck, which we term Representation Degradation. As noise levels increase, the outputs of the trained model exhibit progressive structural distortion, which can destabilize training and impair generation quality. Our analysis suggests that this instability is driven by mismatched target recoverability, which is associated with Neural Tangent Kernel (NTK) spectral weakening and effective low-rank behavior. To address this, we propose Elucidated Representation Diffusion (ERD), a plug-and-play framework that dynamically reallocates optimization effort according to effective recoverability. By stabilizing representation learning without external supervision, ERD accelerates convergence and achieves strong empirical performance across diffusion backbones.

1 INTRODUCTION

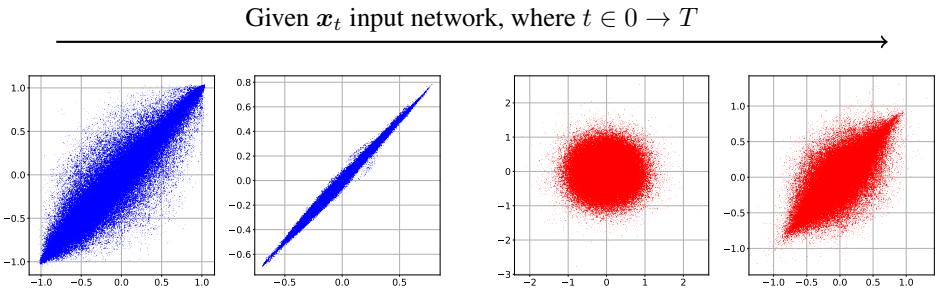
Diffusion models Sohl-Dickstein et al. (2015); Ho et al. (2020); Song et al. (2020b) have become a dominant paradigm for generative modeling Li et al. (2022); Ho et al. (2022b); Kong et al. (2020); Zhou et al. (2021). As the field increasingly shifts toward scalable Transformer-based architectures Peebles & Xie (2023); Bao et al. (2023), improving training efficiency has become a central challenge. Existing methods often accelerate training by heuristically reallocating the optimization effort across noise levels Hang et al. (2023); Karras et al. (2022); Wang et al. (2024); Zheng et al. (2024a); Hang & Gu (2024); Zheng et al. (2024b), but they remain largely agnostic to the quality of the internal features learned. More recently, Representation Alignment (REPA) Yu et al. (2024) has shown that aligning diffusion features with pre-trained priors can substantially accelerate training. This empirical success raises a fundamental question:

> *"what intrinsic bottleneck makes standard diffusion training inefficient in the first place?"*

To answer this question, we empirically examine how the output of the trained model evolves across noise levels. We find that although the predicted geometry aligns well with the target distribution at low noise levels, it becomes increasingly distorted and eventually collapses under high noise. As illustrated in Figure 1, the predicted distribution, when projected onto a 2D plane, preserves meaningful structure at low noise but loses it severely at high noise. We refer to this phenomenon as *Representation Degradation* Gao et al. (2019). These observations reveal an intrinsic instability in diffusion training, demonstrating that as noise increases, the model becomes progressively less able to preserve semantically meaningful structure.

To explain this phenomenon, we analyze diffusion training in the Neural Tangent Kernel (NTK) regime as a continuous-time, noise-conditioned regression process Jacot et al. (2018); Kingma & Gao (2023). Our analysis identifies three interacting factors governing optimization across noise levels: optimization-effort allocation, kernel-mediated contraction, and target-dependent conditional uncertainty. The Bayes decomposition isolates irreducible target uncertainty, while the corresponding conditional residual can contribute stochastic variation through shared-parameter updates. Under a fixed-noise spectral surrogate, smaller NTK eigenvalues imply weaker mode-wise contraction; empirically, we observe spectral weakening as noise increases. This mismatch between training allocation, conditional uncertainty, and contraction strength provides a mechanism that can contribute to representation degradation under high-noise conditions.

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



(a) x_θ-prediction. (b) ϵ_θ-prediction.

Figure 1: We visualized pre-trained model predictions as forward diffusion progressively corrupts the inputs. Figure 1a shows that the image distribution becomes increasingly stretched and flattened, while Figure 1b shows the predicted noise shifting from an isotropic to an anisotropic distribution.

These findings identify representation degradation as an intrinsic optimization bottleneck rather than a purely architectural limitation. To address it, we propose **Elucidated Representation Diffusion (ERD)**, a training framework that reallocates optimization effort using a target-dependent effective-amplitude score without relying on external alignment networks. Specifically, ERD adopts a target-adaptive weighting rule that emphasizes noise levels with stronger effective target exposure and downweights regions receiving excessive optimization mass. In this way, ERD improves optimization stability, mitigates representation degradation, and accelerates convergence. Our main contributions are summarized as follows:

- **Representation Degradation:** We identify *Representation Degradation* in diffusion training, showing that model outputs undergo substantial structural distortion and collapse as noise increases, reflecting an instability in the learned representations.
- **Optimization Analysis:** We develop an NTK-based framework that decomposes diffusion optimization into allocation, target-dependent conditional uncertainty, and kernel-mediated contraction, and empirically observe spectral weakening at high noise.
- **Elucidated Representation Diffusion:** We propose an architecture-agnostic training framework that derives a target-adaptive weighting rule from effective target amplitude, improving optimization stability and accelerating diffusion training without external priors.

2 Preliminaries

In this section, we introduce the diffusion formulation in the log-signal-to-noise ratio (log-SNR) domain and the continuous-time Neural Tangent Kernel framework used in subsequent analysis.

2.1 Diffusion Models in the Log-SNR Domain

Let $x_0 \sim q(x_0)$ be a data sample in $\mathbb{R}^d$. The forward diffusion process generates latent variables through the transition kernel $q(x_t \mid x_0) = \mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 I)$. We reparameterize the process by the log-SNR $\lambda \triangleq \log\left(\frac{\alpha_t^2}{\sigma_t^2}\right)$. We assume that the diffusion schedule is chosen so that the map $t \mapsto \lambda_t$ is strictly monotone on the interval of interest, allowing t and λ to be used interchangeably. Under this parameterization, the corrupted variable can be written as $x_\lambda = \alpha(\lambda)x_0 + \sigma(\lambda)\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$.

We consider a generalized target of the form $y_\lambda = c_x(\lambda)x_0 + c_\epsilon(\lambda)\epsilon$, which subsumes standard parameterizations such as x_0-prediction, ϵ-prediction, and v-prediction. Following the continuous-time variational formulation of diffusion training Kingma et al. (2021); Kingma & Gao (2023), we consider the corresponding weighted training objective

$$\mathcal{L}_{\text{ELBO}}(\theta) = \int_{\lambda_{\min}}^{\lambda_{\max}} M(\lambda) \left(\frac{1}{2} \mathbb{E}_{x_0,\epsilon}\left[\| f_\theta(x_\lambda, \lambda) - y_\lambda \|_2^2 \right] \right) \mathrm{d}\lambda, \tag{1}$$

where f_θ is parameterized by $\theta \in \mathbb{R}^P$, and $M(\lambda) \triangleq \rho(\lambda) w(\lambda) \left| \dot{\lambda}_{t(\lambda)} \right|$ denotes the effective weighting function induced by the loss weighting, the sampling distribution over noise levels, and the diffusion schedule. Its equivalent sampling and importance-sampling representations in the log-SNR domain are established in Lemma C.1. Throughout the paper, $M(\lambda)$ serves as a compact description of how the objective weighting is distributed across the log-SNR domain.

2.2 Continuous-Time Dynamics and the Neural Tangent Kernel

We study Eq. (1) from a continuous-time optimization perspective. Let $\tau \geq 0$ denote training time. Under gradient flow, the parameters evolve according to $\frac{\mathrm{d}\theta(\tau)}{\mathrm{d}\tau} = -\nabla_\theta \mathcal{L}_{\mathrm{ELBO}}(\theta)$.

Because diffusion models use a shared-parameter architecture across all noise levels, optimization at one noise level generally interacts with optimization at others. This motivates describing the training process on the joint input–noise space rather than treating each noise level independently. For this purpose, we introduce the matrix-valued joint Neural Tangent Kernel (NTK)

$$\Theta\big((x,\lambda),(x',\lambda')\big) = \big(\nabla_\theta f_\theta(x,\lambda)\big)\big(\nabla_\theta f_\theta(x',\lambda')\big)^\top \in \mathbb{R}^{d\times d}, \tag{2}$$

where $\nabla_\theta f_\theta(x,\lambda) \in \mathbb{R}^{d\times P}$ denotes the Jacobian of the network output with respect to the parameters.

This joint kernel captures both coupling across output dimensions and cross-noise interactions induced by parameter sharing. In particular, it provides a convenient framework for describing how training signals at different noise levels interact along a shared optimization trajectory.

In the overparameterized regime Jacot et al. (2018), the NTK perspective offers a useful approximation for analyzing training dynamics. In the following sections, we will build on this viewpoint to study how effective weighting, input corruption, and target structure jointly influence diffusion training across noise levels.

3 Elucidating Representation Degradation

To explain standard diffusion training inefficiency, we study overparameterized continuous-time dynamics. By connecting joint NTK dynamics with the global ELBO, we formalize target-dependent *recoverability mismatch*: the effective weighting may emphasize weakly recoverable signals. Under our surrogate dynamics, the resulting excess-risk dynamics separate kernel-induced contraction from conditional-residual forcing, providing a mechanism that can affect representation learning. Proofs are in Appendices A to D.

3.1 Neural Tangent Kernel Dynamics in Diffusion Models

We first derive continuous-time dynamics on the joint input–noise space. Under the assumptions in Appendix A, let τ denote training time and m the width scaling parameter. The time-dependent joint NTK is denoted by $\Theta_{\tau,m}\big((x,\lambda),(x',\lambda')\big)$, and the residual by $e_{\tau,m}(x,y,\lambda) \triangleq f_{\tau,m}(x,\lambda) - y$.

Proposition 3.1 (Finite-Width Joint Dynamics)**.** *Under the population gradient flow associated with the population squared-error risk, the residual $e_{\tau,m}(x,y,\lambda)$ follows the dynamics:*

$$\frac{\partial}{\partial\tau} e_{\tau,m}(x,y,\lambda) = -\mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}}\big[\Theta_{\tau,m}\big((x,\lambda),(x',\lambda')\big) e_{\tau,m}(x',y',\lambda')\big]\,. \tag{3}$$

Thus, residual evolution at noise level λ is governed by a global expectation over all training noise levels λ'. Through the joint NTK, optimization signals from different noise regimes interact along the same parameter trajectory, so updates from one regime can influence residuals elsewhere. This makes explicit that fixed-noise training cannot, in general, be treated as an isolated subsystem under shared-parameter diffusion training.

3.2 Evidence Lower Bound Dynamics

We next use the joint NTK structure to analyze the global ELBO. For fixed λ, let $f^\star_\lambda(x) \triangleq \mathbb{E}[y_\lambda \mid x_\lambda = x]$ denote the regression function, equivalently the Bayes predictor under squared loss. Under the diffusion parameterization and the assumptions in Appendix A, the decomposition below is exact at the population level.

Lemma 3.2 (ELBO Bayes Decomposition)**.** *For each fixed log-SNR level λ, the local weighted ELBO loss density admits an exact orthogonal decomposition $\ell_{\tau,m}(\lambda) = \ell^\star_\lambda + \bar{\ell}_{\tau,m}(\lambda)$, where $\ell^\star_\lambda \triangleq \frac{1}{2}M(\lambda)\mathbb{E}\big[\|f^\star_\lambda(x_\lambda) - y_\lambda\|_2^2\big]$ is the local Bayes-risk density, and $\bar{\ell}_{\tau,m}(\lambda) \triangleq \frac{1}{2}M(\lambda)\|f_{\tau,m}(\cdot,\lambda) - f^\star_\lambda(\cdot)\|^2_{L^2(q_\lambda)}$ is the local excess-risk density.*

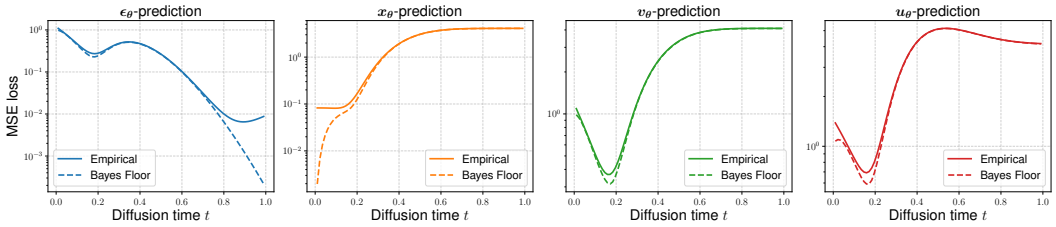


Figure 2: Empirical MSE versus the Bayes optimal error (Bayes floor) across diffusion time t for different prediction targets [1]. The Bayes floor is given by the conditional expectation $\mathbb{E}[y \mid x_t]$. The gap between the empirical loss and the Bayes floor reflects the learnability of each parameterization.

Thus only $\bar{\ell}_{\tau,m}(\lambda)$ is parameter-dependent, while $\ell^\star_\lambda$ is irreducible. Equivalently, the population MSE decomposes into the Bayes risk $\mathbb{E}[\|y_\lambda - \mathbb{E}[y_\lambda \mid x_\lambda]\|_2^2]$ and an excess-risk term. As shown in Figure 2, the Bayes risk lower-bounds the achievable population error, while the remaining gap corresponds to excess risk; under large conditional target uncertainty, the former can dominate the local ELBO. Although ξ_λ is conditionally centered, Corollary C.4 shows that its Jacobian-projected component can have nonzero conditional second moment and contribute stochastic gradient variation. Under the sample-path surrogate below, the same residual appears as the forcing term ξ_τ.

To characterize the excess dynamics, we adopt the continuous-time sample-path surrogate of SGD from Appendix C, with trajectory $\tau \mapsto (x_{\lambda_\tau}, \lambda_\tau)$ and conditional residual $\xi_\tau \triangleq f^\star_{\lambda_\tau}(x_{\lambda_\tau}) - y_{\lambda_\tau}$. Let $r_{\tau,m}(x,\lambda) \triangleq f_{\tau,m}(x,\lambda) - f^\star_\lambda(x)$ denote the estimation error.

Theorem 3.3 (Local Excess-Risk Dynamics)**.** *Under the continuous-time surrogate dynamics, the local excess-risk density satisfies the exact differential identity:*

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) = -M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} - M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\rangle_{L^2(q_\lambda)}. \tag{4}$$

The first term represents the kernel-mediated estimation-error component, whereas the second introduces conditional-residual forcing through the shared-parameter trajectory. To obtain an explicit long-time bound, we impose the pathwise coercivity condition

$$M(\lambda)\Big\langle r_{u,m}(\cdot,\lambda), \frac{M(\lambda_u)}{\rho(\lambda_u)}\Theta_{u,m}\big((\cdot,\lambda),(x_{\lambda_u},\lambda_u)\big) r_{u,m}(x_{\lambda_u},\lambda_u)\Big\rangle_{L^2(q_\lambda)} \geq \mu_u(\lambda)\,\bar{\ell}_{u,m}(\lambda). \tag{5}$$

Here $\mu_u(\lambda)$ is the pathwise coercivity coefficient under the surrogate dynamics. For $\beta > 0$, define $\Gamma_{\tau,s}(\lambda) \triangleq \exp\big(-\int_s^\tau (\mu_u(\lambda) - \beta)\,\mathrm{d}u\big)$ and $F_s(\lambda) \triangleq \frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s$. Integrating Eq. (4) under Eq. (5) yields:

Theorem 3.4 (Non-asymptotic Excess-Risk Bound)**.** *For any evaluation time $\tau \geq 0$, the local excess-risk density is bounded by:*

$$\bar{\ell}_{\tau,m}(\lambda) \leq \Gamma_{\tau,0}(\lambda)\bar{\ell}_{0,m}(\lambda) + \frac{M(\lambda)}{2\beta}\int_0^\tau \Gamma_{\tau,s}(\lambda)\big\|F_s(\lambda)\big\|^2_{L^2(q_\lambda)}\,\mathrm{d}s. \tag{6}$$

Remark 3.5 (Rate interpretation). The bound in Theorem 3.4 yields a continuous-time rate interpretation. If, for a fixed evaluation level λ, the coercivity coefficient satisfies $\mu_u(\lambda) \geq \mu(\lambda) > 0$ and the forcing term is uniformly bounded as $\|F_s(\lambda)\|^2_{L^2(q_\lambda)} \leq B(\lambda)$, then for any $\beta \in (0, \mu(\lambda))$,

$$\bar{\ell}_{\tau,m}(\lambda) \leq e^{-(\mu(\lambda)-\beta)\tau}\bar{\ell}_{0,m}(\lambda) + \frac{M(\lambda)B(\lambda)}{2\beta(\mu(\lambda)-\beta)}\Big(1 - e^{-(\mu(\lambda)-\beta)\tau}\Big). \tag{7}$$

Thus the transient excess contracts at rate $O(e^{-(\mu(\lambda)-\beta)\tau})$ toward a forcing-controlled neighborhood of size $O(M(\lambda)B(\lambda)/[\beta(\mu(\lambda)-\beta)])$. Reaching accuracy ε above this neighborhood requires training time $O\big((\mu(\lambda)-\beta)^{-1}\log(\bar{\ell}_{0,m}(\lambda)/\varepsilon)\big)$. Thus smaller $\mu(\lambda)$ slows contraction, while larger $M(\lambda)$ or $B(\lambda)$ enlarges the forcing-controlled neighborhood.

[1] ϵ and x denote the noise and data, v denotes the velocity Salimans & Ho (2022), and u denotes the flow matching vector field Lipman et al. (2022).

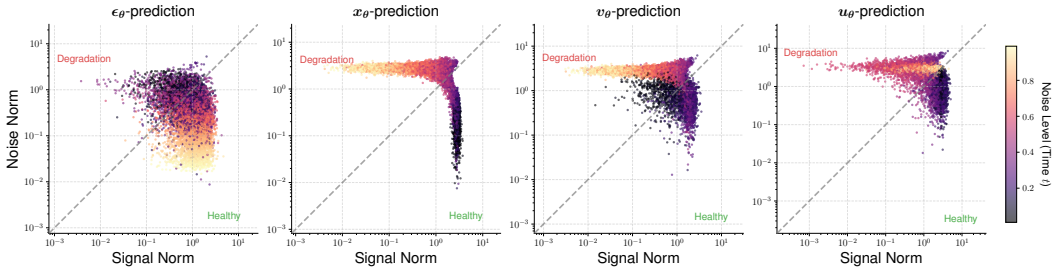


Figure 3: Signal–noise decomposition across diffusion time t. The horizontal axis shows the recoverable signal norm $\|\mathbb{E}[y_\lambda \mid x_t]\|_2$, and the vertical axis shows the irreducible Bayes-noise norm $\|y_\lambda - \mathbb{E}[y_\lambda \mid x_t]\|_2$. Larger vertical values indicate stronger Bayes-noise contamination.

This rate view shows that persistent conditional-residual forcing can control the remaining error even after the initial excess contracts, motivating the mismatch analysis below.

3.3 Stochastic Forcing and Representation Dynamics

The balance between contraction and forcing varies across both noise levels and prediction targets, yielding *recoverability mismatch*: $M(\lambda)$ may remain large in regimes where a substantial component of y_λ is not recoverable from x_λ. Since representation quality is closely tied to posterior estimation quality Li et al. (2025), such mismatch can affect learned features.

For targets whose recoverability depends on the data signal, the direct data-dependent contribution $\alpha(\lambda)x_0$ vanishes in the extreme-noise limit, so the local ELBO may become dominated by target-dependent Bayes risk. Combining this observation with Theorem 3.4 gives:

Theorem 3.6 (Local ELBO Bound)**.** *Under the sample-path surrogate and the pathwise coercivity condition underlying Theorem 3.4, the local ELBO loss density is bounded by three components:*

$$\ell_{\tau,m}(\lambda) \le \underbrace{\ell^\star_\lambda}_{\substack{\text{Target-dependent}\\ \text{Bayes Risk}}} + \underbrace{\Gamma_{\tau,0}(\lambda)\bar{\ell}_{0,m}(\lambda)}_{\text{Contracted Initial Excess}} + \underbrace{\frac{M(\lambda)}{2\beta}\int_0^\tau \Gamma_{\tau,s}(\lambda)\big\|F_s(\lambda)\big\|^2_{L^2(q_\lambda)}\,\mathrm{d}s}_{\text{Cross-Noise Forcing}}. \tag{8}$$

This decomposes the local ELBO into a target-dependent Bayes-risk term, a contracted initial excess term, and cross-noise forcing from shared-parameter training.

Thus the mechanism is not confined to a single noise level; it is mediated by cross-noise interactions through $F_s(\lambda)$. At noise levels where the Bayes risk is large and the coercivity coefficient is small, the local objective contains a larger irreducible contribution together with slower excess-risk contraction, while cross-noise forcing persists through the joint NTK.

We next connect this loss-level mechanism to representation learning. Under $f_\theta(x,\lambda) = W_\theta h_\theta(x,\lambda)$, the same conditional-residual component enters the representation-parameter gradient.

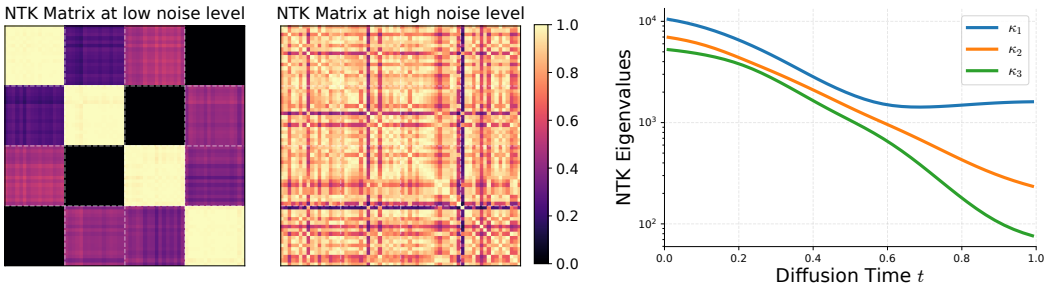
Proposition 3.7 (Representation-Gradient Decomposition)**.** *The stochastic gradient with respect to the representation parameters ϑ exactly decomposes as:*

$$\nabla_\vartheta \widehat{\ell}(\theta) = \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\vartheta h_\theta(x_\lambda,\lambda)\big)^\top W_\theta^\top r_\theta(x_\lambda,\lambda) + \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\vartheta h_\theta(x_\lambda,\lambda)\big)^\top W_\theta^\top \xi_\lambda, \tag{9}$$

where r_θ is the estimation error and ξ_λ is the zero-conditional-mean residual.

The relative magnitudes of the two gradient components depend on their projections through the representation Jacobian and readout. As shown in Figure 3, severe mismatch regimes, *e.g.*, high noise for x_0-prediction or low noise for ϵ-prediction, exhibit larger conditional-residual norms, increasing their potential influence on representation updates under large effective weighting.

Finally, we examine mode-wise contraction using a local fixed-noise spectral surrogate at evaluation level λ. In the NTK regime, the frozen kernel on $L^2(q_\lambda;\mathbb{R}^d)$ admits an orthonormal eigenbasis $\{\phi_j^\lambda\}_{j\ge1}$ with nonnegative eigenvalues $\{\kappa_j^\lambda\}_{j\ge1}$.



(a) NTK heatmaps (b) NTK spectrum

Figure 4: NTK analysis across diffusion noise levels. Figure 4a Joint NTK heatmaps visualize cross-noise coupling induced by shared parameters. Figure 4b The NTK spectrum characterizes mode-wise contraction strength and reveals weakened high-noise optimization modes.

Theorem 3.8 (Spectral Local ELBO Bound)**.** *Under a fixed-noise surrogate approximation, the local ELBO density satisfies the following mode-wise bound for any constant $\gamma > 0$:*

$$\ell_{\tau,m}(\lambda) \le \ell_\lambda^\star + \frac{M(\lambda)}{2}\sum_{j\ge 1} e^{-(2M(\lambda)\kappa_j^\lambda-\gamma)\tau}|a_{0,j}^\lambda|^2 + \frac{M(\lambda)}{2\gamma}\sum_{j\ge 1}\int_0^\tau e^{-(2M(\lambda)\kappa_j^\lambda-\gamma)(\tau-s)}|\eta_{s,j}^\lambda|^2 \mathrm{d}s \quad (10)$$

where $a_{0,j}^\lambda$ and $\eta_{s,j}^\lambda$ denote the modal coefficients of the initial error and forcing, respectively.

This bound gives a mode-wise view of the same mechanism: modes with $2M(\lambda)\kappa_j^\lambda > \gamma$ contract exponentially, while small κ_j^λ yields weaker contraction and slower attenuation of the forcing contribution. As shown in Figure 4, the empirical NTK spectrum weakens at high noise, and the joint NTK heatmaps reveal cross-noise coupling. Together with Proposition 3.7, these observations suggest that weakened spectral contraction and conditional-residual contributions to representation gradients can jointly contribute to representation degradation.

The analysis above motivates reducing the effective weighting assigned to poorly recoverable regimes, which leads to the target-adaptive weighting rule introduced next.

3.4 Elucidated Representation Diffusion

Motivated by the mismatch analysis above, we propose Elucidated Representation Diffusion (ERD), a target-adaptive weighting framework that reallocates optimization effort using a target-dependent effective-amplitude proxy. ERD requires no external priors or architectural modifications.

Using the generalized parameterization from Section 2.1, assume $x_0 \perp \epsilon$, $\mathbb{E}[\|x_0\|_2^2] < \infty$, and $\epsilon \sim \mathcal{N}(0, I)$. Existing efficient training methods reweight losses using target energy, marginal SNR, or timestep allocation Choi et al. (2022); Hang et al. (2023); Yu et al. (2023); Zheng et al. (2024a); Wang et al. (2024). However, these criteria do not explicitly measure how strongly each target component is expressed in the corrupted input. Since x_0 and ϵ enter x_λ with amplitudes $\alpha(\lambda)$ and $\sigma(\lambda)$, respectively, the input-side expression of y_λ depends on both target coefficients and corruption amplitudes.

We define the channel-scaled effective target as $\widetilde{y}_\lambda \triangleq c_x(\lambda)\alpha(\lambda)x_0 + c_\epsilon(\lambda)\sigma(\lambda)\epsilon$. The **effective-amplitude score** $\omega_y(\lambda)$ is its expected root-mean-square amplitude. By the independence of x_0 and ϵ, together with $\epsilon \sim \mathcal{N}(0, I)$, Proposition E.1 gives

$$\omega_y(\lambda) = \sqrt{\frac{1}{d}\mathbb{E}[\|\widetilde{y}_\lambda\|_2^2]} = \sqrt{c_x(\lambda)^2\alpha(\lambda)^2 q_x + c_\epsilon(\lambda)^2\sigma(\lambda)^2}. \quad (11)$$

Here, $q_x \triangleq \frac{1}{d}\mathbb{E}\big[\|x_0\|_2^2\big]$ [2] and $\omega_y(\lambda)$ is a tractable component-wise proxy for target recoverability rather than an exact posterior quantity such as $\mathbb{E}[y_\lambda \mid x_\lambda]$. Unlike SNR- or target-energy-based weighting, ERD weights a target by the amplitude of its components expressed in the corrupted input, so targets with similar energy can receive different weights when their effective amplitudes differ.

[2] In our latent-diffusion experiments, we set $q_x = 1$ after latent-space normalization, an approximation empirically validated in Appendix G.4.



Figure 5: **Selected** 256×256 **samples.** We use a CFG scale of 4.0 and 50 EDM Heun steps.

For fixed noise-level sampling and schedule parameterization, the target-dependent component of the effective allocation $M(\lambda)$ is controlled by the loss weight $w(\lambda)$. ERD sets $w_y(\lambda) \propto \omega_y(\lambda)$ and normalizes it to preserve the average loss scale, so that the reweighting changes only the relative allocation across noise levels. For non-uniform base allocations, the same principle applies after accounting for the sampling density and schedule factor.

The RMS form is sign-invariant: it measures the effective amplitude contributed by each independent source channel and avoids spurious cancellation between components with opposite signs. Therefore, targets with different coefficient signs but identical component amplitudes receive consistent treatment. Substituting the target-specific coefficients $c_x(\lambda)$ and $c_\epsilon(\lambda)$ into Eq. (11) yields a unified weighting rule for canonical diffusion and flow-matching objectives.

Applying $w_y(\lambda)$ reduces optimization mass where the effective-amplitude proxy is weak. This directly targets the target-dependent allocation mismatch identified above by reducing $M(\lambda)$ in regimes with weaker effective target exposure. ERD therefore provides a simple target-adaptive mechanism for reallocating training emphasis without external supervision or architectural overhead.

4 Experiments

This section details the experimental setup for image generation, including datasets, architectures, and training configurations. We present ablation studies and method comparisons to demonstrate our approach's adaptability to different models and targets. Lastly, system-level evaluations assess the overall effectiveness of our method. More experimental results are provided in Appendices F and G.

4.1 Experimental Setup

Datasets. We evaluate image generation on ImageNet Deng et al. (2009) and CelebA Liu et al. (2015). For ImageNet, which contains over 1.3M images from 1,000 classes, we use the 256×256 version and adopt Latent Diffusion Model (LDM) training Rombach et al. (2022) for efficiency, encoding images into 32×32×4 latent representations with the Stable Diffusion VQ-VAE encoder Van Den Oord et al. (2017). For CelebA, which contains over 200K celebrity face images, we follow Song et al. (2020b) to crop and resize images to 64×64 and train models in pixel space for unsupervised generation.

Training. Most experiments use DiT Peebles & Xie (2023) and U-ViT Bao et al. (2023) as backbones optimized via Adam Kingma (2014); Loshchilov & Hutter (2017), while ablations also include UNet Ronneberger et al. (2015) under ADM framework Dhariwal & Nichol (2021). Unless otherwise specified, training is conducted in the LDM latent space Rombach et al. (2022) with VQ-VAE encoding Van Den Oord et al. (2017); we use the DDPM framework Ho et al. (2020) with $T = 1000$ diffusion steps and apply Exponential Moving Average (EMA) with decay 0.9999 Ho et al. (2020).

Evaluation. We evaluate EMA models on 50,000 generated images, *i.e.*, FID-50K. Performance is measured by FID Heusel et al. (2017) and sFID Nash et al. (2021), where lower is better, and Inception Score (IS) Salimans et al. (2016), where higher is better. We report Precision and Recall Kynkäänniemi et al. (2019) for fidelity and coverage. For class-conditional generation, we use classifier-free sampling Ho & Salimans (2022); all sampling uses the Heun sampler from EDM Karras et al. (2022).

Table 1: Training strategy ablation of DiT Peebles & Xie (2023) in latent space Rombach et al. (2022) without classifier-free guidance (CFG).

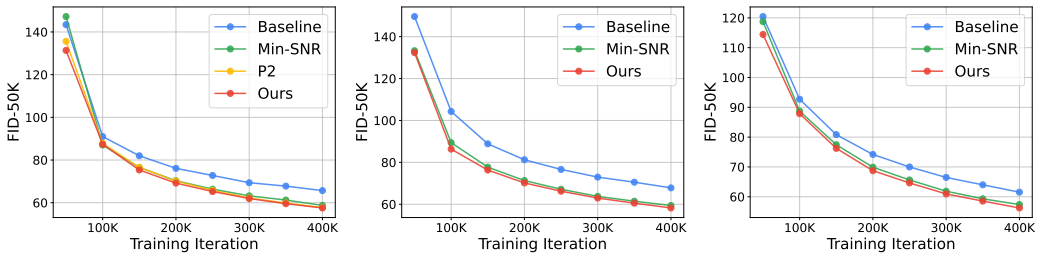
Training Strategy	**FID-50k↓**
Baseline Improvement	
DiT-S/2 Peebles & Xie (2023)	69.35
~~+ Ours~~	~~62.92~~
Training Trick	
DiT-S/2 Peebles & Xie (2023)	69.35
+ Adam $\beta_2 = 0.95$ Ahmadian et al. (2023)	68.77
+ Cosine schedule Nichol & Dhariwal (2021)	65.68
+ Ours	57.54

Table 2: FID results on CelebA 64×64 from architectural ablations centered on UNet and DiT frameworks.

Method	#Params	FID↓
Ablation on U-Net		
DDIM Song et al. (2020a)	79M	3.26
Soft Truncation Song et al. (2020a)	62M	1.90
UNet (Ours)	59M	**1.53**
Ablation on Transformer		
U-ViT-S Bao et al. (2023)	44M	2.87
DiT-S	33M	3.00
DiT-S (Ours)	33M	**2.37**

4.2 Ablation Study

We conduct ablation studies to evaluate the effects of model architecture and training strategy on image generation performance. For training-strategy ablations, we train DiT in latent space on ImageNet for 400K steps with a batch size of 256 and a fixed learning rate of 10^{-4}, keeping other configurations consistent with the pixel-space setup. For architecture ablations, we train UNet (following the architecture configurations in Hang et al. (2023)) and DiT in pixel space on CelebA with a batch size of 128. We optimize both models using Adam and a cosine noise schedule Nichol & Dhariwal (2021) over 500K training iterations. Specifically, we set the learning rate to 10^{-3} with betas $(0.9, 0.95)$ for DiT, and the learning rate to 10^{-4} with betas $(0.9, 0.999)$ for UNet Hang et al. (2023). Table 1 shows consistent improvements across different training strategies, while Table 2 shows our method remains effective across architectures.



(a) ϵ_θ-prediction (b) x_θ-prediction (c) v_θ-prediction

Figure 6: Comparing different loss weighting designs by predicting ϵ_θ, x_θ and v_θ on DiT-S.

4.3 Comparison with Other Methods

Referring to Table 1, we configured the cosine learning rate scheduler with the Adam optimizer's betas set to $(0.9, 0.95)$, and conducted the training with a batch size of 256 over 400K steps, ensuring robust convergence and optimization. We evaluated different prediction objectives, ϵ_θ, x_θ, and v_θ Salimans & Ho (2022), by comparing the Baseline, Min-SNR Hang et al. (2023), P2 Choi et al. (2022), and our proposed weighting method for training DiT-S on ImageNet. As illustrated in Figure 6, our method consistently achieves the lowest FID-50K scores across all prediction objectives, demonstrating superior performance over other weighting methods throughout the training iterations.

4.4 System Level Comparison

We thoroughly evaluate our method on both DiT and U-ViT architectures using the ImageNet 256×256 dataset. In Table 3, we report FID scores without classifier-free guidance (CFG). Under the same training schedule (400K iterations), our method consistently outperforms the original DiT and U-ViT, as well as the SiT baseline, across all model scales. We also evaluate system-level performance under CFG, as shown in Table 4. Compared with strong baselines including LDM Rombach et al. (2022), U-ViT-H/2 Bao et al. (2023), and SD-DiT Zhu et al. (2024), our method achieves highly competitive performance, significantly narrowing the gap with the recent REPA Yu et al. (2024).

Table 3: **FID comparisons with DiTs, SiTs and U-ViT** on ImageNet 256×256. We do not use classifier-free guidance (CFG). ↓ denotes lower values are better. Iter. indicates the training iteration.

Model	#Params	Iter.	FID↓
DiT-B/2	130M	400K	43.4
SiT-B/2	130M	400K	33.0
Ours	**130M**	**400K**	**32.3**
DiT-L/2	458M	400K	23.3
SiT-L/2	458M	400K	18.8
Ours	**458M**	**400K**	**17.2**
DiT-XL/2	675M	400K	19.5
SiT-XL	675M	400K	17.2
Ours	**675M**	**400K**	**15.0**
U-ViT-M/2	131M	400K	27.7
Ours	**131M**	**400K**	**19.9**
U-ViT-L/2	287M	400K	20.4
Ours	**287M**	**400K**	**15.0**
U-ViT-H/2	501M	400K	13.7
Min-SNR	501M	400K	11.7
Ours	**501M**	**400K**	**8.9**

Table 4: **System-level comparison** on ImageNet 256×256 with CFG. ↓ and ↑ indicate whether lower or higher values are better, respectively. Results that include additional CFG scheduling are marked with an asterisk (*), where the guidance interval from Kynkäänniemi et al. (2024) is applied for.

Model	Epochs	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Pixel diffusion						
ADM-U Dhariwal & Nichol (2021)	400	3.94	6.14	186.7	0.82	0.52
VDM++Kingma et al. (2021)	560	2.40	-	225.3	-	-
Simple diffusion Hoogeboom et al. (2023)	800	2.77	-	211.8	-	-
CDM Ho et al. (2022a)	2160	4.88	-	158.7	-	-
Latent diffusion, U-Net						
LDM-4 Rombach et al. (2022)	200	3.60	-	247.7	0.87	0.48
Latent diffusion, Transformer						
MaskDiT Zheng et al. (2023)	1600	2.28	5.67	276.6	0.80	0.61
SD-DiT Zhu et al. (2024)	480	3.23	-	-	-	-
Min-SNR* Hang et al. (2023)	1400	1.57	-	-	-	-
DiT-XL/2 Peebles & Xie (2023)	1400	2.27	4.60	278.2	**0.83**	0.57
SiT-XL/2 Ma et al. (2024)	1400	2.06	4.50	270.3	0.82	0.59
REPA* Yu et al. (2024)	800	**1.42**	4.70	305.7	0.80	**0.65**
Latent diffusion, Transformer + U-Net hybrid						
DiffiT* Hatamizadeh et al. (2024)	-	1.73	-	276.5	0.80	0.62
MDTv2-XL/2* Gao et al. (2023)	1080	1.58	4.52	314.7	0.79	0.65
U-ViT-H/2 Bao et al. (2023)	400	2.29	5.68	263.9	0.82	0.57
Ours	800	1.89	4.48	282.5	0.82	0.60
Ours*	800	1.45	4.51	**315.4**	0.81	0.62

5 Related Work

Diffusion Model Training Acceleration. Prior work improves diffusion training efficiency mainly through resource allocation across noise levels Choi et al. (2022); Hang et al. (2023); Yu et al. (2023); Yao et al. (2024); Xu et al. (2024); Zheng et al. (2024b;a); Wang et al. (2024); Nichol & Dhariwal (2021); Hang & Gu (2024), while other approaches modify the training dynamics or diffusion process Li et al. (2023); Wu et al. (2023). Despite their effectiveness, these methods are largely heuristic and do not explain the intrinsic optimization difficulty of diffusion training.

Representation Alignment in Diffusion Models. Recent works show that improving representation quality can accelerate diffusion training through representation alignment, pretraining, or latent modeling Yu et al. (2024); Leng et al. (2025); Yao et al. (2025); Wu et al. (2025); Jiang et al. (2025); Zheng et al. (2025); Shi et al. (2025); Chu et al. (2025). However, these methods mainly rely on external priors, auxiliary objectives, or modified latent frameworks, without explaining the intrinsic role of learned representations in standard diffusion training. In contrast, we study representation degradation arising within standard diffusion training itself.

Theoretical Analyses of Diffusion and NTK Dynamics. Prior theory has studied diffusion models from variational, dynamical, and score-based perspectives Kingma et al. (2021); Luo (2022); Kingma & Gao (2023); Karras et al. (2024); Han et al. (2024). In parallel, NTK, spectral-bias, and collapse-related work Jacot et al. (2018); Lee et al. (2019); Rahaman et al. (2019); Jing et al. (2022) provides tools for analyzing optimization geometry and representation collapse. These directions remain unconnected in explaining extreme-noise representation degradation. We bridge this gap by linking diffusion optimization to noise-dependent NTK dynamics and empirical representation degradation.

6 Limitations & Conclusion

Limitations. Our analysis relies on continuous-time optimization, NTK-regime, and sample-path surrogate assumptions, which may not fully capture finite-step stochastic and highly nonlinear diffusion training dynamics. In addition, ERD uses an effective-amplitude score as a tractable proxy for target recoverability rather than directly estimating exact posterior recoverability.

Conclusion. Our analysis links representation degradation to target-dependent Bayes risk, conditional-residual forcing, and empirically observed spectral weakening across noise levels. ERD mitigates the resulting allocation mismatch by reweighting training according to the effective-amplitude proxy, improving training efficiency and generation quality across architectures and prediction targets.

AI USE STATEMENT

In this work, generative AI tools are used exclusively for linguistic polishing, textual editing, and phrasing refinement of the manuscript. Generative AI was not used to generate experimental data, execute experiments, produce numerical results, devise core theoretical derivations, or design experimental protocols. All theoretical claims, mathematical derivations, experimental designs, quantitative results, citations, and conclusions have been reviewed and verified by the authors. All AI-generated suggestions were independently validated against mathematical analysis, experimental evidence, and published literature before inclusion. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies optimization and representation dynamics of diffusion models and proposes a loss-weighting strategy to improve training efficiency. No human subjects, crowdsourcing, or new human-subject data are involved. Experiments use established public benchmarks (ImageNet, CelebA) solely for evaluating image generation methods. On the positive side, improved training efficiency may reduce computational and energy costs of generative model training. On the negative side, more efficient generative models may indirectly lower the cost of synthetic media production and share broader misuse risks associated with generative image models. We acknowledge that existing datasets may inherit biases from their collection processes; CelebA is used only as a standard benchmark, with no claims regarding demographic fairness or downstream face applications. This research is conducted in accordance with the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

All results in this work are reproducible. Section 4.1 describes the datasets, diffusion frameworks, model backbones, training configurations, sampling procedures, and evaluation protocols. Theoretical assumptions are stated alongside the corresponding results, with full derivations and proofs provided in Appendices A to E. Appendix F details the controlled 2D Gaussian-mixture experiments, including diffusion schedules, prediction targets, network architecture, optimization settings, and analysis procedures. Appendix G provides implementation details, hyperparameters, computational resources, sampling and evaluation settings, overhead analysis, and complete quantitative results. Experiments are conducted on 8 NVIDIA A100 80GB GPUs using mixed-precision training. All datasets and evaluation protocols are publicly available, and code will be released to facilitate reproduction.

REFERENCES

Arash Ahmadian, Saurabh Dash, Hongyu Chen, Bharat Venkitesh, Zhen Stephen Gou, Phil Blunsom, Ahmet Üstün, and Sara Hooker. Intriguing properties of quantization at scale. *Advances in Neural Information Processing Systems*, 36:34278–34294, 2023. 8

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023. 1, 7, 8, 9, 36, 40

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11472–11481, 2022. 6, 8, 9

Xiangxiang Chu, Renda Li, and Yong Wang. Usp: Unified self-supervised pretraining for image generation and understanding. *arXiv preprint arXiv:2503.06132*, 2025. 9

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 7

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 7, 9, 36

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*, 2019. 1

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023. 9

Yinbin Han, Meisam Razaviyayn, and Renyuan Xu. Neural network-based score estimation in diffusion models: Optimization and generalization. *arXiv preprint arXiv:2401.15604*, 2024. 9

Tiankai Hang and Shuyang Gu. Improved noise schedule for diffusion training. *arXiv preprint arXiv:2407.03297*, 2024. 1, 9

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023. 1, 6, 8, 9

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. In *European Conference on Computer Vision*, pp. 37–55. Springer, 2024. 9

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. 7, 36

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 7

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 7, 32

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022a. 9

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022b. 1

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pp. 13213–13232. PMLR, 2023. 9

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018. 1, 3, 9

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025. 9, 39

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=YevsQ05DEN7. 9

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. 1, 7, 36

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024. 9, 36

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. 1, 2, 9, 31

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021. 2, 9, 31

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7, 36

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 1

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019. 7, 36

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. *arXiv preprint arXiv:2404.07724*, 2024. 9, 40

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019. 9

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025. 9

Wenhao Li, Xiu Su, Shan You, Tao Huang, Fei Wang, Chen Qian, and Chang Xu. Not all steps are equal: Efficient generation with progressive diffusion models. *arXiv preprint arXiv:2312.13307*, 2023. 9

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35: 4328–4343, 2022. 1

Xiao Li, Zekai Zhang, Xiang Li, Siyi Chen, Zhihui Zhu, Peng Wang, and Qing Qu. Understanding representation dynamics of diffusion models via low-dimensional modeling. *arXiv preprint arXiv:2502.05743*, 2025. 5

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 4, 32

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015. 7

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7, 36

Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 9

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024. 9, 32, 38, 39, 40

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 7, 36

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021. 8, 9, 36

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. 1, 7, 8, 9, 36, 38, 40

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019. 9

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022. 7, 8, 9, 36

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015. 7

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 4, 8, 32

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. 7, 36

Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arXiv preprint arXiv:2510.15301*, 2025. 9

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a. 8

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*, 2020b. 1, 7, 32

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016. 36

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 7

Kai Wang, Yukun Zhou, Mingjia Shi, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. A closer look at time steps is worthy of triple speed-up for diffusion model training. *arXiv preprint arXiv:2405.17403*, 2024. 1, 6, 9

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025. 9

Zike Wu, Pan Zhou, Kenji Kawaguchi, and Hanwang Zhang. Fast diffusion model, 2023. URL https://arxiv.org/abs/2306.06991. 9

Tianshuo Xu, Peng Mi, Ruilin Wang, and Yingcong Chen. Towards faster training of diffusion models: An inspiration of a consistency phenomenon. *arXiv preprint arXiv:2404.07946*, 2024. 9

Jingfeng Yao, Wang Cheng, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *arXiv preprint arXiv:2410.10356*, 2024. 9

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*, pp. 15703–15712, 2025. 9

Hu Yu, Li Shen, Jie Huang, Man Zhou, Hongsheng Li, and Feng Zhao. Debias the training of diffusion models. *arXiv preprint arXiv:2310.08442*, 2023. 6, 9

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. 1, 8, 9, 37

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. 9

Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *arXiv preprint arXiv:2306.09305*, 2023. 9

Tianyi Zheng, Cong Geng, Peng-Tao Jiang, Ben Wan, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Non-uniform timestep sampling: Towards faster diffusion model training. In *ACM Multimedia 2024*, 2024a. 1, 6, 9

Tianyi Zheng, Peng-Tao Jiang, Ben Wan, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Beta-tuned timestep diffusion model. In *European Conference on Computer Vision*, 2024b. 1, 9

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5826–5835, 2021. 1

Rui Zhu, Yingwei Pan, Yehao Li, Ting Yao, Zhenglong Sun, Tao Mei, and Chang Wen Chen. Sd-dit: Unleashing the power of self-supervised discrimination in diffusion transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8435–8445, 2024. 8, 9

Elucidating Representation Degradation Problem in Diffusion Model Training
Appendix

CONTENTS

A Foundations and Problem Formulation **15**

B Neural Tangent Kernel Dynamics in Diffusion Models **16**

C Evidence Lower Bound Dynamics **17**

D Stochastic Forcing and Representation Dynamics **24**

E Unified Effective-Amplitude-Aware Loss Weight **29**

F Toy Experiments on 2D GMM Dataset **31**

F.1 Experimental Setup 31

F.2 Gradient Contamination and the Bayes Floor Gap 32

F.3 Representation Collapse in Feature Space. 33

F.4 Neural Tangent Kernel Analysis 33

G Experimental Supplement **36**

G.1 Further Implementation Details. 36

G.2 Evaluation Details 36

G.3 Computational Overhead Analysis 37

G.4 Detailed Quantitative Results 37

G.5 Qualitative Results 40

A FOUNDATIONS AND PROBLEM FORMULATION

In this section, we introduce the mathematical definitions and foundational assumptions used throughout the appendix for analyzing Neural Tangent Kernel dynamics and optimization behavior in diffusion models. We also organize the later results so that the subsequent conclusions are derived explicitly from the preceding ones, rather than appearing as isolated statements.

Definition A.1 (Forward process and joint matrix-valued Neural Tangent Kernel)**.** Let the data distribution be $q(x_0)$ on $\mathbb{R}^d$. The forward diffusion process yields a family of latent variables x_t with conditional law

$$q(x_t \mid x_0) = \mathcal{N}\big(x_t; \alpha_t x_0, \sigma_t^2 I\big), \tag{12}$$

where the log signal-to-noise ratio is defined by

$$\lambda_t = \log\left(\frac{\alpha_t^2}{\sigma_t^2}\right). \tag{13}$$

We assume that the diffusion schedule is chosen so that the map $t \mapsto \lambda_t$ is strictly monotone on the domain of interest. Hence t and λ provide equivalent parameterizations of the same forward corruption process, and we may equivalently write the model as $f_{\theta,m}(x,t)$ or $f_{\theta,m}(x,\lambda)$ with $\lambda = \lambda_t$. We write $\alpha(\lambda) \triangleq \alpha_{t(\lambda)}$ and $\sigma(\lambda) \triangleq \sigma_{t(\lambda)}$.

Let $f_{\theta,m}(x,\lambda) : \mathbb{R}^d \times [\lambda_{\min}, \lambda_{\max}] \to \mathbb{R}^d$ be a shared-parameter neural network parameterized by $\theta \in \mathbb{R}^P$, where m denotes the width scaling parameter. For vector-valued outputs, we use the matrix-valued Neural Tangent Kernel formulation. The joint matrix-valued Neural Tangent Kernel on the input–noise space is defined by

$$\Theta_{\theta,m}\big((x,\lambda),(x',\lambda')\big) = \big(\nabla_\theta f_{\theta,m}(x,\lambda)\big)\big(\nabla_\theta f_{\theta,m}(x',\lambda')\big)^\top \in \mathbb{R}^{d\times d}, \tag{14}$$

where $\nabla_\theta f_{\theta,m}(x,\lambda) \in \mathbb{R}^{d\times P}$ denotes the Jacobian of the network output with respect to the parameters. The kernel $\Theta_{\theta,m}\big((x,\lambda),(x',\lambda')\big)$ captures both coupling across output dimensions and coupling across noise levels induced by parameter sharing. As a local special case, the fixed-noise Neural Tangent Kernel is obtained by restricting both kernel arguments to the same noise level λ, namely

$$\Theta_{\theta,m,\lambda}(x,x') \triangleq \Theta_{\theta,m}\big((x,\lambda),(x',\lambda)\big). \tag{15}$$

Assumption A.2 (Optimization setup)**.** To analyze the continuous-time training dynamics, we impose the following conditions:

- **Shared-parameter architecture:** The same parameter vector θ is used across all noise levels, so the training dynamics are governed by the joint kernel $\Theta_{\theta,m}\big((x,\lambda),(x',\lambda')\big)$ on the input–noise space.
- **Initialization and activation:** The network is initialized with a width-scaled random initialization, and the activation function is assumed to be Lipschitz continuous and at least twice differentiable.
- **Continuous-time dynamics:** The optimization is modeled in continuous training time τ by gradient flow, viewed as a continuous-time idealization of gradient descent.
- **Integrability and regularity:** All expectations and conditional expectations appearing below are well-defined, the relevant second moments are finite, measurable versions of conditional expectations exist, and differentiation may be interchanged with expectation wherever used.
- **Nonnegative weighting function:** The weighting function introduced later satisfies $M(\lambda) \geq 0$ on the entire log-SNR interval.

B Neural Tangent Kernel Dynamics in Diffusion Models

We first establish the shared-parameter finite-width training dynamics on the joint input–noise space. This result serves as the starting point for all subsequent population-risk analyses.

Let m denote the width parameter of the network, or more generally the common scaling parameter controlling the hidden-layer widths in the architecture, and write the model as $f_{\theta,m}(x,\lambda)$. For each fixed finite m, let $p_\lambda(x,y)$ denote the joint distribution of the input–target pair at noise level λ, and consider the population squared-error risk

$$\mathcal{L}_m(\theta) = \frac{1}{2}\mathbb{E}_{\lambda\sim\rho,\ (x,y)\sim p_\lambda}\left[\|f_{\theta,m}(x,\lambda) - y\|_2^2\right], \tag{16}$$

where $\rho(\lambda)$ denotes the sampling distribution over noise levels and $y \in \mathbb{R}^d$ is the corresponding training target. For a gradient-flow trajectory $\theta_m(\tau)$, define $f_{\tau,m}(x,\lambda) \triangleq f_{\theta_m(\tau),m}(x,\lambda)$ and $e_{\tau,m}(x,y,\lambda) \triangleq f_{\tau,m}(x,\lambda) - y$, and write the time-dependent joint matrix-valued Neural Tangent Kernel as

$$\Theta_{\tau,m}\big((x,\lambda),(x',\lambda')\big) \triangleq \big(\nabla_\theta f_{\tau,m}(x,\lambda)\big)\big(\nabla_\theta f_{\tau,m}(x',\lambda')\big)^\top \in \mathbb{R}^{d\times d}. \tag{17}$$

Proposition B.1 (Finite-width training dynamics with shared parameters)**.** *Under Assumption A.2, for each finite width parameter m, if $\theta_m(\tau)$ evolves by gradient flow for the population risk Eq. (16), then the sample residual satisfies*

$$\frac{\partial}{\partial\tau} e_{\tau,m}(x,y,\lambda) = -\mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}}\left[\Theta_{\tau,m}\big((x,\lambda),(x',\lambda')\big) e_{\tau,m}(x',y',\lambda')\right]. \tag{18}$$

Proof. Under the regularity conditions in Assumption A.2, differentiation may be interchanged with expectation. Differentiating Eq. (16) with respect to θ gives

$$\begin{aligned}\nabla_\theta \mathcal{L}_m(\theta) &= \nabla_\theta \left(\frac{1}{2} \mathbb{E}_{\lambda\sim\rho,\ (x,y)\sim p_\lambda} \left[\|f_{\theta,m}(x,\lambda) - y\|_2^2 \right] \right) \\ &= \mathbb{E}_{\lambda\sim\rho,\ (x,y)\sim p_\lambda} \left[\nabla_\theta \left(\frac{1}{2} \|f_{\theta,m}(x,\lambda) - y\|_2^2 \right) \right] \\ &= \mathbb{E}_{\lambda\sim\rho,\ (x,y)\sim p_\lambda} \left[\big(\nabla_\theta f_{\theta,m}(x,\lambda)\big)^\top \big(f_{\theta,m}(x,\lambda) - y\big) \right]. \end{aligned} \tag{19}$$

Using gradient flow, we obtain

$$\begin{aligned}\frac{\mathrm{d}\theta_m(\tau)}{\mathrm{d}\tau} &= -\nabla_\theta \mathcal{L}_m(\theta_m(\tau)) \\ &= -\mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}} \left[\big(\nabla_\theta f_{\tau,m}(x',\lambda')\big)^\top \big(f_{\tau,m}(x',\lambda') - y'\big) \right] \\ &= -\mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}} \left[\big(\nabla_\theta f_{\tau,m}(x',\lambda')\big)^\top e_{\tau,m}(x',y',\lambda') \right]. \end{aligned} \tag{20}$$

Fixing (x, y, λ) and applying the chain rule along the trajectory $\theta_m(\tau)$, we have

$$\begin{aligned}\frac{\partial}{\partial\tau} f_{\tau,m}(x,\lambda) &= \nabla_\theta f_{\tau,m}(x,\lambda) \frac{\mathrm{d}\theta_m(\tau)}{\mathrm{d}\tau} \\ &= -\nabla_\theta f_{\tau,m}(x,\lambda) \mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}} \left[\big(\nabla_\theta f_{\tau,m}(x',\lambda')\big)^\top e_{\tau,m}(x',y',\lambda') \right] \\ &= -\mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}} \left[\nabla_\theta f_{\tau,m}(x,\lambda) \big(\nabla_\theta f_{\tau,m}(x',\lambda')\big)^\top e_{\tau,m}(x',y',\lambda') \right] \\ &= -\mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}} \left[\Theta_{\tau,m}\big((x,\lambda),(x',\lambda')\big) e_{\tau,m}(x',y',\lambda') \right]. \end{aligned} \tag{21}$$

Since $e_{\tau,m}(x, y, \lambda) = f_{\tau,m}(x, \lambda) - y$ and y is independent of τ, it follows that

$$\begin{aligned}\frac{\partial}{\partial\tau} e_{\tau,m}(x,y,\lambda) &= \frac{\partial}{\partial\tau} f_{\tau,m}(x,\lambda) \\ &= -\mathbb{E}_{\lambda'\sim\rho,\ (x',y')\sim p_{\lambda'}} \left[\Theta_{\tau,m}\big((x,\lambda),(x',\lambda')\big) e_{\tau,m}(x',y',\lambda') \right], \end{aligned} \tag{22}$$

which is exactly Eq. (18). ■

C Evidence Lower Bound Dynamics

We now use the joint Neural Tangent Kernel structure established in Proposition B.1 to analyze the global ELBO objective. The key organizational principle of this section is as follows. First, we decompose the local ELBO loss density into a Bayes-risk term and an excess-risk term. Second, we characterize the contribution of the conditional residual to the sampled stochastic gradient. Third, we derive an exact continuous-time excess-risk dynamics identity relative to a formal sample-path surrogate of stochastic gradient descent, from which the later quantitative bounds follow.

We work in the log-SNR domain $\lambda \in [\lambda_{\min}, \lambda_{\max}]$. For a finite-width network $f_{\theta,m}(x, \lambda)$ with width scaling parameter m, let the forward-corrupted input and the generalized diffusion target be parameterized by $x_\lambda = \alpha(\lambda)x_0 + \sigma(\lambda)\epsilon$ and $y_\lambda = c_x(\lambda)x_0 + c_\epsilon(\lambda)\epsilon$, respectively, where $x_0 \sim q(x_0)$ and $\epsilon \sim \mathcal{N}(0, I)$ are independent. This parameterization subsumes standard diffusion targets such as noise prediction and data prediction. Let $\rho(\lambda)$ denote the sampling density over log-SNR levels and let $\dot\lambda_t \triangleq \frac{\mathrm{d}\lambda_t}{\mathrm{d}t}$. Following the minimization-form ELBO convention for diffusion training, we express the continuous-time ELBO objective as

$$\mathcal{L}_{\mathrm{ELBO},m}(\theta) = \int_{\lambda_{\min}}^{\lambda_{\max}} M(\lambda) \left(\frac{1}{2} \mathbb{E}_{x_0\sim q,\ \epsilon\sim\mathcal{N}(0,I)} \left[\|f_{\theta,m}(x_\lambda,\lambda) - y_\lambda\|_2^2 \right] \right) \mathrm{d}\lambda, \tag{23}$$

where $M(\lambda) \triangleq \rho(\lambda) w(\lambda) \left|\dot\lambda_{t(\lambda)}\right|$ denotes the effective weighting function in the log-SNR domain induced by noise-level sampling, loss weighting, and the schedule parameterization. By assumption, $M(\lambda) \geq 0$ on the interval of interest.

Lemma C.1 (Sampling representation)**.** *Suppose that the log-SNR sampling density $\rho(\lambda)$ is induced by a timestep sampling density $p_t(t)$ through a continuously differentiable and strictly monotone map $t \mapsto \lambda_t$ with $\dot{\lambda}_t \neq 0$, so that*

$$\rho(\lambda) = p_t(t(\lambda)) \left| \frac{\mathrm{d}t}{\mathrm{d}\lambda} \right|. \tag{24}$$

Then the ELBO objective admits the equivalent representations

$$\begin{aligned}
\mathcal{L}_{\mathrm{ELBO},m}(\theta) &= \mathbb{E}_{t\sim p_t}\left[w(\lambda_t) \left|\dot{\lambda}_t\right| \left(\frac{1}{2} \mathbb{E}_{x_0\sim q,\ \epsilon\sim\mathcal{N}(0,I)} \left[\|f_{\theta,m}(x_{\lambda_t}, \lambda_t) - y_{\lambda_t}\|_2^2 \right] \right) \right] \\
&= \int_{\lambda_{\min}}^{\lambda_{\max}} M(\lambda) \left(\frac{1}{2} \mathbb{E}_{x_0\sim q,\ \epsilon\sim\mathcal{N}(0,I)} \left[\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2 \right] \right) \mathrm{d}\lambda \\
&= \mathbb{E}_{\lambda\sim\rho}\left[\frac{M(\lambda)}{\rho(\lambda)} \left(\frac{1}{2} \mathbb{E}_{x_0\sim q,\ \epsilon\sim\mathcal{N}(0,I)} \left[\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2 \right] \right) \right],
\end{aligned} \tag{25}$$

where, for $\rho(\lambda) > 0$,

$$\frac{M(\lambda)}{\rho(\lambda)} = w(\lambda) \left|\dot{\lambda}_{t(\lambda)}\right|. \tag{26}$$

When $\rho(\lambda)$ is induced by $p_t(t)$ as above, the sampling density and the schedule factor are linked by the change of variables. In particular,

$$M(\lambda) = p_t(t(\lambda)) w(\lambda), \tag{27}$$

since $\left|\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right| \left|\dot{\lambda}_{t(\lambda)}\right| = 1$. Thus the factorization $M(\lambda) = \rho(\lambda) w(\lambda) |\dot{\lambda}_{t(\lambda)}|$ describes the effective weighting in the log-SNR domain, while these factors need not constitute independent degrees of freedom when ρ is induced by timestep sampling.

More generally, for any proposal density $\pi(\lambda) > 0$ wherever $M(\lambda) > 0$, the same objective can be written as

$$\mathcal{L}_{\mathrm{ELBO},m}(\theta) = \mathbb{E}_{\lambda\sim\pi}\left[\frac{M(\lambda)}{\pi(\lambda)} \left(\frac{1}{2} \mathbb{E}_{x_0\sim q,\ \epsilon\sim\mathcal{N}(0,I)} \left[\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2 \right] \right) \right]. \tag{28}$$

Proof. By the definition of expectation with respect to p_t,

$$\mathcal{L}_{\mathrm{ELBO},m}(\theta) = \int p_t(t) w(\lambda_t) \left|\dot{\lambda}_t\right| \left(\frac{1}{2} \mathbb{E}_{x_0,\epsilon} \left[\|f_{\theta,m}(x_{\lambda_t}, \lambda_t) - y_{\lambda_t}\|_2^2 \right] \right) \mathrm{d}t. \tag{29}$$

Using the change of variables $t = t(\lambda)$ gives

$$\begin{aligned}
\mathcal{L}_{\mathrm{ELBO},m}(\theta) &= \int_{\lambda_{\min}}^{\lambda_{\max}} p_t(t(\lambda)) w(\lambda) \left|\dot{\lambda}_{t(\lambda)}\right| \left|\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right| \left(\frac{1}{2} \mathbb{E}_{x_0,\epsilon} \left[\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2 \right] \right) \mathrm{d}\lambda \\
&= \int_{\lambda_{\min}}^{\lambda_{\max}} \rho(\lambda) w(\lambda) \left|\dot{\lambda}_{t(\lambda)}\right| \left(\frac{1}{2} \mathbb{E}_{x_0,\epsilon} \left[\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2 \right] \right) \mathrm{d}\lambda \\
&= \int_{\lambda_{\min}}^{\lambda_{\max}} M(\lambda) \left(\frac{1}{2} \mathbb{E}_{x_0,\epsilon} \left[\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2 \right] \right) \mathrm{d}\lambda,
\end{aligned} \tag{30}$$

where Eq. (24) and $M(\lambda) = \rho(\lambda) w(\lambda) |\dot{\lambda}_{t(\lambda)}|$ were used. Since $\left|\frac{\mathrm{d}t}{\mathrm{d}\lambda}\right| \left|\dot{\lambda}_{t(\lambda)}\right| = 1$, Eq. (24) also gives $M(\lambda) = p_t(t(\lambda)) w(\lambda)$. Dividing and multiplying the integrand by $\rho(\lambda)$ gives the final representation in Eq. (25). Likewise, dividing and multiplying by an arbitrary proposal density $\pi(\lambda)$ gives Eq. (28). ■

Recall the trajectory output $f_{\tau,m}$ defined above. Under the diffusion parameterization, the instantaneous prediction residual is $e_{\tau,m}(x_\lambda, y_\lambda, \lambda) \triangleq f_{\tau,m}(x_\lambda, \lambda) - y_\lambda$. The corresponding local weighted ELBO loss density is defined by

$$\ell_{\tau,m}(\lambda) \triangleq \frac{1}{2} M(\lambda) \mathbb{E}_{x_0\sim q,\ \epsilon\sim\mathcal{N}(0,I)} \left[\|e_{\tau,m}(x_\lambda, y_\lambda, \lambda)\|_2^2 \right], \tag{31}$$

so that

$$\mathcal{L}_{\mathrm{ELBO},m}(\theta_m(\tau)) = \int_{\lambda_{\min}}^{\lambda_{\max}} \ell_{\tau,m}(\lambda)\,\mathrm{d}\lambda. \tag{32}$$

For each fixed log-SNR level λ, define the regression function, equivalently the Bayes predictor under squared loss, by $f^\star_\lambda(x) \triangleq \mathbb{E}[y_\lambda \mid x_\lambda = x]$. Correspondingly, define the parameter-indexed estimation error and the trajectory-indexed estimation error by $r_{\theta,m}(x,\lambda) \triangleq f_{\theta,m}(x,\lambda) - f^\star_\lambda(x)$ and $r_{\tau,m}(x,\lambda) \triangleq f_{\tau,m}(x,\lambda) - f^\star_\lambda(x)$, the local excess-risk density by

$$\bar{\ell}_{\tau,m}(\lambda) \triangleq \frac{1}{2}M(\lambda)\|r_{\tau,m}(\cdot,\lambda)\|^2_{L^2(q_\lambda)}, \tag{33}$$

and the local Bayes-risk density by

$$\ell^\star_\lambda \triangleq \frac{1}{2}M(\lambda)\mathbb{E}_{x_0\sim q,\ \epsilon\sim\mathcal{N}(0,I)}\left[\|f^\star_\lambda(x_\lambda) - y_\lambda\|_2^2\right]. \tag{34}$$

The next lemma is the basic decomposition from which all later local ELBO statements will be derived.

Lemma C.2 (Bayes-risk decomposition)**.** *For each fixed log-SNR level λ, the local weighted ELBO loss density admits the exact orthogonal decomposition*

$$\ell_{\tau,m}(\lambda) = \ell^\star_\lambda + \bar{\ell}_{\tau,m}(\lambda). \tag{35}$$

In particular,

$$\ell_{\tau,m}(\lambda) \geq \ell^\star_\lambda. \tag{36}$$

Proof. For each fixed λ, decompose the prediction residual around the regression function:

$$\begin{aligned} e_{\tau,m}(x_\lambda, y_\lambda, \lambda) &= f_{\tau,m}(x_\lambda,\lambda) - y_\lambda \\ &= f_{\tau,m}(x_\lambda,\lambda) - f^\star_\lambda(x_\lambda) + f^\star_\lambda(x_\lambda) - y_\lambda \\ &= \underbrace{f_{\tau,m}(x_\lambda,\lambda) - f^\star_\lambda(x_\lambda)}_{r_{\tau,m}(x_\lambda,\lambda)} + \underbrace{f^\star_\lambda(x_\lambda) - y_\lambda}_{\xi_\lambda}. \end{aligned} \tag{37}$$

Substituting Eq. (37) into Eq. (31) yields

$$\begin{aligned} \ell_{\tau,m}(\lambda) &= \frac{1}{2}M(\lambda)\mathbb{E}\left[\|r_{\tau,m}(x_\lambda,\lambda) + \xi_\lambda\|_2^2\right] \\ &= \frac{1}{2}M(\lambda)\mathbb{E}\left[\|r_{\tau,m}(x_\lambda,\lambda)\|_2^2 + \|\xi_\lambda\|_2^2 + 2\langle r_{\tau,m}(x_\lambda,\lambda), \xi_\lambda\rangle\right] \\ &= \frac{1}{2}M(\lambda)\mathbb{E}\|r_{\tau,m}(x_\lambda,\lambda)\|_2^2 + \frac{1}{2}M(\lambda)\mathbb{E}\|\xi_\lambda\|_2^2 + M(\lambda)\mathbb{E}\langle r_{\tau,m}(x_\lambda,\lambda), \xi_\lambda\rangle. \end{aligned} \tag{38}$$

It remains to show that the cross-term vanishes. Since $r_{\tau,m}(x_\lambda,\lambda)$ is a measurable function of x_λ alone and

$$\begin{aligned} \mathbb{E}[\xi_\lambda \mid x_\lambda] &= \mathbb{E}[f^\star_\lambda(x_\lambda) - y_\lambda \mid x_\lambda] \\ &= f^\star_\lambda(x_\lambda) - \mathbb{E}[y_\lambda \mid x_\lambda] \\ &= 0, \end{aligned} \tag{39}$$

the law of total expectation gives

$$\begin{aligned} \mathbb{E}\langle r_{\tau,m}(x_\lambda,\lambda), \xi_\lambda\rangle &= \mathbb{E}[\langle r_{\tau,m}(x_\lambda,\lambda), \mathbb{E}[\xi_\lambda \mid x_\lambda]\rangle] \\ &= 0. \end{aligned} \tag{40}$$

Therefore

$$\begin{aligned} \ell_{\tau,m}(\lambda) &= \frac{1}{2}M(\lambda)\mathbb{E}\|f^\star_\lambda(x_\lambda) - y_\lambda\|_2^2 + \frac{1}{2}M(\lambda)\|f_{\tau,m}(\cdot,\lambda) - f^\star_\lambda(\cdot)\|^2_{L^2(q_\lambda)} \\ &= \ell^\star_\lambda + \bar{\ell}_{\tau,m}(\lambda), \end{aligned} \tag{41}$$

which proves Eq. (35). Since $\bar{\ell}_{\tau,m}(\lambda) \geq 0$, the lower bound Eq. (36) follows immediately. ■

The preceding decomposition is a population identity. We next isolate the corresponding decomposition at the level of the sampled stochastic gradient, which will later be inserted into the continuous-time surrogate dynamics.

Lemma C.3 (Stochastic gradient decomposition)**.** *For a sampled triple (x_0, ϵ, λ) with $\lambda \sim \rho$, define the stochastic local ELBO loss by*

$$\widehat{\ell}_m(\theta; x_0, \epsilon, \lambda) \triangleq \frac{1}{2}\frac{M(\lambda)}{\rho(\lambda)}\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2. \tag{42}$$

Then the corresponding stochastic gradient admits the exact decomposition

$$\begin{aligned}\nabla_\theta \widehat{\ell}_m(\theta; x_0, \epsilon, \lambda) = &\frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\theta f_{\theta,m}(x_\lambda, \lambda)\big)^\top \big(f_{\theta,m}(x_\lambda, \lambda) - f_\lambda^\star(x_\lambda)\big) \\ &+ \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\theta f_{\theta,m}(x_\lambda, \lambda)\big)^\top \big(f_\lambda^\star(x_\lambda) - y_\lambda\big).\end{aligned} \tag{43}$$

Proof. Differentiate Eq. (42) with respect to θ:

$$\begin{aligned}\nabla_\theta \widehat{\ell}_m(\theta; x_0, \epsilon, \lambda) &= \nabla_\theta \left(\frac{1}{2}\frac{M(\lambda)}{\rho(\lambda)}\|f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\|_2^2\right) \\ &= \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\theta f_{\theta,m}(x_\lambda, \lambda)\big)^\top \big(f_{\theta,m}(x_\lambda, \lambda) - y_\lambda\big).\end{aligned} \tag{44}$$

Now decompose the instantaneous prediction residual around the regression function:

$$f_{\theta,m}(x_\lambda, \lambda) - y_\lambda = f_{\theta,m}(x_\lambda, \lambda) - f_\lambda^\star(x_\lambda) + f_\lambda^\star(x_\lambda) - y_\lambda. \tag{45}$$

Substituting Eq. (45) into Eq. (44) yields

$$\begin{aligned}\nabla_\theta \widehat{\ell}_m(\theta; x_0, \epsilon, \lambda) = &\frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\theta f_{\theta,m}(x_\lambda, \lambda)\big)^\top \big(f_{\theta,m}(x_\lambda, \lambda) - f_\lambda^\star(x_\lambda)\big) \\ &+ \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\theta f_{\theta,m}(x_\lambda, \lambda)\big)^\top \big(f_\lambda^\star(x_\lambda) - y_\lambda\big),\end{aligned} \tag{46}$$

which is exactly Eq. (43). ■

The preceding decomposition shows that the conditional residual enters the sampled stochastic gradient explicitly. Although this residual has zero conditional mean, it can still contribute stochastic variation to the parameter update. The following corollary makes this contribution explicit.

Corollary C.4 (Conditional-residual contribution to stochastic gradients)**.** *For a sampled triple (x_0, ϵ, λ) with $\lambda \sim \rho$, define*

$$g_{\xi,\theta,m}(x_0, \epsilon, \lambda) \triangleq \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\theta f_{\theta,m}(x_\lambda, \lambda)\big)^\top \xi_\lambda. \tag{47}$$

Then, conditioned on (x_λ, λ),

$$\mathbb{E}[g_{\xi,\theta,m} \mid x_\lambda, \lambda] = 0, \tag{48}$$

and

$$\mathbb{E}\big[\|g_{\xi,\theta,m}\|_2^2 \mid x_\lambda, \lambda\big] = \left(\frac{M(\lambda)}{\rho(\lambda)}\right)^2 \mathrm{Tr}\Big(\big(\nabla_\theta f_{\theta,m}\big)^\top \mathrm{Cov}(y_\lambda \mid x_\lambda, \lambda)\big(\nabla_\theta f_{\theta,m}\big)\Big), \tag{49}$$

where $\nabla_\theta f_{\theta,m}$ is evaluated at (x_λ, λ). Thus, the conditional residual is conditionally centered but can contribute stochastic variation to the parameter update.

Proof. By Eq. (47) and Eq. (39),

$$\begin{aligned}\mathbb{E}[g_{\xi,\theta,m}(x_0, \epsilon, \lambda) \mid x_\lambda, \lambda] &= \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\theta f_{\theta,m}(x_\lambda, \lambda)\big)^\top \mathbb{E}[\xi_\lambda \mid x_\lambda, \lambda] \\ &= 0,\end{aligned} \tag{50}$$

which proves Eq. (48). Since $\xi_\lambda = f^\star_\lambda(x_\lambda) - y_\lambda$ and $f^\star_\lambda(x_\lambda) = \mathbb{E}[y_\lambda \mid x_\lambda]$, we have

$$\mathbb{E}[\xi_\lambda \xi_\lambda^\top \mid x_\lambda, \lambda] = \mathrm{Cov}(y_\lambda \mid x_\lambda, \lambda). \tag{51}$$

Therefore,

$$\begin{aligned}\mathbb{E}\big[\|g_{\xi,\theta,m}(x_0,\epsilon,\lambda)\|_2^2 \mid x_\lambda, \lambda\big] &= \left(\frac{M(\lambda)}{\rho(\lambda)}\right)^2 \mathbb{E}\Big[\xi_\lambda^\top \big(\nabla_\theta f_{\theta,m}(x_\lambda,\lambda)\big)\big(\nabla_\theta f_{\theta,m}(x_\lambda,\lambda)\big)^\top \xi_\lambda \mid x_\lambda, \lambda\Big] \\ &= \left(\frac{M(\lambda)}{\rho(\lambda)}\right)^2 \mathrm{Tr}\Big(\big(\nabla_\theta f_{\theta,m}(x_\lambda,\lambda)\big)^\top \mathrm{Cov}(y_\lambda \mid x_\lambda, \lambda)\big(\nabla_\theta f_{\theta,m}(x_\lambda,\lambda)\big)\Big),\end{aligned} \tag{52}$$

which proves Eq. (49). ■

The preceding results show that the conditional residual is centered in the conditional mean gradient while remaining as a stochastic component of the sampled update. We next introduce the continuous-time surrogate that connects this stochastic gradient decomposition to a function-space Neural Tangent Kernel dynamics. The following statement should be interpreted as a formal sample-path interpolation of stochastic gradient descent, rather than as a rigorous stochastic-approximation limit theorem.

Lemma C.5 (Continuous-time SGD surrogate)**.** *Consider the stochastic gradient descent update*

$$\theta_{k+1} = \theta_k - \eta\, \nabla_\theta \widehat{\ell}_m(\theta_k; x_0^{(k)}, \epsilon^{(k)}, \lambda^{(k)}), \tag{53}$$

where $\lambda^{(k)} \sim \rho$. At iteration k, let $x_{\lambda^{(k)}} = \alpha(\lambda^{(k)})x_0^{(k)} + \sigma(\lambda^{(k)})\epsilon^{(k)}$ and $\xi^{(k)} = f^\star_{\lambda^{(k)}}(x_{\lambda^{(k)}}) - y_{\lambda^{(k)}}$. Let $\tau = k\eta$ and consider the piecewise-linear interpolation of the iterates. In the small-step-size regime, we adopt the formal sample-path surrogate

$$\begin{aligned}\frac{\mathrm{d}\theta_m(\tau)}{\mathrm{d}\tau} &= -\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\big(\nabla_\theta f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\big)^\top \big(f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau) - f^\star_{\lambda_\tau}(x_{\lambda_\tau})\big) \\ &\quad - \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\big(\nabla_\theta f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\big)^\top \xi_\tau,\end{aligned} \tag{54}$$

where $\xi_\tau \triangleq f^\star_{\lambda_\tau}(x_{\lambda_\tau}) - y_{\lambda_\tau}$. All subsequent identities in this subsection are exact relative to the surrogate dynamics Eq. (54).

Proof. Starting from Eq. (53), divide by η to obtain the difference quotient

$$\frac{\theta_{k+1} - \theta_k}{\eta} = -\nabla_\theta \widehat{\ell}_m(\theta_k; x_0^{(k)}, \epsilon^{(k)}, \lambda^{(k)}). \tag{55}$$

Applying Eq. (43) at the iterate θ_k and the sampled triple $(x_0^{(k)}, \epsilon^{(k)}, \lambda^{(k)})$ gives

$$\begin{aligned}\frac{\theta_{k+1} - \theta_k}{\eta} &= -\frac{M(\lambda^{(k)})}{\rho(\lambda^{(k)})}\big(\nabla_\theta f_{\theta_k,m}(x_{\lambda^{(k)}},\lambda^{(k)})\big)^\top \big(f_{\theta_k,m}(x_{\lambda^{(k)}},\lambda^{(k)}) - f^\star_{\lambda^{(k)}}(x_{\lambda^{(k)}})\big) \\ &\quad - \frac{M(\lambda^{(k)})}{\rho(\lambda^{(k)})}\big(\nabla_\theta f_{\theta_k,m}(x_{\lambda^{(k)}},\lambda^{(k)})\big)^\top \xi^{(k)}.\end{aligned} \tag{56}$$

Introducing $\tau = k\eta$ and replacing the sampled sequence by a continuously refreshed signal $(x_{0,\tau}, \epsilon_\tau, \lambda_\tau)$ yields the surrogate pathwise relation

$$\begin{aligned}\frac{\mathrm{d}\theta_m(\tau)}{\mathrm{d}\tau} &= -\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\big(\nabla_\theta f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\big)^\top \big(f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau) - f^\star_{\lambda_\tau}(x_{\lambda_\tau})\big) \\ &\quad - \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\big(\nabla_\theta f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\big)^\top \xi_\tau,\end{aligned} \tag{57}$$

which is exactly Eq. (54). ■

The following theorem is the central result of this section. All subsequent upper bounds and later consequences will be obtained by explicitly invoking it.

Theorem C.6 (Local excess-risk dynamics)**.** *Under the preceding definitions and relative to the surrogate dynamics Eq. (54), the estimation error satisfies*

$$\frac{\partial}{\partial\tau} r_{\tau,m}(x,\lambda) = -\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((x,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau) - \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((x,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau. \tag{58}$$

Consequently, the local excess-risk density satisfies the exact differential identity

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) = -M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} - M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\rangle_{L^2(q_\lambda)}. \tag{59}$$

Proof. Fix an evaluation pair (x,λ). By the chain rule along the trajectory $\theta_m(\tau)$,

$$\frac{\partial}{\partial\tau} f_{\tau,m}(x,\lambda) = \nabla_\theta f_{\tau,m}(x,\lambda)\frac{\mathrm{d}\theta_m(\tau)}{\mathrm{d}\tau}. \tag{60}$$

Substituting Eq. (54) into Eq. (60), we obtain

$$\begin{aligned}\frac{\partial}{\partial\tau} f_{\tau,m}(x,\lambda) &= -\nabla_\theta f_{\tau,m}(x,\lambda)\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\big(\nabla_\theta f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\big)^\top r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau) \\ &\quad - \nabla_\theta f_{\tau,m}(x,\lambda)\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\big(\nabla_\theta f_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\big)^\top \xi_\tau \\ &= -\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((x,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau) \\ &\quad - \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((x,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau.\end{aligned} \tag{61}$$

Since $f^\star_\lambda(x)$ is independent of τ, differentiating $r_{\tau,m}(x,\lambda) = f_{\tau,m}(x,\lambda) - f^\star_\lambda(x)$ gives

$$\frac{\partial}{\partial\tau} r_{\tau,m}(x,\lambda) = \frac{\partial}{\partial\tau} f_{\tau,m}(x,\lambda), \tag{62}$$

and substituting Eq. (61) proves Eq. (58).

Next differentiate Eq. (33) with respect to τ:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) &= \frac{\mathrm{d}}{\mathrm{d}\tau}\left(\frac{1}{2}M(\lambda)\|r_{\tau,m}(\cdot,\lambda)\|^2_{L^2(q_\lambda)}\right) \\ &= M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{\partial}{\partial\tau} r_{\tau,m}(\cdot,\lambda)\Big\rangle_{L^2(q_\lambda)}.\end{aligned} \tag{63}$$

Substituting Eq. (58) into Eq. (63) yields

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) = -M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} - M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\rangle_{L^2(q_\lambda)}, \tag{64}$$

which is exactly Eq. (59). ∎

The next result is the first quantitative consequence of Theorem C.6. It controls the local excess-risk dynamics by separating the kernel term from the forcing term involving ξ_τ.

Proposition C.7 (Local excess-risk upper bound)**.** *For every $\beta > 0$, the local excess-risk density satisfies*

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) \le -M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} + \beta\,\bar{\ell}_{\tau,m}(\lambda) + \frac{M(\lambda)}{2\beta}\Big\|\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\|^2_{L^2(q_\lambda)}. \tag{65}$$

Moreover, if the trajectory remains in the Neural Tangent Kernel regime so that

$$\Theta_{\tau,m}\big((x,\lambda),(x',\lambda')\big) \approx \Theta_{0,m}\big((x,\lambda),(x',\lambda')\big), \tag{66}$$

then the corresponding frozen-kernel approximation satisfies

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) \lesssim -M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{0,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} + \beta\,\bar{\ell}_{\tau,m}(\lambda) + \frac{M(\lambda)}{2\beta}\Big\| \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{0,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\|^2_{L^2(q_\lambda)}. \tag{67}$$

Proof. Starting from the exact identity Eq. (59) in Theorem C.6, apply Cauchy–Schwarz to the forcing term:

$$\Big|\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\rangle_{L^2(q_\lambda)}\Big| \le \|r_{\tau,m}(\cdot,\lambda)\|_{L^2(q_\lambda)}\Big\| \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\|_{L^2(q_\lambda)}. \tag{68}$$

Applying Young's inequality $ab \le \frac{\beta}{2}a^2 + \frac{1}{2\beta}b^2$ yields

$$\Big|\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\rangle_{L^2(q_\lambda)}\Big| \le \frac{\beta}{2}\|r_{\tau,m}(\cdot,\lambda)\|^2_{L^2(q_\lambda)} + \frac{1}{2\beta}\Big\| \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\|^2_{L^2(q_\lambda)}. \tag{69}$$

Using the identity

$$\frac{1}{2}M(\lambda)\|r_{\tau,m}(\cdot,\lambda)\|^2_{L^2(q_\lambda)} = \bar{\ell}_{\tau,m}(\lambda), \tag{70}$$

we obtain from Eq. (59)

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) &\le -M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} \\ &\quad + \frac{\beta M(\lambda)}{2}\|r_{\tau,m}(\cdot,\lambda)\|^2_{L^2(q_\lambda)} + \frac{M(\lambda)}{2\beta}\Big\| \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\|^2_{L^2(q_\lambda)} \\ &= -M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} \\ &\quad + \beta\,\bar{\ell}_{\tau,m}(\lambda) + \frac{M(\lambda)}{2\beta}\Big\| \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\Big\|^2_{L^2(q_\lambda)},\end{aligned} \tag{71}$$

which proves Eq. (65). Under the frozen-kernel approximation Eq. (66), replacing $\Theta_{\tau,m}$ by $\Theta_{0,m}$ yields Eq. (67). ■

The following pathwise coercivity condition is the additional assumption that allows the preceding differential inequality to be integrated explicitly. It should be interpreted as a sufficient sample-path lower bound on the kernel term and is not implied solely by positive semidefiniteness of the corresponding Neural Tangent Kernel operator.

Theorem C.8 (Non-asymptotic excess-risk bound)**.** *Fix $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ and $\beta > 0$. Suppose that along the trajectory there exists a measurable function $\mu_\tau(\lambda)$ such that*

$$M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big) r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} \ge \mu_\tau(\lambda)\,\bar{\ell}_{\tau,m}(\lambda) \tag{72}$$

for all $\tau \ge 0$. Then for every $\tau \ge 0$,

$$\bar{\ell}_{\tau,m}(\lambda) \le \exp\Big(-\int_0^\tau \big(\mu_s(\lambda) - \beta\big)\,\mathrm{d}s\Big)\,\bar{\ell}_{0,m}(\lambda) + \int_0^\tau \exp\Big(-\int_s^\tau \big(\mu_u(\lambda) - \beta\big)\,\mathrm{d}u\Big)\,\frac{M(\lambda)}{2\beta}\Big\| \frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\Big\|^2_{L^2(q_\lambda)}\,\mathrm{d}s. \tag{73}$$

In particular, if $\mu_\tau(\lambda) \geq \underline{\mu}(\lambda) > \beta$ for all $\tau \geq 0$, then

$$\bar{\ell}_{\tau,m}(\lambda) \leq e^{-(\underline{\mu}(\lambda)-\beta)\tau}\bar{\ell}_{0,m}(\lambda) + \int_0^\tau e^{-(\underline{\mu}(\lambda)-\beta)(\tau-s)}\frac{M(\lambda)}{2\beta}\left\|\frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\right\|^2_{L^2(q_\lambda)}\mathrm{d}s. \tag{74}$$

Consequently,

$$\limsup_{\tau\to\infty}\bar{\ell}_{\tau,m}(\lambda) \leq \limsup_{\tau\to\infty}\int_0^\tau e^{-(\underline{\mu}(\lambda)-\beta)(\tau-s)}\frac{M(\lambda)}{2\beta}\left\|\frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\right\|^2_{L^2(q_\lambda)}\mathrm{d}s. \tag{75}$$

Proof. Under the coercivity condition Eq. (72), the differential inequality in Eq. (65) from Proposition C.7 yields

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\bar{\ell}_{\tau,m}(\lambda) \leq -\big(\mu_\tau(\lambda)-\beta\big)\bar{\ell}_{\tau,m}(\lambda) + \frac{M(\lambda)}{2\beta}\left\|\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\right\|^2_{L^2(q_\lambda)}. \tag{76}$$

Define the integrating-factor phase $\Phi(\tau) \triangleq \int_0^\tau \big(\mu_s(\lambda)-\beta\big)\,\mathrm{d}s$. Multiplying Eq. (76) by $e^{\Phi(\tau)}$ gives

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\Big(e^{\Phi(\tau)}\bar{\ell}_{\tau,m}(\lambda)\Big) \leq e^{\Phi(\tau)}\frac{M(\lambda)}{2\beta}\left\|\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau\right\|^2_{L^2(q_\lambda)}. \tag{77}$$

Integrating from 0 to τ yields

$$e^{\Phi(\tau)}\bar{\ell}_{\tau,m}(\lambda) - \bar{\ell}_{0,m}(\lambda) \leq \int_0^\tau e^{\Phi(s)}\frac{M(\lambda)}{2\beta}\left\|\frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\right\|^2_{L^2(q_\lambda)}\mathrm{d}s. \tag{78}$$

Multiplying both sides by $e^{-\Phi(\tau)}$ and using

$$e^{-\Phi(\tau)}e^{\Phi(s)} = \exp\left(-\int_s^\tau \big(\mu_u(\lambda)-\beta\big)\,\mathrm{d}u\right), \tag{79}$$

we obtain

$$\bar{\ell}_{\tau,m}(\lambda) \leq \exp\left(-\int_0^\tau \big(\mu_s(\lambda)-\beta\big)\,\mathrm{d}s\right)\bar{\ell}_{0,m}(\lambda) + \int_0^\tau \exp\left(-\int_s^\tau \big(\mu_u(\lambda)-\beta\big)\,\mathrm{d}u\right)\frac{M(\lambda)}{2\beta}\left\|\frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\right\|^2_{L^2(q_\lambda)}\mathrm{d}s, \tag{80}$$

which proves Eq. (73). The bound Eq. (74) follows immediately from the lower bound $\mu_\tau(\lambda) \geq \underline{\mu}(\lambda) > \beta$. Finally, taking $\limsup_{\tau\to\infty}$ in Eq. (74) yields Eq. (75). ■

D Stochastic Forcing and Representation Dynamics

We now connect the preceding ELBO-level analysis to representation and spectral dynamics. In particular, the preceding results show that the conditional residual is conditionally unbiased but remains as a stochastic component of the sampled gradient. In this section, we combine the Bayes-risk decomposition in Lemma C.2, the non-asymptotic excess-risk bound in Theorem C.8, and the high-noise limit established next, and then examine how the same conditional residual acts on representation parameters and fixed-noise spectral modes.

To analyze the asymptotic extreme-noise regime, extend the schedule beyond the finite training interval and consider the idealized limit $\lambda \to -\infty$. Assume further that $\alpha(\lambda) \to 0$ and $\sigma(\lambda) \to \sigma_{\max} < \infty$ as $\lambda \to -\infty$.

Lemma D.1 (High-noise Gaussian limit)**.** *Assume in addition that the data distribution has finite second moment $\mathbb{E}_{x_0\sim q}\big[\|x_0\|_2^2\big] < \infty$. Let the input to the network be $x_\lambda = \alpha(\lambda)x_0 + \sigma(\lambda)\epsilon$, where $x_0 \sim q(x_0)$ and $\epsilon \sim \mathcal{N}(0, I)$. As the forward process approaches the extreme-noise regime and $\lambda \to -\infty$, the marginal distribution q_λ of x_λ converges in 2-Wasserstein distance to the isotropic Gaussian distribution $\mathcal{N}(0, \sigma_{\max}^2 I)$, and hence also converges weakly. More precisely,*

$$\mathrm{W}_2\big(q_\lambda, \mathcal{N}(0, \sigma_{\max}^2 I)\big) \to 0 \quad \text{as } \lambda \to -\infty. \tag{81}$$

Proof. By the definition of the 2-Wasserstein metric W_2, we upper bound the transport distance by choosing the natural joint coupling induced by the shared noise variable ϵ. Under this coupling, we obtain

$$\begin{aligned}
\mathrm{W}_2^2\big(q_\lambda, \mathcal{N}(0, \sigma_{\max}^2 I)\big) &\le \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \|(\alpha(\lambda)x_0 + \sigma(\lambda)\epsilon) - \sigma_{\max}\epsilon\|_2^2 \; q(x_0)p(\epsilon)\,\mathrm{d}x_0\,\mathrm{d}\epsilon \\
&= \mathbb{E}_{x_0,\epsilon}\Big[\|\alpha(\lambda)x_0 + (\sigma(\lambda) - \sigma_{\max})\epsilon\|_2^2\Big] \\
&= \mathbb{E}_{x_0,\epsilon}\big[\alpha(\lambda)^2\|x_0\|_2^2 + 2\alpha(\lambda)(\sigma(\lambda) - \sigma_{\max})\langle x_0, \epsilon\rangle + (\sigma(\lambda) - \sigma_{\max})^2\|\epsilon\|_2^2\big] \\
&= \alpha(\lambda)^2\mathbb{E}_{x_0}\big[\|x_0\|_2^2\big] + 2\alpha(\lambda)(\sigma(\lambda) - \sigma_{\max})\mathbb{E}[\langle x_0, \epsilon\rangle] + (\sigma(\lambda) - \sigma_{\max})^2\mathbb{E}_\epsilon\big[\|\epsilon\|_2^2\big] \\
&= \alpha(\lambda)^2\mathbb{E}_{x_0}\big[\|x_0\|_2^2\big] + (\sigma(\lambda) - \sigma_{\max})^2 d, \qquad (82)
\end{aligned}$$

where $\mathbb{E}[\langle x_0, \epsilon\rangle] = 0$ by independence and $\mathbb{E}[\|\epsilon\|_2^2] = d$ for $\epsilon \sim \mathcal{N}(0, I)$. Since $\lim_{\lambda\to-\infty}\alpha(\lambda) = 0$ and $\lim_{\lambda\to-\infty}\sigma(\lambda) = \sigma_{\max}$, the right-hand side of Eq. (82) converges to 0. Hence

$$\lim_{\lambda\to-\infty} \mathrm{W}_2^2\big(q_\lambda, \mathcal{N}(0, \sigma_{\max}^2 I)\big) = 0, \qquad (83)$$

which implies Eq. (81). Therefore q_λ converges in 2-Wasserstein distance, and hence weakly, to $\mathcal{N}(0, \sigma_{\max}^2 I)$. ■

We now combine the Bayes-risk decomposition in Lemma C.2, the excess-risk bound in Theorem C.8, and the high-noise Gaussian limit in Lemma D.1. This yields a local ELBO bound that makes explicit how the Bayes-risk and excess-risk components enter the optimization dynamics.

Corollary D.2 (Local ELBO bound)**.** *Fix $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ and $\beta > 0$. Suppose that along the trajectory there exists a measurable function $\mu_\tau(\lambda)$ such that*

$$M(\lambda)\Big\langle r_{\tau,m}(\cdot,\lambda), \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{\tau,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)r_{\tau,m}(x_{\lambda_\tau},\lambda_\tau)\Big\rangle_{L^2(q_\lambda)} \ge \mu_\tau(\lambda)\,\bar{\ell}_{\tau,m}(\lambda) \qquad (84)$$

for all $\tau \ge 0$. Then the local ELBO loss density admits the bound

$$\begin{aligned}
\ell_{\tau,m}(\lambda) &\le \ell_\lambda^\star + \exp\Big(-\int_0^\tau \big(\mu_s(\lambda) - \beta\big)\,\mathrm{d}s\Big)\,\bar{\ell}_{0,m}(\lambda) \\
&\quad + \int_0^\tau \exp\Big(-\int_s^\tau \big(\mu_u(\lambda) - \beta\big)\,\mathrm{d}u\Big)\,\frac{M(\lambda)}{2\beta}\Big\|\frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\Big\|_{L^2(q_\lambda)}^2\,\mathrm{d}s. \qquad (85)
\end{aligned}$$

In particular, at noise levels where the Bayes-risk density $\ell_\lambda^\star$ is larger and the coercivity coefficient $\mu_\tau(\lambda)$ is smaller, the bound contains a larger irreducible contribution together with slower contraction of the excess-risk term. Moreover, the forcing term depends on cross-noise kernel interactions induced by shared parameters, so the local dynamics at one noise level generally remain coupled to samples drawn at other noise levels.

Proof. By the Bayes-risk decomposition established in Lemma C.2, we have

$$\ell_{\tau,m}(\lambda) = \ell_\lambda^\star + \bar{\ell}_{\tau,m}(\lambda). \qquad (86)$$

By the non-asymptotic excess-risk bound in Theorem C.8, which is itself derived from the excess-risk dynamics in Theorem C.6, the coercivity assumption Eq. (84) yields

$$\begin{aligned}
\bar{\ell}_{\tau,m}(\lambda) &\le \exp\Big(-\int_0^\tau \big(\mu_s(\lambda) - \beta\big)\,\mathrm{d}s\Big)\,\bar{\ell}_{0,m}(\lambda) \\
&\quad + \int_0^\tau \exp\Big(-\int_s^\tau \big(\mu_u(\lambda) - \beta\big)\,\mathrm{d}u\Big)\,\frac{M(\lambda)}{2\beta}\Big\|\frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\Big\|_{L^2(q_\lambda)}^2\,\mathrm{d}s. \qquad (87)
\end{aligned}$$

Substituting Eq. (87) into Eq. (86), we obtain

$$\begin{aligned}
\ell_{\tau,m}(\lambda) &= \ell_\lambda^\star + \bar{\ell}_{\tau,m}(\lambda) \\
&\le \ell_\lambda^\star + \exp\Big(-\int_0^\tau \big(\mu_s(\lambda) - \beta\big)\,\mathrm{d}s\Big)\,\bar{\ell}_{0,m}(\lambda) \\
&\quad + \int_0^\tau \exp\Big(-\int_s^\tau \big(\mu_u(\lambda) - \beta\big)\,\mathrm{d}u\Big)\,\frac{M(\lambda)}{2\beta}\Big\|\frac{M(\lambda_s)}{\rho(\lambda_s)}\Theta_{s,m}\big((\cdot,\lambda),(x_{\lambda_s},\lambda_s)\big)\xi_s\Big\|_{L^2(q_\lambda)}^2\,\mathrm{d}s, \qquad (88)
\end{aligned}$$

which is exactly Eq. (85).

It remains to relate this bound to the high-noise limit. By Lemma D.1, in the idealized extreme-noise regime the corrupted input distribution approaches an isotropic Gaussian law. Moreover, since $\alpha(\lambda) \to 0$, the direct data-dependent contribution $\alpha(\lambda)x_0$ to the corrupted input vanishes in this limit. The implication for target uncertainty depends on the target parameterization $y_\lambda = c_x(\lambda)x_0 + c_\epsilon(\lambda)\epsilon$: the direct x_0-dependent contribution to the corrupted input vanishes in the extreme-noise limit, whereas noise-dependent target components need not exhibit the same limiting behavior. Accordingly, the weighted Bayes-risk density $\ell^\star_\lambda$ is target dependent. Whenever $\ell^\star_\lambda$ is larger and the coercivity coefficient $\mu_\tau(\lambda)$ is smaller at a given noise level, the bound above reflects both a larger irreducible contribution and slower contraction of the excess risk, while the remaining forcing term captures the contribution of the conditional residual through cross-noise kernel interactions. ■

The preceding corollary characterizes the loss-level decomposition and excess-risk dynamics. We next show that the same conditional residual that contributes stochastic variation to the sampled gradient also appears explicitly in the gradient acting on learned representations.

Proposition D.3 (Representation-gradient decomposition)**.** *Suppose the network admits a representation-readout factorization*

$$f_{\theta,m}(x,\lambda) = W_{\theta,m}h_{\theta,m}(x,\lambda), \tag{89}$$

where $h_{\theta,m}(x,\lambda) \in \mathbb{R}^m$ is the representation and $W_{\theta,m} \in \mathbb{R}^{d\times m}$ is the readout. Let $\vartheta \subseteq \theta$ denote parameters that affect $h_{\theta,m}$ but leave $W_{\theta,m}$ fixed. Then for a sampled triple (x_0,ϵ,λ) with $\lambda \sim \rho$,

$$\begin{aligned}\nabla_\vartheta \widehat{\ell}_m(\theta;x_0,\epsilon,\lambda) = &\frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\vartheta h_{\theta,m}(x_\lambda,\lambda)\big)^\top W_{\theta,m}^\top r_{\theta,m}(x_\lambda,\lambda)\\ &+\frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\vartheta h_{\theta,m}(x_\lambda,\lambda)\big)^\top W_{\theta,m}^\top \xi_\lambda.\end{aligned} \tag{90}$$

Consequently, the gradient norm satisfies

$$\big\|\nabla_\vartheta \widehat{\ell}_m(\theta;x_0,\epsilon,\lambda)\big\|_2 \le \frac{M(\lambda)}{\rho(\lambda)}\big\|\nabla_\vartheta h_{\theta,m}(x_\lambda,\lambda)\big\|_{\mathrm{F}}\|W_{\theta,m}\|_{\mathrm{op}}\Big(\|r_{\theta,m}(x_\lambda,\lambda)\|_2 + \|\xi_\lambda\|_2\Big). \tag{91}$$

Therefore, the stochastic gradient acting on the representation parameters contains both an estimation-error component and a component induced by the conditional residual ξ_λ. Their relative magnitudes depend on their images under the representation Jacobian and readout and cannot in general be determined from $\|r_{\theta,m}(x_\lambda,\lambda)\|_2$ and $\|\xi_\lambda\|_2$ alone.

Proof. For the sampled local loss $\widehat{\ell}_m(\theta;x_0,\epsilon,\lambda) \triangleq \frac{1}{2}\frac{M(\lambda)}{\rho(\lambda)}\|f_{\theta,m}(x_\lambda,\lambda) - y_\lambda\|_2^2$, differentiate with respect to ϑ. Since $M(\lambda)/\rho(\lambda)$ is independent of θ, we obtain

$$\nabla_\vartheta \widehat{\ell}_m(\theta;x_0,\epsilon,\lambda) = \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\vartheta f_{\theta,m}(x_\lambda,\lambda)\big)^\top\big(f_{\theta,m}(x_\lambda,\lambda) - y_\lambda\big). \tag{92}$$

Because ϑ affects $h_{\theta,m}$ but leaves $W_{\theta,m}$ fixed, the chain rule gives

$$\begin{aligned}\nabla_\vartheta f_{\theta,m}(x_\lambda,\lambda) &= \nabla_\vartheta\big(W_{\theta,m}h_{\theta,m}(x_\lambda,\lambda)\big)\\ &= W_{\theta,m}\,\nabla_\vartheta h_{\theta,m}(x_\lambda,\lambda).\end{aligned} \tag{93}$$

Substituting Eq. (93) into Eq. (92), we obtain

$$\nabla_\vartheta \widehat{\ell}_m(\theta;x_0,\epsilon,\lambda) = \frac{M(\lambda)}{\rho(\lambda)}\big(\nabla_\vartheta h_{\theta,m}(x_\lambda,\lambda)\big)^\top W_{\theta,m}^\top\big(f_{\theta,m}(x_\lambda,\lambda) - y_\lambda\big). \tag{94}$$

Now decompose the instantaneous prediction residual around the regression function:

$$\begin{aligned}f_{\theta,m}(x_\lambda,\lambda) - y_\lambda &= f_{\theta,m}(x_\lambda,\lambda) - f^\star_\lambda(x_\lambda) + f^\star_\lambda(x_\lambda) - y_\lambda\\ &= r_{\theta,m}(x_\lambda,\lambda) + \xi_\lambda.\end{aligned} \tag{95}$$

Substituting Eq. (95) into Eq. (94) yields

$$\begin{aligned}\nabla_\vartheta \widehat{\ell}_m(\theta; x_0, \epsilon, \lambda) &= \frac{M(\lambda)}{\rho(\lambda)} \big(\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big)^\top W_{\theta,m}^\top \big(r_{\theta,m}(x_\lambda, \lambda) + \xi_\lambda\big) \\ &= \frac{M(\lambda)}{\rho(\lambda)} \big(\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big)^\top W_{\theta,m}^\top r_{\theta,m}(x_\lambda, \lambda) \\ &\quad + \frac{M(\lambda)}{\rho(\lambda)} \big(\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big)^\top W_{\theta,m}^\top \xi_\lambda, \end{aligned} \tag{96}$$

which proves Eq. (90).

To prove the norm bound, apply the triangle inequality to Eq. (90):

$$\begin{aligned}\big\|\nabla_\vartheta \widehat{\ell}_m(\theta; x_0, \epsilon, \lambda)\big\|_2 &\le \frac{M(\lambda)}{\rho(\lambda)} \Big\| \big(\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big)^\top W_{\theta,m}^\top r_{\theta,m}(x_\lambda, \lambda) \Big\|_2 \\ &\quad + \frac{M(\lambda)}{\rho(\lambda)} \Big\| \big(\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big)^\top W_{\theta,m}^\top \xi_\lambda \Big\|_2. \end{aligned} \tag{97}$$

Using the operator bound

$$\Big\| \big(\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big)^\top W_{\theta,m}^\top v \Big\|_2 \le \big\|\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big\|_{\mathrm{F}} \|W_{\theta,m}\|_{\mathrm{op}} \|v\|_2 \tag{98}$$

for any $v \in \mathbb{R}^d$, and applying Eq. (98) first with $v = r_{\theta,m}(x_\lambda, \lambda)$ and then with $v = \xi_\lambda$, we obtain

$$\begin{aligned}\big\|\nabla_\vartheta \widehat{\ell}_m(\theta; x_0, \epsilon, \lambda)\big\|_2 &\le \frac{M(\lambda)}{\rho(\lambda)} \big\|\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big\|_{\mathrm{F}} \|W_{\theta,m}\|_{\mathrm{op}} \|r_{\theta,m}(x_\lambda, \lambda)\|_2 \\ &\quad + \frac{M(\lambda)}{\rho(\lambda)} \big\|\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big\|_{\mathrm{F}} \|W_{\theta,m}\|_{\mathrm{op}} \|\xi_\lambda\|_2 \\ &= \frac{M(\lambda)}{\rho(\lambda)} \big\|\nabla_\vartheta h_{\theta,m}(x_\lambda, \lambda)\big\|_{\mathrm{F}} \|W_{\theta,m}\|_{\mathrm{op}} \Big(\|r_{\theta,m}(x_\lambda, \lambda)\|_2 + \|\xi_\lambda\|_2 \Big), \end{aligned} \tag{99}$$

which is exactly Eq. (91). The decomposition therefore identifies an estimation-error component and a conditional-residual component in the representation gradient, while the norm bound controls their combined magnitude. ■

Together, Corollary D.2 and Proposition D.3 provide complementary views of the same mechanism: in the extreme-noise regime the direct data-dependent component of the corrupted input vanishes, the local excess-risk dynamics contain both contraction and conditional-residual forcing, and the same residual acts explicitly on representation-parameter updates.

Finally, under an additional fixed-noise surrogate closure, the preceding mechanism admits a modal refinement in the eigenbasis of the frozen fixed-noise kernel operator. The following theorem characterizes the optimization consequence of a given fixed-noise kernel spectrum: smaller kernel eigenvalues imply weaker mode-wise contraction. It does not establish that the eigenvalues themselves decrease with the noise level, nor does it imply rank degradation of the hidden representation. These noise-dependent spectral and representation changes are examined empirically.

Theorem D.4 (Fixed-noise spectral bound)**.** *Fix $\lambda \in [\lambda_{\min}, \lambda_{\max}]$. Assume that the frozen-kernel approximation Eq. (66) holds on the time interval of interest and that, for the purpose of local spectral analysis at the fixed noise level λ, the estimation error satisfies*

$$\begin{aligned}\frac{\partial}{\partial \tau} r_{\tau,m}(\cdot, \lambda) &= -M(\lambda) \int \Theta_{0,m}\big((\cdot, \lambda), (x', \lambda)\big) r_{\tau,m}(x', \lambda)\, q_\lambda(x')\, \mathrm{d}x' \\ &\quad - \frac{M(\lambda_\tau)}{\rho(\lambda_\tau)} \Theta_{0,m}\big((\cdot, \lambda), (x_{\lambda_\tau}, \lambda_\tau)\big) \xi_\tau. \end{aligned} \tag{100}$$

Assume moreover that

$$-\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)} \Theta_{0,m}\big((\cdot, \lambda), (x_{\lambda_\tau}, \lambda_\tau)\big) \xi_\tau \in L^2(q_\lambda; \mathbb{R}^d) \tag{101}$$

for all $\tau \geq 0$. Assume further that there exists a complete orthonormal system $\{\phi_j^\lambda\}_{j\geq 1} \subset L^2(q_\lambda;\mathbb{R}^d)$ with nonnegative eigenvalues $\{\kappa_j^\lambda\}_{j\geq 1}$ such that

$$\int \Theta_{0,m}\big((x,\lambda),(x',\lambda)\big)\phi_j^\lambda(x')\,q_\lambda(x')\,\mathrm{d}x' = \kappa_j^\lambda \phi_j^\lambda(x), \qquad j \geq 1. \tag{102}$$

Write, in $L^2(q_\lambda;\mathbb{R}^d)$,

$$r_{\tau,m}(x,\lambda) = \sum_{j\geq 1} a_{\tau,j}^\lambda \phi_j^\lambda(x), \qquad a_{\tau,j}^\lambda = \langle r_{\tau,m}(\cdot,\lambda), \phi_j^\lambda\rangle_{L^2(q_\lambda)}, \tag{103}$$

and define

$$\eta_{\tau,j}^\lambda = \Big\langle -\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{0,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau, \phi_j^\lambda\Big\rangle_{L^2(q_\lambda)}. \tag{104}$$

Then for every $\gamma > 0$,

$$\begin{aligned}\ell_{\tau,m}(\lambda) \leq \ell_\lambda^\star &+ \frac{M(\lambda)}{2}\sum_{j\geq 1} e^{-(2M(\lambda)\kappa_j^\lambda-\gamma)\tau}|a_{0,j}^\lambda|^2 \\ &+ \frac{M(\lambda)}{2\gamma}\sum_{j\geq 1}\int_0^\tau e^{-(2M(\lambda)\kappa_j^\lambda-\gamma)(\tau-s)}|\eta_{s,j}^\lambda|^2\,\mathrm{d}s.\end{aligned} \tag{105}$$

In particular, for modes satisfying $2M(\lambda)\kappa_j^\lambda > \gamma$, the corresponding contribution decays exponentially, whereas modes associated with smaller eigenvalues exhibit weaker contraction and slower attenuation of the forcing contribution.

Proof. We begin from the fixed-noise surrogate dynamics Eq. (100). By the definition of the modal coefficients in Eq. (103), we have

$$a_{\tau,j}^\lambda = \langle r_{\tau,m}(\cdot,\lambda), \phi_j^\lambda\rangle_{L^2(q_\lambda)}. \tag{106}$$

Differentiating Eq. (106) with respect to τ yields

$$\frac{\mathrm{d}}{\mathrm{d}\tau}a_{\tau,j}^\lambda = \Big\langle \frac{\partial}{\partial\tau} r_{\tau,m}(\cdot,\lambda), \phi_j^\lambda\Big\rangle_{L^2(q_\lambda)}. \tag{107}$$

Substituting Eq. (100) into Eq. (107), we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}a_{\tau,j}^\lambda &= \Big\langle -M(\lambda)\int \Theta_{0,m}\big((\cdot,\lambda),(x',\lambda)\big)r_{\tau,m}(x',\lambda)\,q_\lambda(x')\,\mathrm{d}x', \phi_j^\lambda\Big\rangle_{L^2(q_\lambda)} \\ &\quad + \Big\langle -\frac{M(\lambda_\tau)}{\rho(\lambda_\tau)}\Theta_{0,m}\big((\cdot,\lambda),(x_{\lambda_\tau},\lambda_\tau)\big)\xi_\tau, \phi_j^\lambda\Big\rangle_{L^2(q_\lambda)} \\ &= -M(\lambda)\Big\langle \int \Theta_{0,m}\big((\cdot,\lambda),(x',\lambda)\big)r_{\tau,m}(x',\lambda)\,q_\lambda(x')\,\mathrm{d}x', \phi_j^\lambda\Big\rangle_{L^2(q_\lambda)} + \eta_{\tau,j}^\lambda.\end{aligned} \tag{108}$$

Using the residual expansion Eq. (103), we compute

$$\begin{aligned}\int \Theta_{0,m}\big((x,\lambda),(x',\lambda)\big)r_{\tau,m}(x',\lambda)\,q_\lambda(x')\,\mathrm{d}x' &= \int \Theta_{0,m}\big((x,\lambda),(x',\lambda)\big)\Big(\sum_{k\geq 1} a_{\tau,k}^\lambda\phi_k^\lambda(x')\Big)q_\lambda(x')\,\mathrm{d}x' \\ &= \sum_{k\geq 1} a_{\tau,k}^\lambda \int \Theta_{0,m}\big((x,\lambda),(x',\lambda)\big)\phi_k^\lambda(x')\,q_\lambda(x')\,\mathrm{d}x' \\ &= \sum_{k\geq 1} a_{\tau,k}^\lambda\kappa_k^\lambda\phi_k^\lambda(x),\end{aligned} \tag{109}$$

where the last line follows from Eq. (102). Taking the $L^2(q_\lambda)$ inner product of Eq. (109) with ϕ_j^λ and using orthonormality, we get

$$\begin{aligned}\Big\langle \int \Theta_{0,m}\big((\cdot,\lambda),(x',\lambda)\big)r_{\tau,m}(x',\lambda)\,q_\lambda(x')\,\mathrm{d}x', \phi_j^\lambda\Big\rangle_{L^2(q_\lambda)} &= \Big\langle \sum_{k\geq 1} a_{\tau,k}^\lambda\kappa_k^\lambda\phi_k^\lambda, \phi_j^\lambda\Big\rangle_{L^2(q_\lambda)} \\ &= \sum_{k\geq 1} a_{\tau,k}^\lambda\kappa_k^\lambda\langle\phi_k^\lambda,\phi_j^\lambda\rangle_{L^2(q_\lambda)} \\ &= a_{\tau,j}^\lambda\kappa_j^\lambda.\end{aligned} \tag{110}$$

Substituting Eq. (110) into Eq. (108), we conclude that

$$\frac{\mathrm{d}}{\mathrm{d}\tau}a_{\tau,j}^{\lambda} = -M(\lambda)\kappa_j^{\lambda}a_{\tau,j}^{\lambda} + \eta_{\tau,j}^{\lambda}. \tag{111}$$

We next derive a mode-wise energy bound. Differentiating $|a_{\tau,j}^{\lambda}|^2$ and applying Eq. (111), we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\tau}|a_{\tau,j}^{\lambda}|^2 &= 2a_{\tau,j}^{\lambda}\frac{\mathrm{d}}{\mathrm{d}\tau}a_{\tau,j}^{\lambda} \\ &= 2a_{\tau,j}^{\lambda}\Big(-M(\lambda)\kappa_j^{\lambda}a_{\tau,j}^{\lambda} + \eta_{\tau,j}^{\lambda}\Big) \\ &= -2M(\lambda)\kappa_j^{\lambda}|a_{\tau,j}^{\lambda}|^2 + 2a_{\tau,j}^{\lambda}\eta_{\tau,j}^{\lambda}.\end{aligned} \tag{112}$$

Applying Young's inequality with parameter $\gamma > 0$, we have

$$2|a_{\tau,j}^{\lambda}\eta_{\tau,j}^{\lambda}| \le \gamma|a_{\tau,j}^{\lambda}|^2 + \gamma^{-1}|\eta_{\tau,j}^{\lambda}|^2. \tag{113}$$

Combining Eq. (112) and Eq. (113) gives

$$\frac{\mathrm{d}}{\mathrm{d}\tau}|a_{\tau,j}^{\lambda}|^2 \le -\big(2M(\lambda)\kappa_j^{\lambda} - \gamma\big)|a_{\tau,j}^{\lambda}|^2 + \gamma^{-1}|\eta_{\tau,j}^{\lambda}|^2. \tag{114}$$

Applying the integrating-factor argument to Eq. (114), we obtain

$$|a_{\tau,j}^{\lambda}|^2 \le e^{-(2M(\lambda)\kappa_j^{\lambda}-\gamma)\tau}|a_{0,j}^{\lambda}|^2 + \int_0^{\tau} e^{-(2M(\lambda)\kappa_j^{\lambda}-\gamma)(\tau-s)}\gamma^{-1}|\eta_{s,j}^{\lambda}|^2\,\mathrm{d}s. \tag{115}$$

Since $\{\phi_j^{\lambda}\}_{j\ge1}$ is a complete orthonormal system and the expansion Eq. (103) holds in $L^2(q_\lambda;\mathbb{R}^d)$, Parseval's identity gives

$$\|r_{\tau,m}(\cdot,\lambda)\|_{L^2(q_\lambda)}^2 = \sum_{j\ge1}|a_{\tau,j}^{\lambda}|^2. \tag{116}$$

Recalling that

$$\bar{\ell}_{\tau,m}(\lambda) = \frac{1}{2}M(\lambda)\|r_{\tau,m}(\cdot,\lambda)\|_{L^2(q_\lambda)}^2, \tag{117}$$

we obtain from Eq. (115) and Eq. (116) that

$$\begin{aligned}\bar{\ell}_{\tau,m}(\lambda) &= \frac{M(\lambda)}{2}\sum_{j\ge1}|a_{\tau,j}^{\lambda}|^2 \\ &\le \frac{M(\lambda)}{2}\sum_{j\ge1}e^{-(2M(\lambda)\kappa_j^{\lambda}-\gamma)\tau}|a_{0,j}^{\lambda}|^2 + \frac{M(\lambda)}{2\gamma}\sum_{j\ge1}\int_0^{\tau}e^{-(2M(\lambda)\kappa_j^{\lambda}-\gamma)(\tau-s)}|\eta_{s,j}^{\lambda}|^2\,\mathrm{d}s.\end{aligned} \tag{118}$$

Finally, by the Bayes-risk decomposition in Lemma C.2,

$$\ell_{\tau,m}(\lambda) = \ell_{\lambda}^{\star} + \bar{\ell}_{\tau,m}(\lambda). \tag{119}$$

Substituting Eq. (118) into Eq. (119) yields Eq. (105). ■

E Unified Effective-Amplitude-Aware Loss Weight

The preceding analysis motivates reallocating training emphasis across noise levels in a target-dependent manner. To obtain a simple rule that applies uniformly across different target parameterizations, we use the effective amplitude with which each independent source component is expressed through the corrupted input as a tractable proxy. We now formalize this score and define the corresponding ERD weighting rule for the canonical diffusion and flow-matching targets.

Proposition E.1 (Unified effective-amplitude-aware weight)**.** *Consider the generalized target parameterization*

$$x_\lambda = \alpha(\lambda)x_0 + \sigma(\lambda)\epsilon, \qquad y_\lambda = c_x(\lambda)x_0 + c_\epsilon(\lambda)\epsilon, \tag{120}$$

where $x_0 \perp \epsilon$, $\mathbb{E}[\|x_0\|_2^2] < \infty$, and $\epsilon \sim \mathcal{N}(0, I)$. Consistent with the effective weighting used in Appendix C, define

$$M_y(\lambda) = \rho(\lambda) w_y(\lambda) \left|\dot{\lambda}_{t(\lambda)}\right|. \tag{121}$$

For fixed sampling density $\rho(\lambda)$ and fixed noise schedule, define the effective-amplitude score by

$$\omega_y(\lambda) = \sqrt{\frac{1}{d}\mathbb{E}\Big[\|c_x(\lambda)\alpha(\lambda)x_0 + c_\epsilon(\lambda)\sigma(\lambda)\epsilon\|_2^2\Big]}. \tag{122}$$

Then

$$\omega_y(\lambda) = \sqrt{c_x(\lambda)^2\alpha(\lambda)^2 q_x + c_\epsilon(\lambda)^2\sigma(\lambda)^2}, \tag{123}$$

where

$$q_x \triangleq \frac{1}{d}\mathbb{E}[\|x_0\|_2^2]. \tag{124}$$

We define the ERD weighting rule by

$$w_y(\lambda) \propto \omega_y(\lambda). \tag{125}$$

Consequently, letting $\alpha(\lambda), \sigma(\lambda) \geq 0$, the canonical targets satisfy

$$w_y(\lambda) \propto \begin{cases} \alpha(\lambda)\sqrt{q_x}, & y_\lambda = x_0, \\ \sigma(\lambda), & y_\lambda = \epsilon, \\ \sqrt{\alpha(\lambda)^2 q_x + \sigma(\lambda)^2}, & y_\lambda = u_\lambda = \epsilon - x_0, \\ \alpha(\lambda)\sigma(\lambda)\sqrt{q_x + 1}, & y_\lambda = v_\lambda = \alpha(\lambda)\epsilon - \sigma(\lambda)x_0. \end{cases} \tag{126}$$

Since q_x is independent of λ, the factors $\sqrt{q_x}$ and $\sqrt{q_x + 1}$ can be absorbed into the target-specific normalization constants for the x_0- and v-prediction targets, respectively. Under the normalized second-moment condition $q_x = 1$, the result reduces, up to λ-independent normalization, to $\alpha(\lambda)$, $\sigma(\lambda)$, $\sqrt{\alpha(\lambda)^2 + \sigma(\lambda)^2}$, and $\alpha(\lambda)\sigma(\lambda)$ for the x_0-, ϵ-, u-, and v-prediction targets, respectively.

Proof. Define the effectively scaled target by

$$\widetilde{y}_\lambda \triangleq c_x(\lambda)\alpha(\lambda)x_0 + c_\epsilon(\lambda)\sigma(\lambda)\epsilon. \tag{127}$$

By the definition of $\omega_y(\lambda)$,

$$\begin{aligned} \omega_y(\lambda)^2 &= \frac{1}{d}\mathbb{E}\Big[\|\widetilde{y}_\lambda\|_2^2\Big] \\ &= \frac{1}{d}\mathbb{E}\big[c_x(\lambda)^2\alpha(\lambda)^2\|x_0\|_2^2 + c_\epsilon(\lambda)^2\sigma(\lambda)^2\|\epsilon\|_2^2 + 2c_x(\lambda)c_\epsilon(\lambda)\alpha(\lambda)\sigma(\lambda)\langle x_0, \epsilon\rangle\big] \\ &= \frac{c_x(\lambda)^2\alpha(\lambda)^2}{d}\mathbb{E}\|x_0\|_2^2 + \frac{c_\epsilon(\lambda)^2\sigma(\lambda)^2}{d}\mathbb{E}\|\epsilon\|_2^2 + \frac{2c_x(\lambda)c_\epsilon(\lambda)\alpha(\lambda)\sigma(\lambda)}{d}\mathbb{E}\langle x_0, \epsilon\rangle. \end{aligned} \tag{128}$$

Because x_0 and ϵ are independent and $\mathbb{E}[\epsilon] = 0$, the cross moment vanishes:

$$\begin{aligned} \mathbb{E}\langle x_0, \epsilon\rangle &= \mathbb{E}_{x_0}\big[\mathbb{E}_\epsilon[x_0^\top \epsilon \mid x_0]\big] \\ &= \mathbb{E}_{x_0}\big[x_0^\top \mathbb{E}_\epsilon[\epsilon \mid x_0]\big] \\ &= \mathbb{E}_{x_0}\big[x_0^\top \mathbb{E}[\epsilon]\big] = 0. \end{aligned} \tag{129}$$

Let $\mu_x = \mathbb{E}[x_0]$ and $\Sigma_x = \mathrm{Cov}(x_0)$. The second moment of x_0 satisfies

$$\begin{aligned} \mathbb{E}\|x_0\|_2^2 &= \mathbb{E}\|x_0 - \mu_x + \mu_x\|_2^2 \\ &= \mathbb{E}\|x_0 - \mu_x\|_2^2 + 2\,\langle \mathbb{E}[x_0 - \mu_x], \mu_x\rangle + \|\mu_x\|_2^2 \\ &= \mathrm{Tr}(\Sigma_x) + \|\mu_x\|_2^2. \end{aligned} \tag{130}$$

Consequently,

$$q_x = \frac{1}{d}\mathbb{E}\|x_0\|_2^2 = \frac{\mathrm{Tr}(\Sigma_x) + \|\mu_x\|_2^2}{d}, \tag{131}$$

so that $\mathbb{E}\|x_0\|_2^2 = dq_x$. Moreover, since $\epsilon \sim \mathcal{N}(0, I)$,

$$\begin{aligned}\mathbb{E}\|\epsilon\|_2^2 &= \mathbb{E}\left[\sum_{i=1}^{d} \epsilon_i^2\right] \\ &= \sum_{i=1}^{d} \mathbb{E}[\epsilon_i^2] = d.\end{aligned} \tag{132}$$

Substituting these identities and Eq. (129) into Eq. (128) gives

$$\begin{aligned}\omega_y(\lambda)^2 &= \frac{c_x(\lambda)^2\alpha(\lambda)^2}{d}\, dq_x + \frac{c_\epsilon(\lambda)^2\sigma(\lambda)^2}{d}\, d + \frac{2c_x(\lambda)c_\epsilon(\lambda)\alpha(\lambda)\sigma(\lambda)}{d}\cdot 0 \\ &= c_x(\lambda)^2\alpha(\lambda)^2 q_x + c_\epsilon(\lambda)^2\sigma(\lambda)^2.\end{aligned} \tag{133}$$

Since $\omega_y(\lambda)$ is a root-mean-square amplitude, it is given by the nonnegative square root:

$$\omega_y(\lambda) = \sqrt{c_x(\lambda)^2\alpha(\lambda)^2 q_x + c_\epsilon(\lambda)^2\sigma(\lambda)^2}, \tag{134}$$

which proves Eq. (123).

By the ERD weighting rule Eq. (125), substituting $(c_x, c_\epsilon) = (1, 0)$, $(0, 1)$, $(-1, 1)$, and $(-\sigma(\lambda), \alpha(\lambda))$ for the x_0-, ϵ-, u-, and v-prediction targets, respectively, and using $\alpha(\lambda), \sigma(\lambda) \geq 0$, yields Eq. (126). Finally, because q_x is independent of λ, the factors $\sqrt{q_x}$ and $\sqrt{q_x + 1}$ may be absorbed into the target-specific normalization constants. Under $q_x = 1$, the four weights therefore reduce, up to λ-independent normalization, to $\alpha(\lambda)$, $\sigma(\lambda)$, $\sqrt{\alpha(\lambda)^2 + \sigma(\lambda)^2}$, and $\alpha(\lambda)\sigma(\lambda)$, respectively. ■

Remark E.2 (Reduction from allocation to weighting). When the sampling density $\rho(\lambda)$ and the noise schedule are fixed,

$$M_y(\lambda) = \rho(\lambda) w_y(\lambda) \left|\dot{\lambda}_{t(\lambda)}\right|, \tag{135}$$

so the target-dependent component of the effective allocation is determined by $w_y(\lambda)$. The ERD rule therefore modifies the relative training allocation through the target-adaptive loss weight.

Remark E.3 (Interpretation). The effective-amplitude score $\omega_y(\lambda)$ is a tractable component-wise proxy rather than the exact posterior-recoverable signal $\mathbb{E}[y_\lambda \mid x_\lambda]$. It measures the RMS amplitude contributed by the two independent source channels after scaling by the amplitudes with which those channels appear in the corrupted input. Consequently, the score is invariant to sign flips of the target coefficients and does not collapse under cancellation between oppositely signed target channels, as can occur for velocity prediction.

F Toy Experiments on 2D GMM Dataset

To empirically validate our theoretical findings on Representation Degradation, NTK spectral decay, and gradient contamination, we design controlled experiments on a 2D Gaussian Mixture Model (GMM). This simplified, structured setting allows us to visually and quantitatively analyze representation dynamics, empirical NTK, and phase space trajectories across noise levels, avoiding the confounding factors of complex architectures.

F.1 Experimental Setup

Dataset Construction. We synthesize a 2D GMM with 4 symmetric clusters at $\mu \in (\pm 2.0, \pm 2.0)$ as initial data distribution $q(x_0)$. Samples are generated by uniformly selecting a center and adding Gaussian noise ($\sigma_0 = 0.3$), yielding topologically distinct clusters with intra-cluster variance.

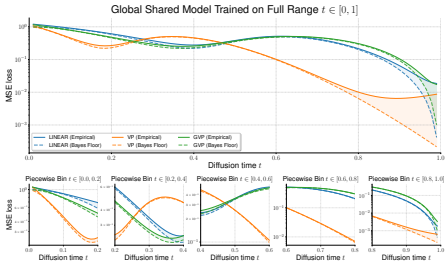
Diffusion Processes and Schedules. For clean data x_0, the perturbed data marginal at continuous time $t \sim \mathcal{U}(0, 1)$ is $x_t = \alpha_t x_0 + \sigma_t \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. To demonstrate the universality of our findings across diverse Signal-to-Noise Ratio (SNR) trajectories Kingma et al. (2021); Kingma & Gao (2023), we evaluate three schedule interpolants:

- *Linear (Flow Matching)* Lipman et al. (2022): $\alpha_t = 1 - t$, $\sigma_t = t$.
- *Variance Preserving (VP)* Song et al. (2020b): Standard Score SDE schedule mapped to $t \in [0, 1]$, with a linear noise schedule $\beta(t) = \beta_{\text{min}} + t(\beta_{\text{max}} - \beta_{\text{min}})$, where $\beta_{\text{min}} = 0.1$ and $\beta_{\text{max}} = 20.0$.
- *Generalized Variance Preserving (GVP)* Ma et al. (2024): A spherical interpolant defined by $\alpha_t = \cos(t\pi/2)$ and $\sigma_t = \sin(t\pi/2)$.

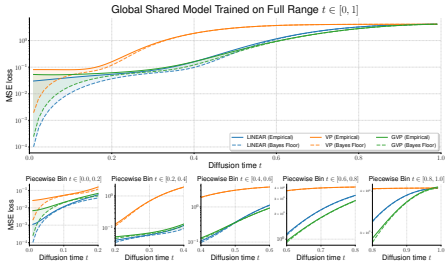
Prediction Target Formulations. To verify that gradient contamination and degradation are universal regardless of output parameterization, we evaluate four prediction targets. The network $f_\theta(x_t, t)$ predicts either the noise ϵ (standard DDPM Ho et al. (2020)), clean data x_0, velocity Salimans & Ho (2022) $v_t = \alpha_t\epsilon - \sigma_t x_0$, or flow matching vector field Lipman et al. (2022) $u_t = \epsilon - x_0$.

Network Architecture. The prediction network is a time-conditioned MLP mapping 2D inputs (x_t, t) to 2D outputs. Continuous time t is projected into a 64-dimensional sinusoidal positional embedding and concatenated with x_t. The feature extractor is a 3-layer MLP (hidden dimension $m = 256$, SiLU activations), followed by a linear readout head.

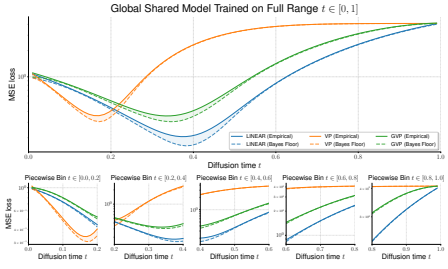
Training Details. We train separate models across the four target formulations (ϵ, x_0, v, u) using the Mean Squared Error (MSE) objective. Models are optimized with Adam (learning rate 1×10^{-3}, batch size 512) for $2,000$ iterations to ensure convergence. Alongside global models trained over $t \in [0, 1]$, we train piecewise independent models on restricted time bins (*e.g.*, $t \in [0.0, 0.2]$) to isolate local learning dynamics and establish empirical Bayes baselines.



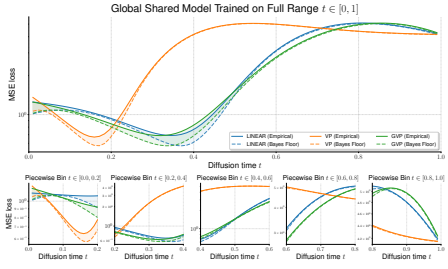
(a) Empirical Loss vs. Bayes Floor (ϵ_θ)



(b) Empirical Loss vs. Bayes Floor (x_θ)



(c) Empirical Loss vs. Bayes Floor (v_θ)



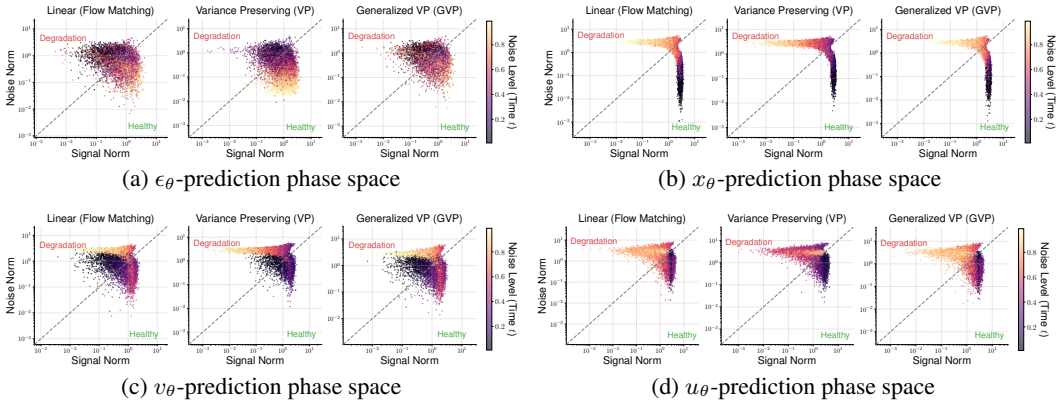
(d) Empirical Loss vs. Bayes Floor (u_θ)

Figure 7: **Empirical Loss and Analytical Bayes Floor across Different Targets.** We evaluate the globally trained shared models (top row of each subfigure) and piecewise independent models (bottom row of each subfigure) across the full diffusion time $t \in [0, 1]$.

F.2 Gradient Contamination and the Bayes Floor Gap

We first empirically validate the local ELBO decomposition and the subsequent degradation mechanisms established in Appendix C and Appendix D.

The Bayes Floor Gap and Loss Domination. Recall from Lemma C.2 that the local loss decomposes into an irreducible Bayes floor $\ell^\star_\lambda$ and an optimizable excess $\bar{\ell}_{\tau,m}(\lambda)$. To validate Corollary D.2, Figure 7 (top rows) reveals a striking loss domination phenomenon in the globally trained model: the overall optimization process is hijacked by regions exhibiting the largest absolute loss. Counterintuitively, the excess gap is minimal where the Bayes floor is large, but massive where the floor is inherently low (*e.g.*, $t \to 1$ for ϵ_θ).



(a) ϵ_θ-prediction phase space (b) x_θ-prediction phase space

(c) v_θ-prediction phase space (d) u_θ-prediction phase space

Figure 8: **Phase Space Trajectories of Gradient Contamination.** The samples universally migrate from a healthy signal-dominated regime to a severely degraded noise-dominated regime as t increases. The diagonal dashed line represents the 50% contamination boundary.

Crucially, partitioning the trajectory into independent piecewise bins (bottom rows) significantly mitigates this bottleneck, allowing local models to closely approximate the Bayes floor. Isolating optimization to narrower noise scales relieves the network from the burden of reconciling conflicting gradients. This clearly corroborates our derivation: the massive excess in theoretically simpler regimes is not a capacity deficit, but a fundamental consequence of weakened effective contraction and cross-noise interference under global parameter sharing.

Phase Space Trajectories and Gradient Contamination. To trace this optimization bottleneck, we analyze the learning signals. Proposition D.3 establishes that as recoverable signal diminishes, parameter gradients become increasingly dominated by the irreducible Bayes-noise term ξ_λ. To empirically visualize this, we map local prediction targets into a 2D phase space spanned by their recoverable signal and unrecoverable noise norms. As Figure 8 illustrates, degradation trajectories are strictly *target-dependent*. Severe contamination crossing into the noise-dominated regime occurs exclusively at *low* noise levels ($t \to 0$) for ϵ_θ, but at *high* noise levels ($t \to 1$) for x_θ. Mixed targets like v_θ and u_θ exhibit more balanced trajectories, avoiding single-pole degradation.

This provides direct empirical evidence for our framework. The network is bombarded by pure Bayes-noise gradients exactly where a target's irreducible Bayes floor is highest. Under global parameter sharing, these contaminated gradients propagate throughout the architecture, fundamentally disrupting the network's capacity to minimize excess risk even in simpler regimes.

F.3 Representation Collapse in Feature Space.

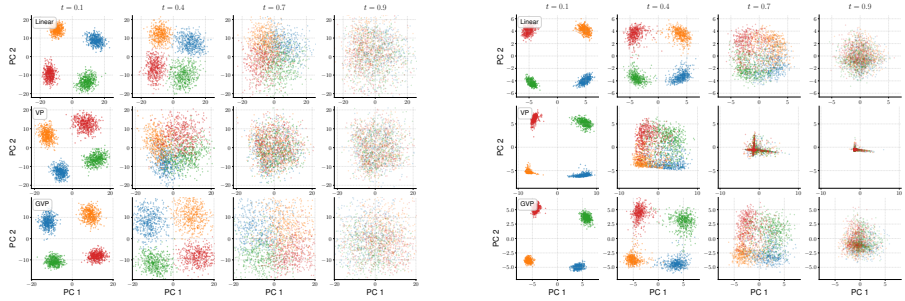
In Theorem D.4, we theoretically proved that under the fixed-noise surrogate, the shared-parameter architecture exhibits spectral decay, wherein modes associated with the fixed-noise kernel are weakly contracted and susceptible to persistent Bayes forcing. To geometrically illustrate its consequence, we track the temporal evolution of deep internal representations.

We extract the 256-dimensional hidden representations $h_{\theta,m}(x_t, t)$ immediately preceding the readout head at four uniformly sampled diffusion times $t \in \{0.1, 0.4, 0.7, 0.9\}$. To ensure a geometrically rigorous comparison, we fit a PCA solely on the healthy low-noise representations at $t = 0.1$ and project all subsequent higher-noise representations onto this exact same principal subspace.

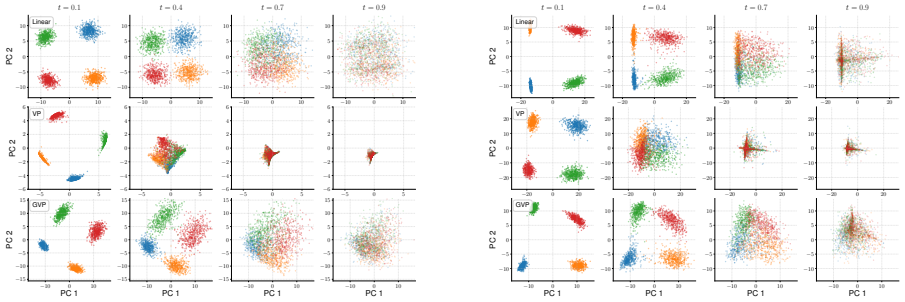
As vividly illustrated in Figure 9, the representation topology undergoes a continuous and catastrophic collapse. At $t = 0.1$, the representations perfectly preserve the structural boundaries. However, as $t \to 0.9$, these distinct clusters physically shrink and aggressively migrate towards the origin, collapsing into a tightly entangled unstructured singular manifold. This geometric visualization provides direct empirical substantiation of the spectral collapse predicted by Theorem D.4.

F.4 Neural Tangent Kernel Analysis

To unequivocally establish that the capacity degeneration formalized in Corollary D.2 and Theorem D.4 is a universal phenomenon intrinsic to global parameter sharing, rather than an artifact of a specific output parameterization, we provide the comprehensive analysis across all four prediction targets (ϵ, x_0, v, and u) in this section.



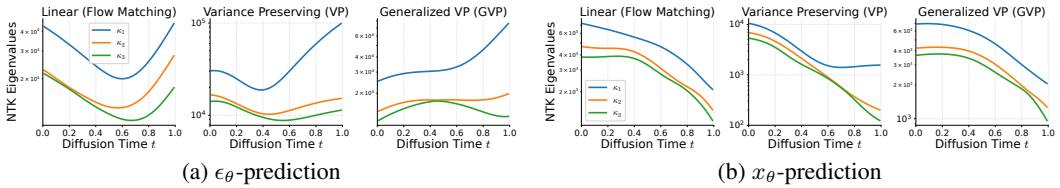
(a) ϵ_θ-prediction (b) x_θ-prediction



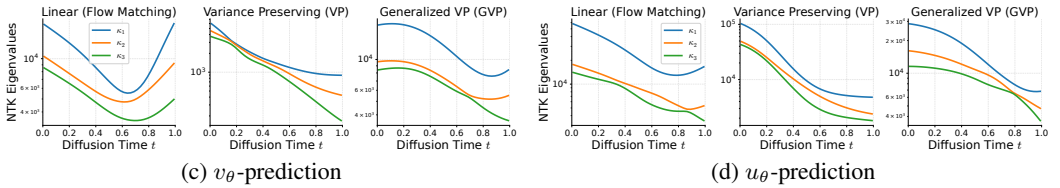
(c) v_θ-prediction (d) u_θ-prediction

Figure 9: **Representation Collapse in Global Shared Models.** The deep hidden representations $h_{\theta,m}(x_t, t)$ are tracked across four increasing diffusion times. Projected onto the PCA basis fitted at $t = 0.1$, the representations catastrophically collapse from cleanly separable data clusters into an entangled, unstructured mass as $t \to 1$.

Eigenvalue Spectral Decay. To explicitly quantify the weakened effective contraction in Theorem D.4, we compute the top eigenvalues $(\kappa_1, \kappa_2, \kappa_3)$ of the Neural Tangent Kernel over normalized time. As Figure 10 demonstrates, the spectrum exhibits severe yet highly target-dependent attenuation. For x_θ and u_θ, eigenvalues suffer a massive monotonic drop spanning several orders of magnitude. Conversely, the ϵ_θ spectrum exhibits a U-shape and increases at high noise levels to indicate retained optimization capacity. Within our theoretical framework, these vanishing eigenvalues translate directly to a diminished contraction rate. The network loses its optimization driving force in specific degraded regimes, leaving learning dynamics vulnerable to persistent Bayes forcing and mathematically confirming our geometric observations.



(a) ϵ_θ-prediction (b) x_θ-prediction

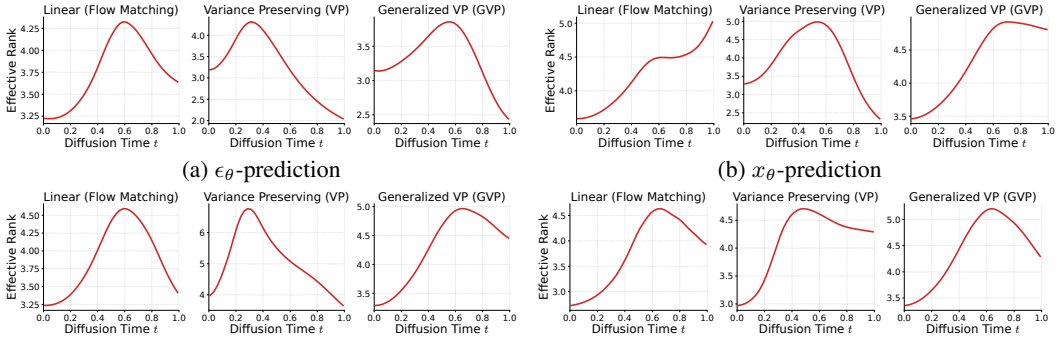


(c) v_θ-prediction (d) u_θ-prediction

Figure 10: **Eigenvalue Spectral Decay.** The aggressive attenuation of the top-3 eigenvalues is a universal bottleneck, empirically supporting the bounds in Theorem D.4.

Effective Rank Collapse. To further assess the global structural integrity of the Neural Tangent Kernel spectrum, we evaluate the effective rank based on Shannon entropy. While tracking individual top eigenvalues reveals absolute magnitude decay, the effective rank quantifies the overall dimensionality and richness of the learned functional space. As illustrated in Figure 11, the effective rank undergoes a severe plummet toward near singularity limits at the extremes of the diffusion process, particularly as $t \to 1$. This widespread rank collapse dictates that the entire eigenvalue distribution becomes heavily skewed and highly correlated. The network fundamentally loses its functional degrees of freedom,

forcing its representational manifold to collapse into a constrained low-dimensional subspace. This global spectral degeneration offers definitive mathematical confirmation of the representation collapse induced by persistent Bayes noise contamination.

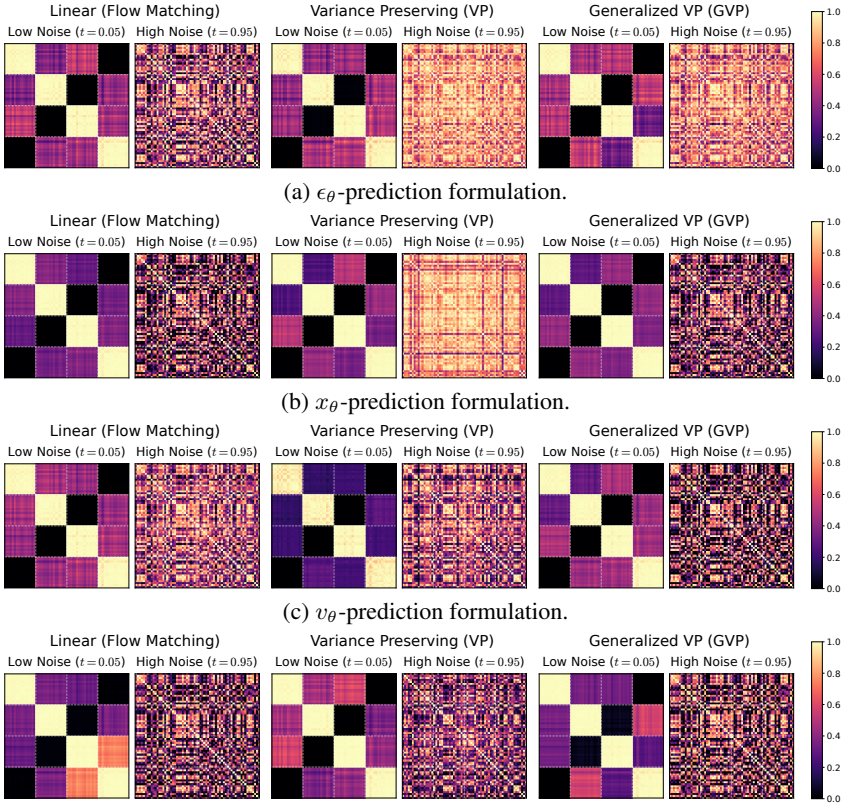


(a) ϵ_θ-prediction (b) x_θ-prediction

(c) v_θ-prediction (d) u_θ-prediction

Figure 11: **Effective Rank Collapse.** The Shannon entropy-based effective rank severely plummets to near-singularity limits as $t \to 1$.

NTK Homogenization. As a direct spatial consequence of the spectral collapse, Figure 12 displays the normalized NTK Gram matrices. Across every formulation and schedule interpolant, the network maintains a highly discriminative block-diagonal feature space at low noise levels ($t = 0.05$). In this healthy state, inputs originating from the same ground-truth cluster share strongly aligned gradients while distinct clusters remain orthogonal. The space catastrophically homogenizes into an entangled uniform matrix at high noise levels ($t = 0.95$) visually confirming the predicted capacity loss. The homogenized state reveals the network processes all inputs identically and completely loses data-conditional discrimination. Under global parameter sharing these highly correlated updates inevitably wash out the refined cluster boundaries.



(a) ϵ_θ-prediction formulation.

(b) x_θ-prediction formulation.

(c) v_θ-prediction formulation.

(d) u_θ-prediction formulation.

Figure 12: **NTK Gram Matrix Evolution.** The structural collapse from cleanly separated data clusters to an entangled homogeneous state occurs across all parameterizations.

G Experimental Supplement

G.1 Further Implementation Details.

In this section, we provide additional implementation details, including training configurations, architecture choices, and sampling procedures. A complete list of hyperparameters is provided in Table 5. Here **Convolution** represents whether to add a 3×3 convolutional block before output.

Basic setting. All experiments were conducted with 8 NVIDIA A100 GPUs (80GB) with a global batch size of 256. To accelerate training, we employ mixed-precision training (fp16). We adopt the AdamW optimizer Kingma (2014); Loshchilov & Hutter (2017) with a constant learning rate of 1×10^{-4} without weight decay.

Following common practice in diffusion-based image generation, we pre-compute latent representations from raw images using the Stable Diffusion VAE Rombach et al. (2022), and all experiments are performed in the latent space. As a result, no data augmentation is applied, which we find has negligible impact on performance, consistent with observations in EDM2 Karras et al. (2024). For image decoding, we use the `stabilityai/sd-vae-ft-ema` decoder.

Diffusion setting. We adopt a discrete-time formulation where the forward process is defined over $t \in [0, 1000)$, utilizing the cosine noise schedule following IDDPM Nichol & Dhariwal (2021). During training, to enable classifier-free guidance (CFG) at inference time, we randomly drop out the class conditioning labels with a probability of $p_{\text{uncond}} = 0.1$, replacing them with a learned null token. Following standard practice to stabilize optimization and improve sample quality, we maintain an Exponential Moving Average (EMA) of the model weights with a decay rate of 0.9999 for all evaluations. For sampling, we use the deterministic EDM Heun sampler Karras et al. (2022) with $N = 50$ integration steps.

Backbone architecture. To validate the universality of our method, we use DiT Peebles & Xie (2023) and U-ViT Bao et al. (2023) as primary backbone networks. Compared with DiT's homogeneous sequence processing, U-ViT adopts a specialized symmetric encoder–decoder architecture. This topology incorporates a designated middle block and long-range skip connections seamlessly bridging corresponding encoder and decoder layers, which significantly improves multi-scale feature interaction across depths. Following the canonical design, hierarchical encoder features are sequentially concatenated with the corresponding decoder features then projected by a dense linear layer.

G.2 Evaluation Details

We adopt the same evaluation protocol as ADM Dhariwal & Nichol (2021), and use the identical reference batches provided in their official implementation.[3] For efficiency, we enable mixed-precision (fp16) during sample generation, and empirically observe no noticeable degradation compared to full fp32 precision. Below, we briefly describe the evaluation metrics used throughout our experiments.

- *Fréchet Inception Distance (FID)* Heusel et al. (2017) quantifies the discrepancy between real and generated image distributions in a learned feature space. The features are extracted using an Inception-v3 network Szegedy et al. (2016), and the distance is computed under the assumption that both distributions follow multivariate Gaussian statistics.
- *Spatial FID (sFID)* Nash et al. (2021) extends FID by leveraging intermediate spatial feature maps of Inception-v3, allowing it to better reflect spatial consistency in generated images.
- *Inception Score (IS)* Salimans et al. (2016) is also based on the Inception-v3 classifier, but evaluates samples using the predicted class logits. Concretely, it computes the Kullback–Leibler divergence between the marginal label distribution and the conditional label distribution obtained after softmax normalization.
- *Precision and Recall* Kynkäänniemi et al. (2019) assess the quality and coverage of generated samples, respectively. Precision reflects the proportion of generated images that are visually realistic, while recall measures how well the generated distribution captures the diversity of the training data manifold.

[3] https://github.com/openai/guided-diffusion/tree/main/evaluations

Table 5: Hyperparameter setup.

Related result	Table 3						Table 4
Architecture							
Model	DiT-B/2	DiT-L/2	DiT-XL/2	U-ViT-M/2	U-ViT-L/2	U-ViT-H/2	U-ViT-H/2
Input dim.	32×32×4	32×32×4	32×32×4	32×32×4	32×32×4	32×32×4	32×32×4
Num. layers	12	24	28	17	21	29	29
Hidden dim.	768	1,024	1,152	768	1,024	1,152	1,152
Num. heads	12	16	16	12	16	16	16
Convolution	∅	∅	∅	✓	✓	✗	✗
Params (M)	130	458	675	131	287	501	501
Optimization							
Training iteration / epoch	400K	400K	400K	400K	400K	400K	800
Batch size	256	256	256	256	256	256	256
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.99, 0.99)	(0.99, 0.99)	(0.99, 0.99)	(0.99, 0.99)
Diffusion settings							
Noise schedule	cosine	cosine	cosine	cosine	cosine	cosine	cosine
Training objective	ϵ-prediction	ϵ-prediction	ϵ_θ-prediction	ϵ_θ-prediction	ϵ_θ-prediction	ϵ_θ-prediction	ϵ_θ-prediction
Sampler	EDM Heun	EDM Heun	EDM Heun	EDM Heun	EDM Heun	EDM Heun	EDM Heun
Sampling steps	50	50	50	50	50	50	50
Guidance scale	-	-	-	-	-	-	1.35

G.3 Computational Overhead Analysis

We analyze the computational overhead of ERD relative to standard diffusion training and external representation alignment methods.

ERD only reweights the training loss. For a batch of N diffusion samples, it assigns a recoverability-aware weight $\omega_y(\lambda_i)$ to each per-sample loss:

$$\mathcal{L}_{\text{ERD}} = \frac{1}{N} \sum_{i=1}^{N} \omega_y(\lambda_i) \ell_i. \tag{136}$$

All quantities required to compute $\omega_y(\lambda)$, including $\alpha(\lambda)$, $\sigma(\lambda)$, and target coefficients, are already available from the diffusion interpolation process. ERD therefore requires no additional network forward pass, feature extraction, Jacobian computation, or backward propagation. Its only extra cost is evaluating per-sample weights and multiplying them by the corresponding losses, resulting in $\mathcal{O}(N)$ complexity with respect to the batch size. For $N = 256$, this amounts to only a few hundred scalar operations per iteration and is negligible compared with diffusion training cost.

We further compare ERD with REPA Yu et al. (2024), which uses a pretrained visual encoder for representation supervision. For REPA-Offline, we report one-time DINOv2-B/14 feature extraction cost on ImageNet-1K. For REPA-Online, we report cumulative feature extraction cost over training iterations. All results in Table 6 assume batch size 256 and a 400K-iteration training budget.

Table 6: Comparison of additional computational overhead.

Method	Additional Operation	Per Step	Total Extra Compute
Baseline	None	0	0
ERD	Scalar loss weighting	$\approx 2.0 \times 10^{-6}$ GFLOPs	$\approx$ 0.8–1.0 GFLOPs
REPA (Offline)	DINOv2-B/14 extraction	–	59.35 PFLOPs
REPA (Online)	DINOv2-B/14 forward	11.86 TFLOPs	4.74 EFLOPs

For fairness, the reported REPA costs include only the DINOv2-B/14 forward pass and exclude data loading, preprocessing, and feature storage. As shown in Table 6, ERD introduces virtually no additional computational overhead while remaining lightweight and architecture-agnostic.

G.4 Detailed Quantitative Results

We provide a detailed quantitative evaluation of ERD from following perspectives: performance across architectures and model scales, training efficiency, comparison with existing loss-weighting methods, robustness to latent-space statistics, complementarity with external representation alignment, and performance under classifier-free guidance. Unless otherwise specified, all results are evaluated on ImageNet 256×256 without classifier-free guidance.

Performance Across Architectures and Model Scales. Tables 7 and 8 present the quantitative comparisons on DiT and U-ViT backbones, respectively. ERD consistently improves generative performance across different architectures and model scales, outperforming the corresponding baselines under the same training budget. The gains are observed on both DiT and U-ViT families, demonstrating the scalability and architecture-agnostic effectiveness of ERD.

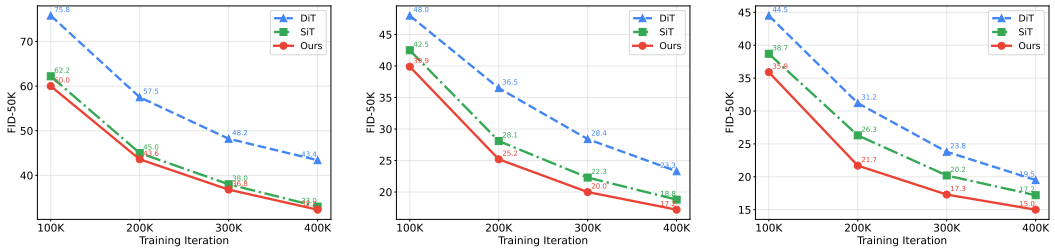
Table 7: Detailed results on DiT models of different sizes, without classifier-free guidance.

Model	#Params	Iter.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
SiT-B/2 Ma et al. (2024)	130M	400K	33.0	6.5	43.7	0.53	0.63
DiT-B/2 Peebles & Xie (2023)	130M	400K	43.4	-	-	-	-
+ Ours	130M	100K	60.0	8.4	21.8	0.40	0.59
+ Ours	130M	200K	43.6	7.6	32.6	0.47	0.64
+ Ours	130M	300K	36.8	7.3	40.5	0.51	0.66
+ Ours	130M	400K	32.3	7.0	46.6	0.53	0.66
SiT-L/2 Ma et al. (2024)	458M	400K	18.8	5.3	72.0	0.64	0.64
DiT-L/2 Peebles & Xie (2023)	458M	400K	23.3	-	-	-	-
+ Ours	458M	100K	39.9	7.7	33.9	0.50	0.64
+ Ours	458M	200K	25.2	7.0	55.8	0.58	0.65
+ Ours	458M	300K	20.0	6.8	69.6	0.61	0.66
+ Ours	458M	400K	17.2	6.5	78.9	0.62	0.67
SiT-XL/2 Ma et al. (2024)	675M	400K	17.2	-	-	-	-
DiT-XL/2 Peebles & Xie (2023)	675M	400K	19.5	-	-	-	-
+ Ours	675M	100K	35.9	7.5	37.4	0.53	0.64
+ Ours	675M	200K	21.7	6.6	62.6	0.60	0.65
+ Ours	675M	300K	17.3	6.4	77.4	0.62	0.67
+ Ours	675M	400K	15.0	6.4	86.4	0.64	0.66

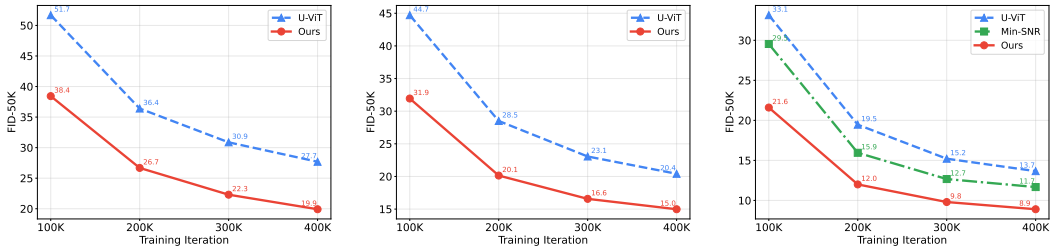
Table 8: Detailed results on U-ViT models of different sizes, without classifier-free guidance.

Model	#Params	Iter.	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
U-ViT-M/2	131M	400K	27.7	7.0	50.9	0.55	0.66
+ Ours	131M	100K	38.4	8.8	37.0	0.52	0.63
+ Ours	131M	200K	26.7	8.1	54.3	0.57	0.65
+ Ours	131M	300K	22.3	7.5	64.1	0.59	0.66
+ Ours	131M	400K	19.9	7.0	70.8	0.60	0.66
U-ViT-L/2	287M	400K	20.4	6.6	68.0	0.60	0.67
+ Ours	287M	100K	31.9	7.4	44.3	0.56	0.63
+ Ours	287M	200K	20.1	7.2	67.3	0.62	0.64
+ Ours	287M	300K	16.6	6.8	79.2	0.63	0.66
+ Ours	287M	400K	15.0	6.7	86.3	0.64	0.65
Min-SNR	501M	400K	11.7	6.6	99.7	0.66	0.65
U-ViT-H/2	501M	400K	13.7	5.6	87.9	0.65	0.65
+ Ours	501M	100K	21.6	6.3	60.6	0.63	0.60
+ Ours	501M	200K	12.0	5.6	92.9	0.67	0.64
+ Ours	501M	300K	9.8	5.5	106.8	0.68	0.64
+ Ours	501M	400K	8.9	5.4	114.3	0.68	0.66

Training Efficiency and Convergence. To evaluate training efficiency, we report the FID convergence curves of different DiT and U-ViT backbones in Figure 13. ERD reaches the 400K-iteration baseline performance with substantially fewer optimization steps. U-ViT-M/2 and U-ViT-L/2 already outperform their respective 400K baselines at 200K iterations, while U-ViT-H/2 reaches comparable performance to Min-SNR using half the training budget. Similarly, DiT-B/2 matches its 400K baseline at 200K iterations, and the larger DiT-L/2 and DiT-XL/2 models surpass their baselines by 300K iterations. These results show that ERD consistently improves training efficiency across architectures and model scales.



(a) DiT-B/2 (b) DiT-L/2 (c) DiT-XL/2



(d) U-ViT-M/2 (e) U-ViT-L/2 (f) U-ViT-H/2

Figure 13: Convergence curves on ImageNet 256 × 256. Intermediate FID values of DiT are approximated from the training curves in Ma et al. (2024); Jiang et al. (2025) for visualization only.

Comparison with Existing Loss-Weighting Methods. We further compare ERD with existing noise-dependent loss-weighting methods under the same 100K-iteration setting.

As shown in Table 9, ERD consistently achieves the lowest FID across all evaluated DiT and U-ViT configurations. The improvement is observed across different architectures and model scales, indicating that the benefit of recoverability-aware weighting is not restricted to a particular model configuration. In contrast to Min-SNR, which allocates optimization effort according to truncated marginal SNR, ERD determines its weights from the recoverability of the prediction target.

Table 9: Comparison with different methods at 100K training iters under same setting. FID-50K evaluated on ImageNet 256 × 256, no CFG.

Model	Baseline	SiT	Min-SNR	ERD
DiT-B/2	75.8	62.2	62.57	**60.0**
DiT-L/2	48.0	42.5	43.44	**39.9**
DiT-XL/2	44.5	38.7	37.82	**35.9**
U-ViT-H/2	33.1	–	29.5	**21.6**

Robustness to Latent-Space Statistics. Proposition E.1 derives the ERD weight for a general data distribution using the average per-dimension second moment $q_x = d^{-1}\mathbb{E}[\|x_0\|_2^2]$, without assuming zero mean or identity covariance. In our main experiments, we set $q_x = 1$, motivated by the standard latent rescaling factor 0.18215 used for pretrained VAE latents.[4] Since this rescaling only calibrates the overall latent magnitude, we additionally estimate $\hat{q}_x$ from the training latents and compare the simplified and generalized weights under x_0-prediction:

$$\omega_{x_0,\text{simple}}(\lambda) = \alpha(\lambda), \quad \omega_{x_0,\text{generalized}}(\lambda) = \alpha(\lambda)\sqrt{\hat{q}_x}. \tag{137}$$

As shown in Table 10, the two variants perform nearly identically across all metrics, indicating that the approximation $q_x = 1$ has negligible practical impact in the normalized latent space.

Table 10: Comparison of the simplified and generalized ERD weights under the DiT-B x_0-prediction.

Method	ERD Weight	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Baseline	None	47.76	8.34	33.55	0.44	0.62
Simple ERD	$\omega_{x_0}(\lambda) = \alpha(\lambda)$	36.36	**7.18**	44.78	**0.50**	**0.64**
Generalized ERD	$\omega_{x_0}(\lambda) = \alpha(\lambda)\sqrt{\hat{q}_x}$	**36.34**	7.19	**44.82**	0.49	**0.64**

[4] https://github.com/huggingface/diffusers/issues/437

Complementarity with External Representation Alignment. ERD and REPA improve representation optimization through complementary mechanisms. REPA introduces external semantic supervision using a pretrained visual encoder, whereas ERD reallocates optimization effort across noise levels without relying on an external teacher. We therefore evaluate their combination using

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{ERD}} + \beta \mathcal{L}_{\text{REPA}}, \tag{138}$$

where β controls the strength of representation alignment. Following the optimal REPA configuration, we align the eighth transformer layer and use $\beta = 0.5$ for REPA alone. For ERD+REPA, we only adjust β to 0.1, since ERD weighting reduces the overall scale of the diffusion loss and consequently changes the relative contribution of the alignment objective.

Table 11: **Table.** FID convergence of ERD and REPA under the same DiT training configuration.

Method	100K	200K	300K	400K
Baseline	75.80	57.50	48.20	43.40
ERD	60.00	43.60	36.80	32.30
REPA	55.65	42.68	37.08	34.47
ERD + REPA	**52.58**	**36.62**	**31.20**	**28.87**

As shown in Table 11, combining ERD with REPA consistently improves FID throughout training. The joint method outperforms either component alone, demonstrating that recoverability-aware loss allocation and external representation supervision are complementary, although their effects partially overlap at later stages of training.

Classifier-Free Guidance and Guidance Intervals. Finally, we evaluate the U-ViT-H/2 model trained for 4M iterations under classifier-free guidance. Table 12 reports the results obtained with different guidance scales. Increasing the guidance scale from $w = 1.30$ to $w = 1.50$ consistently improves FID, sFID, and IS, with the best FID of 1.89 obtained at $w = 1.50$.

We additionally apply guidance intervals following Kynkäänniemi et al. (2024). As shown in Table 13, restricting classifier-free guidance to an appropriate portion of the sampling trajectory further improves generation quality. The best configuration uses the interval $[0, 0.8]$ and $w = 2.10$, achieving an FID of 1.45.

Table 12: Results with different classifier-free guidance scales w.

Model	#Params	Iter.	w	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
DiT-XL/2 Peebles & Xie (2023)	675M	7M	1.500	2.27	4.60	278.24	0.83	0.57
SiT-XL/2 Ma et al. (2024)	675M	7M	1.500	2.06	4.50	270.3	0.82	0.59
U-ViT-H/2 Bao et al. (2023)	501M	2M	1.500	2.29	5.68	263.88	0.82	0.57
Ours	501M	4M	1.300	2.08	4.62	255.6	0.79	0.64
Ours	501M	4M	1.350	2.00	4.57	266.9	0.80	0.63
Ours	501M	4M	1.400	1.94	4.53	273.8	0.81	0.62
Ours	501M	4M	1.450	1.91	4.50	279.3	0.82	0.61
Ours	501M	4M	1.500	**1.89**	**4.48**	**282.5**	**0.82**	0.60

Table 13: Results with different classifier-free guidance intervals Kynkäänniemi et al. (2024).

Model	#Params	Iter.	Interval	w	FID↓	sFID↓	IS↑	Prec.↑	Rec.↑
Ours	501M	4M	$[0, 0.9]$	2.10	1.71	4.43	326.8	0.83	0.59
Ours	501M	4M	$[0, 0.85]$	2.10	1.56	4.47	319.5	0.82	0.60
Ours	501M	4M	$[0, 0.8]$	2.00	1.48	4.54	304.8	0.81	0.63
Ours	501M	4M	$[0, 0.8]$	2.10	**1.45**	**4.51**	312.4	0.81	0.62
Ours	501M	4M	$[0, 0.8]$	2.20	1.49	4.53	311.7	0.82	0.61
Ours	501M	4M	$[0, 0.75]$	2.10	1.52	4.66	300.3	0.80	0.64
Ours	501M	4M	$[0, 0.7]$	2.10	1.60	4.83	289.6	0.79	0.65

G.5 Qualitative Results



Figure 14: We use classifier-free guidance with $w = 4.0$. Class label = "loggerhead sea turtle" (33).



Figure 15: We use classifier-free guidance with $w = 4.0$. Class label = "macaw" (88).



Figure 16: We use classifier-free guidance with $w = 4.0$. Class label = "sulphur-crested cockatoo" (89).



Figure 17: We use classifier-free guidance with $w = 4.0$. Class label = "golden retriever" (207).



Figure 18: We use classifier-free guidance with $w = 4.0$. Class label = "husky" (250).



Figure 19: We use classifier-free guidance with $w = 4.0$. Class label = "arctic wolf" (270).



Figure 20: We use classifier-free guidance with $w = 4.0$. Class label = "arctic fox" (279).



Figure 21: We use classifier-free guidance with $w = 4.0$. Class label = "lion" (291).



Figure 22: We use classifier-free guidance with $w = 4.0$. Class label = "otter" (360).



Figure 23: We use classifier-free guidance with $w = 4.0$. Class label = "red panda" (387).



Figure 24: We use classifier-free guidance with $w = 4.0$. Class label = "panda" (388).



Figure 25: We use classifier-free guidance with $w = 4.0$. Class label = "acoustic guitar" (402).



Figure 26: We use classifier-free guidance with $w = 4.0$. Class label = "balloon" (417).



Figure 27: We use classifier-free guidance with $w = 4.0$. Class label = "baseball" (429).



Figure 28: We use classifier-free guidance with $w = 4.0$. Class label = "dog sled" (537).



Figure 29: We use classifier-free guidance with $w = 4.0$. Class label = "fire truck" (555).



Figure 30: We use classifier-free guidance with $w = 4.0$. Class label = "laptop" (620).



Figure 31: We use classifier-free guidance with $w = 4.0$. Class label = "space shuttle" (812).



Figure 32: We use classifier-free guidance with $w = 4.0$. Class label = "ice cream" (928).



Figure 33: We use classifier-free guidance with $w = 4.0$. Class label = "cheeseburger" (933).

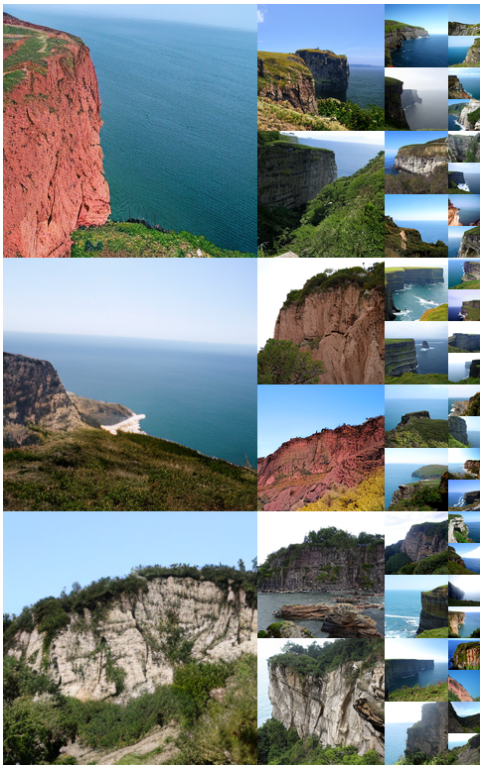


Figure 34: We use classifier-free guidance with $w = 4.0$. Class label = "cliff drop-off" (972).



Figure 35: We use classifier-free guidance with $w = 4.0$. Class label = "coral reef" (973).



Figure 36: We use classifier-free guidance with $w = 4.0$. Class label = "lake shore" (975).



Figure 37: We use classifier-free guidance with $w = 4.0$. Class label = "volcano" (980).