Where Should Diffusion Enter a Language Model? Geometry-Guided Hidden-State Replacement

Injin Kong[1] **Hyoungjoon Lee**[2] **Yohan Jo**[1,†]
mtkong77@snu.ac.kr hjoon721@snu.ac.kr yohan.jo@snu.ac.kr

[1]Graduate School of Data Science, Seoul National University, Seoul, Republic of Korea
[2]Department of Biosystems & Biomaterials Science and Engineering, Seoul National University, Seoul, Republic of Korea

Abstract

Continuous diffusion language models lag behind autoregressive transformers, partly because diffusion is applied in spaces poorly suited to language denoising and token recovery. We propose DiHAL, a geometry-guided diffusion–transformer hybrid that asks where diffusion should enter a pretrained transformer. DiHAL scores layers with geometry-based proxies, selects a diffusion-friendly hidden-state interface, and replaces the lower transformer prefix with a diffusion bridge while retaining the upper layers and original LM head. By reconstructing the selected-layer hidden state rather than tokens, DiHAL avoids direct continuous-to-discrete recovery. Experiments on 8B-scale backbones show that the geometry score predicts effective shallow insertion layers under a fixed bridge-training protocol and that hidden-state recovery improves over continuous diffusion baselines in a diagnostic comparison matching the diffusion/recovery training budget. These results suggest that hidden–state geometry helps identify where diffusion–based replacement is feasible inside pretrained language models.

1. Introduction

Large language models have achieved remarkable progress across a wide range of language generation tasks, but this progress has come with increasing size and computational cost (Brown et al., 2020; Hoffmann et al., 2022; Yang et al., 2025). Diffusion models offer a different generative paradigm based on iterative denoising and have become a dominant approach in image generation (Song et al., 2021). Their success has motivated growing interest in diffusion-based language generation (Li et al., 2022; Strudel et al., 2023; Nie et al., 2025). However, transferring diffusion from images to text is difficult because text generation must ultimately handle discrete tokens.

A natural response is to adapt diffusion to the discreteness of text. Prior work has explored discrete token corruption, masked diffusion, continuous-to-discrete recovery, and continuous diffusion over token embeddings, self-conditioned embeddings, or learned text latents (Li et al., 2022; Strudel et al., 2023; Lovelace et al., 2023; Gong et al., 2023; Zhang et al., 2025). Despite these efforts, diffusion-based language models still lag behind autoregressive Transformers, particularly in continuous diffusion settings (Jo and Hwang, 2026). A common explanation is that denoised continuous vectors must eventually be mapped back to discrete tokens, so small errors in representation space can change the recovered token (Li et al., 2022).

Why does this gap remain? We start from a complementary hypothesis: discreteness is important but may not fully explain the gap. Transformer language models also use discrete tokens, yet most computation occurs in continuous hidden states later mapped to vocabulary logits (Vaswani et al., 2017). This suggests that the difficulty may arise not from continuity itself, but from applying diffusion in continuous spaces with unsuitable geometry.

If the choice of continuous space matters, then the central question becomes: what makes a representation space suitable for diffusion? We call such a space *diffusion-friendly*: a space that is easy to denoise, stable under imperfect score

[†]Corresponding author: Yohan Jo <yohan.jo@snu.ac.kr>.
Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

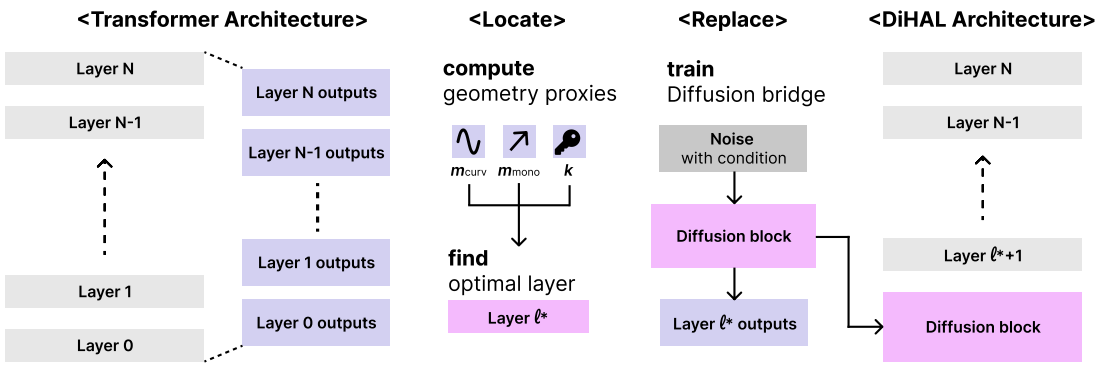

Figure 1. Locate-and-Replace framework. Layer-wise geometric proxies score transformer layers, select an insertion point, and guide replacement with a diffusion block.

estimates, and simple enough for diffusion to learn. We later motivate these requirements using tools from Langevin dynamics and concentration theory (Villani, 2009; Bakry et al., 2014; Ledoux, 2001).

Where can such a space be found in a language model? A pretrained Transformer already contains many continuous hidden spaces between the token embedding layer and the LM head. These hidden states are not decoded directly into tokens; they are consumed by the remaining Transformer layers before the LM head produces the final token distribution (Vaswani et al., 2017). Diffusion at an internal layer can therefore target hidden-state recovery rather than direct token recovery (Lovelace et al., 2023; Rombach et al., 2022). Since hidden-state geometry varies across depth, we ask: **which transformer layer provides the most diffusion-friendly representation space?**

To answer this question, we propose **DiHAL** (**Di**ffusion-Transformer **H**ybrid **A**rchitecture for **L**anguage Generation), a hybrid architecture based on a *Locate-and-Replace* strategy. As illustrated in Figure 1, DiHAL locates diffusion-friendly layers using geometry-based criteria, then replaces the lower transformer layers with a *diffusion bridge* that reconstructs the selected-layer hidden state while retaining the upper layers and original LM head for token prediction. This reduces continuous-to-discrete recovery error and reframes continuous diffusion for language as a problem of choosing the right internal representation space for denoising. Our contributions are threefold.

- We formulate diffusion insertion in pretrained transformer language models as a **geometry-guided interface-selection problem** and propose practical layer-wise proxies—local compactness, global stiffness, and effective rank—for identifying diffusion-friendly hidden spaces.
- We introduce a fixed geometry score that narrows the search for effective insertion layers without exhaustive layer-wise bridge training and correlates strongly with hidden-state reconstruction quality under a one-epoch bridge-training protocol across 8B-scale backbones.
- We introduce **DiHAL**, a Locate-and-Replace hybrid that replaces lower transformer layers with a conditional diffusion bridge and reuses the upper layers and LM head. Under a diagnostic diffusion/recovery budget, DiHAL shows that hidden-state recovery can improve generative perplexity and diversity over embedding-, latent-, and continuous-to-discrete interfaces.

2. Background

Transformer language models take discrete tokens as input, but most computation occurs in continuous hidden spaces. Given $x_{1:T}$, an autoregressive model factorizes $p(x_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$. Each token is mapped to an embedding $h_t^{(0)} = E(x_t) + p_t$, hidden states are updated as $H^{(\ell)} = F_\ell(H^{(\ell-1)})$, and the final state is projected to vocabulary logits $z_t = W_{\mathrm{LM}} h_t^{(L)} + b$. Thus, discreteness appears at the input and output interfaces, while intermediate computation is continuous (Vaswani et al., 2017).

Diffusion models generate samples by gradually adding noise to data and then learning to reverse this noising process (Song et al., 2021). In continuous space, this process can be written as a stochastic differential equation $dX_t = f(X_t, t)\,dt + g(t)\,dW_t$, where X_t is the noisy representation at time t and W_t is Brownian motion. The reverse

process depends on the score $\nabla_x \log p_t(x)$, which is approximated by a neural network. Applying this idea to language requires choosing what representation the diffusion model should denoise. Discrete diffusion models corrupt tokens directly (Austin et al., 2021; Hoogeboom et al., 2021), while continuous diffusion language models denoise token embeddings or learned latent vectors (Li et al., 2022; Strudel et al., 2023; Lovelace et al., 2023).

Continuous token-level diffusion suffers from recovery errors: small denoising deviations can flip the recovered token (Zhang et al., 2025; Shen et al., 2026). Learned latent diffusion reduces this issue but still requires an interface for converting latents to text. We instead target internal transformer hidden states, where recovery becomes hidden-state reconstruction rather than direct token decoding.

3. Method

DiHAL is a diffusion–transformer hybrid architecture that replaces part of a pretrained transformer, rather than a standalone diffusion language model. Figure 1 illustrates our Locate-and-Replace procedure. This section develops DiHAL in three steps: we first motivate diffusion-friendly representations using geometric principles from Langevin dynamics and concentration theory, then instantiate them as layer-wise proxies for locating a suitable hidden-state interface, and finally replace the lower transformer layers with a conditional diffusion bridge while retaining the upper layers and original LM head. Rather than modeling token probabilities or recovering discrete tokens directly, the bridge reconstructs an internal boundary representation that the retained upper layers can already process.

3.1. Geometric Principles for Diffusion-Friendly Layer Selection

We now make the notion of a *diffusion-friendly* representation space more concrete. Intuitively, a good diffusion space should satisfy three properties: denoising should contract quickly toward the target representation distribution, remain stable under score-estimation error, and have low effective complexity, meaning that variation is concentrated in relatively few active directions.

We formalize the first two properties through overdamped Langevin dynamics, an idealized setting with clean convergence and stability guarantees. The third property is captured by effective rank, which measures active variance directions. The theorem settings in this section are idealized: they motivate geometric surrogates, not assumptions that transformer hidden states exactly satisfy them.

Throughout this section, W_2 denotes the 2-Wasserstein distance and $\mathcal{P}_2(\mathbb{R}^d)$ denotes probability measures with finite second moment. For a density p, define $U(x) = -\log p(x)$. We interpret strong convexity of U as a curvature-like restoring force toward high-density regions. Theorem 3.1 introduces the curvature parameter m and shows that larger m yields faster convergence to the target distribution.

Theorem 3.1 (Wasserstein contraction under strong log-concavity (Villani, 2009; Bakry et al., 2014))**.** *Let $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ be a probability measure with density p, and define $U(x) := -\log p(x)$. Assume that $U \in C^2(\mathbb{R}^d)$, U is m-strongly convex, i.e. $\nabla^2 U(x) \succeq mI$ for all $x \in \mathbb{R}^d$, for some $m > 0$, and that ∇U is globally Lipschitz. Let $(X_t)_{t\geq 0}$ satisfy the overdamped Langevin stochastic differential equation (SDE) $dX_t = -\nabla U(X_t)\,dt + \sqrt{2}\,dW_t$, where W_t denotes Brownian motion.*

Then μ is an invariant distribution of (X_t), and for every initial law $\nu_0 \in \mathcal{P}_2(\mathbb{R}^d)$,

$$W_2(\nu_t, \mu) \leq e^{-mt} W_2(\nu_0, \mu), \qquad \nu_t := \mathcal{L}(X_t),$$

where $\mathcal{L}(X_t)$ denotes the distribution of X_t. The invariant distribution μ is unique in $\mathcal{P}_2(\mathbb{R}^d)$.

Theorem 3.1 gives the first criterion. If the curvature parameter m is large, the distance to the target distribution shrinks as e^{-mt}. Thus, larger m means faster contraction, which is desirable for diffusion because denoising should quickly return noisy samples to the data distribution.

Fast contraction alone is not enough. In practice, the score is unknown and estimated by a neural network. Theorem 3.2 gives the second criterion. If the score error is at most ε, the induced distributional error is bounded by ε/m. Thus, larger m corresponds to stability under imperfect score estimation.

Theorem 3.2 (Stability of an invariant measure under score perturbation)**.** *Let $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ have density p and define $U(x) := -\log p(x)$. Assume that $U \in C^2(\mathbb{R}^d)$ is m-strongly convex, and that ∇U is globally Lipschitz. Let*

$s(x) := \nabla \log p(x) = -\nabla U(x)$, and let $\hat{s} : \mathbb{R}^d \to \mathbb{R}^d$ be globally Lipschitz and satisfy $\sup_{x\in\mathbb{R}^d} \|\hat{s}(x) - s(x)\| \le \varepsilon$. Consider the two SDEs $dX_t = s(X_t)\,dt + \sqrt{2}\,dW_t$ and $d\hat{X}_t = \hat{s}(\hat{X}_t)\,dt + \sqrt{2}\,dW_t$. Assume that the second SDE admits an invariant distribution $\hat{\mu}$.

Then $\hat{\mu} \in \mathcal{P}_2(\mathbb{R}^d)$ and

$$W_2(\hat{\mu}, \mu) \le \frac{\varepsilon}{m}.$$

Together, Theorems 3.1 and 3.2 suggest that curvature is a useful proxy for convergence speed and stability under score-estimation error. However, curvature alone does not capture whether a representation is easy to model: variation may still spread across many directions. We therefore use effective rank as a proxy for dimensionality. If activations concentrate near a low-dimensional manifold, diffusion needs to model a few meaningful directions. Here, $\mathrm{tr}(\Sigma)$ is total variance and $\|\Sigma\|$ is the largest covariance eigenvalue, so $r_{\mathrm{eff}}(\Sigma) = \mathrm{tr}(\Sigma)/\|\Sigma\|$ measures the effective number of active variance directions.

Lemma 3.3 (Approximate manifold support implies low effective rank)**.** *Let X be an $\mathbb{R}^d$-valued random variable with covariance $\Sigma := \mathrm{Cov}(X)$. Assume there exist a k-dimensional C^2 manifold $\mathcal{M} \subset \mathbb{R}^d$ and a measurable map $\Pi : \mathbb{R}^d \to \mathcal{M}$ such that $\mathrm{tr}(\mathrm{Cov}(\Pi(X))) \le C_1 k$, $\mathbb{E}\|X - \Pi(X)\|^2 \le C_2(\delta^2 + \eta)$, and $\|\Sigma\| \ge c > 0$. Then*

$$r_{\mathrm{eff}}(\Sigma) := \frac{\mathrm{tr}(\Sigma)}{\|\Sigma\|} \le \frac{2C_1}{c}\,k + \frac{2C_2}{c}\,(\delta^2 + \eta).$$

In particular, if δ, η are controlled constants and $\|\Sigma\|$ is bounded above and below by constants, then

$$r_{\mathrm{eff}}(\Sigma) = O(k).$$

Lemma 3.3 justifies using $r_{\mathrm{eff}}(\Sigma)$ as an operational intrinsic-dimension proxy: near a k-dimensional manifold with controlled off-manifold error, effective rank is controlled by k rather than ambient dimension d. Theorem 3.4 combines this dimension control with the curvature conditions from Theorems 3.1 and 3.2, connecting low effective dimensionality to representation concentration while curvature controls fluctuations around the mean. The concentration part follows standard Bakry–Émery and Herbst arguments (Bakry et al., 2014; Ledoux, 2001).

Theorem 3.4 (Intrinsic dimension and effective representation complexity)**.** *Let $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ have density $p(x) = Z^{-1}e^{-U(x)}$ for some $U \in C^2(\mathbb{R}^d)$, and let $\Sigma := \mathrm{Cov}(\mu)$. Assume that $\nabla^2 U(x) \succeq mI$ for all $x \in \mathbb{R}^d$, for some $m > 0$, and that ∇U is globally Lipschitz.*

Assume moreover that there exist a k-dimensional C^2 manifold $\mathcal{M} \subset \mathbb{R}^d$ and a measurable map $\Pi : \mathbb{R}^d \to \mathcal{M}$. For $X \sim \mu$, suppose that $\mathrm{tr}(\mathrm{Cov}(\Pi(X))) \le C_1 k$, $\mathbb{E}\|X - \Pi(X)\|^2 \le C_2(\delta^2 + \eta)$, and $\|\Sigma\| \ge c_0 > 0$.

Then

$$r_{\mathrm{eff}}(\Sigma) \le \frac{2C_1}{c_0}\,k + \frac{2C_2}{c_0}(\delta^2 + \eta),$$

and hence

$$\big(\mathbb{E}\|X - \mathbb{E}X\|^2\big)^{1/2} = \sqrt{\mathrm{tr}(\Sigma)} \le \|\Sigma\|^{1/2}\left(\frac{2C_1}{c_0}\,k + \frac{2C_2}{c_0}(\delta^2 + \eta)\right)^{1/2}.$$

Furthermore, there exists an absolute constant $c > 0$ such that for all $t \ge 0$,

$$\mathbb{P}(|\|X - \mathbb{E}X\| - \mathbb{E}\|X - \mathbb{E}X\|| \ge t) \le 2\exp(-cmt^2).$$

Theorem 3.4 combines Lemma 3.3 with concentration under strong log-concavity. It shows that low-dimensional concentration controls effective representation complexity through effective rank, while the curvature parameter m controls fluctuations around the mean through concentration of measure.

Taken together, these results are not intended as guarantees for transformer activations but as theoretical motivation for what a diffusion-friendly representation should look like. Since reverse diffusion uses time-dependent scores of noisy marginals whereas overdamped Langevin dynamics uses the fixed target-distribution score, we use these results only to motivate qualitative desiderata: contraction-like behavior, robustness to score-estimation error, and low effective

complexity. A good layer should therefore exhibit strong curvature-like contraction for stable denoising and low effective dimensionality for easier modeling. Because the true density, Hessian, and manifold structure are unavailable, we approximate these ideas with empirical spectral proxies: local covariance concentration, global precision-based stiffness, and effective rank. All proofs are in Appendix A.

3.2. Locate: Finding Diffusion-Friendly Layers

These theoretical results serve as surrogate motivation, not assumptions that hidden states are strongly log-concave. Rather than guarantees, they motivate desiderata for diffusion-friendly representations: contraction-like behavior, robustness to score-estimation error, and low effective complexity. Since the true density, Hessian, and manifold structure are unavailable, we approximate these desiderata using empirical spectral quantities (see Appendix B for details). For each layer, let $x \in \mathbb{R}^{M\times D}$ denote the activation matrix over M tokens and D hidden dimensions.

We compute three statistics on x. First, the local curvature proxy $\hat{m}_{\mathrm{curv}}$ is obtained from the covariance of k-nearest-neighbor neighborhoods: $m^{(i)}_{\mathrm{curv}} = 1/\lambda_{\max}(\Sigma^{(i)}_{\mathrm{local}})$, with the layer-level value taken as the median; larger values indicate compact neighborhoods. Second, the monotonicity proxy $\hat{m}_{\mathrm{mono}}$ captures global directional stiffness. With $P = (\Sigma + \lambda I)^{-1}$ denoting the regularized precision of the empirical covariance Σ, we compute $m_{ij} = (x_i - x_j)^\top P(x_i - x_j)/\|x_i - x_j\|^2$ for sampled pairs and take the median. Third, effective intrinsic dimension is estimated as $\hat{k} = r_{\mathrm{eff}}(\Sigma) = \mathrm{tr}(\Sigma)/\|\Sigma\|$. Diffusion-friendly layers should have large curvature-related proxies and small effective rank.

We combine these into a selection score: $\mathrm{selection\_score}(\ell) = z(\log \hat{m}_{\mathrm{curv}}(\ell)) + z(\log \hat{m}_{\mathrm{mono}}(\ell)) - z(\log \hat{k}(\ell))$, where $z(\cdot)$ denotes layer-wise z-score normalization. The score rewards curvature proxies while penalizing effective rank. We define *bridgeability* as reconstructability of a layer's hidden state by the diffusion bridge under a matched training protocol, measured by validation loss. The layer sweep evaluates whether this score predicts bridgeability, not to tune it or select an oracle layer. We select $\ell^* = \arg\max_\ell \mathrm{selection\_score}(\ell)$. This is a low-cost layer-selection criterion, not a direct estimator of theoretical constants. Details are in Appendix C.

3.3. Replace: Hidden-State Diffusion Module

Given the selected insertion layer ℓ^*, we replace lower transformer layers with a conditional diffusion bridge. Let $F_{1:\ell^*}$ denote the original computation up to layer ℓ^*, and $F_{\ell^*+1:L}$ the retained upper layers. For input x, the original model produces $h_{\ell^*} = F_{1:\ell^*}(x)$. The bridge is embedding-conditioned: $c(x)$ is derived from the source model's embedding output before the first transformer block. It is trained to reconstruct $\hat{h}_{\ell^*} = D_\theta(c(x))$ in the same hidden space. The bridge does not generate tokens directly. Instead, it reconstructs the selected-layer hidden state $\hat{h}_{\ell^*}$, which is consumed by the retained upper layers as $h_L = F_{\ell^*+1:L}(\hat{h}_{\ell^*})$. The original LM head then maps h_L to token probabilities.

At inference time, lower layers are skipped and D_θ maps this condition to the selected-layer hidden state. We instantiate D_θ as a UNet-based latent denoiser, using a Stable-Diffusion-style architecture as a conditional denoising backbone for hidden-state activations rather than as an image generator; a small-scale ablation is provided in Appendix D.1. Hidden states are projected into a latent tensor, denoised, and projected back to yield $\hat{h}_{\ell^*}$. The bridge is trained on language-model hidden states, with no text-to-image semantics or image supervision. For causal evaluation, DiHAL uses the backbone's left-to-right interface: at step t, the condition uses only prefix tokens $x_{\le t}$, future positions are masked, and the retained causal suffix produces the next-token distribution. Attention and prefix masks are applied consistently to the conditioning pathway and retained suffix.

The main objective is hidden-state denoising rather than standalone text generation. We optimize a diffusion loss $\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{t,\epsilon}[\|\hat{\epsilon}_\theta(z_t, t, c) - \epsilon\|_2^2]$ and a reconstruction loss $\mathcal{L}_{\mathrm{rec}} = \|\hat{h}_{\ell^*} - h_{\ell^*}\|_2^2$. To preserve compatibility with the retained language-modeling interface, we additionally use next-token and logit-distillation losses, $\mathcal{L}_{\mathrm{LM}}$ and $\mathcal{L}_{\mathrm{KD}}$. The overall objective is $\mathcal{L} = \mathcal{L}_{\mathrm{diff}} + \lambda_{\mathrm{rec}}\mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{LM}}\mathcal{L}_{\mathrm{LM}} + \lambda_{\mathrm{KD}}\mathcal{L}_{\mathrm{KD}}$. Implementation details are provided in Appendix C.7.

4. Experiments

4.1. Experimental Setup

We evaluate DiHAL on two representative 8B-scale decoder-only backbones: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), which has 32 transformer layers with hidden size 4096, and Qwen3-8B (Yang et al., 2025), which has 36 layers

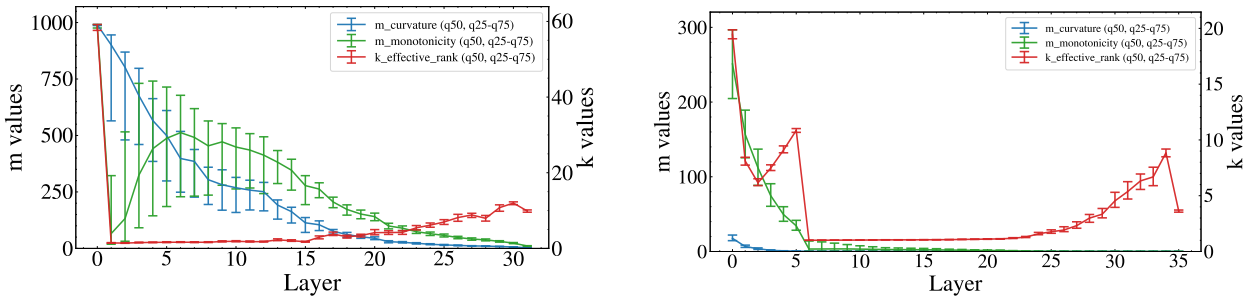

Figure 2. Layer-wise geometry of hidden representations for Llama-3.1-8B-Instruct (left) and Qwen3-8B (right). Curvature-related and effective-rank statistics vary substantially across depth. The selected insertion layer is determined by balancing local compactness, global stiffness, and effective representation complexity, rather than by maximizing a single proxy.

with the same hidden size. For each backbone, we run the source model on 300K sequences from *Dolma v1.7* (Soldaini et al., 2024) and save layerwise hidden states. We estimate the geometric proxies $\hat{m}_{\text{curv}}$, $\hat{m}_{\text{mono}}$, and $\hat{k} = r_{\text{eff}}(\Sigma)$ from 100 repeated 3K-example subsamples, rank candidate insertion layers using the fixed geometry score from Section 3.2, and verify score stability on 30 additional 500-example subsamples.

To test whether the ranking predicts bridgeability, we train one bridge per candidate layer for one epoch on a 150K-example subset with a 9:1 train/validation split and measure validation bridge loss. This sweep evaluates the geometry score but does not fit it. Each bridge is embedding-conditioned and targets the corresponding layer hidden state. We instantiate it with Stable-Diffusion-v1.5-style latent denoising components (Rombach et al., 2022), freezing the VAE while training the UNet and bridge-specific projections. These components are repurposed for hidden-state denoising; training uses no CLIP conditioning, text-to-image objective, or image supervision. Finally, we fully train the highest-scoring layer on the 300K-example corpus for four epochs and report negative log-likelihood (NLL), perplexity (PPL), and output-distribution KL divergence against the original pretrained model.

Evaluation. We evaluate three aspects of DiHAL. For layer selection, we compare the geometry ranking with validation bridge loss and report Spearman correlation, Kendall correlation, the best predicted layer, the best observed layer, and their rank gap. For final model quality, we report NLL and PPL on *WikiText-103* (Merity et al., 2016) and held-out *Dolma v1.7*. For teacher alignment, we compute KL divergence between teacher and DiHAL logits. Additional implementation and hyperparameter details are provided in Appendix D.

4.2. Layer-Wise Geometry

We first examine whether transformer hidden representations exhibit systematic geometric variation across depth. Figure 2 shows clear layer dependence in both backbones: input-adjacent layers tend to have large local curvature values, while global monotonicity and effective rank follow different depth-dependent trends. These patterns suggest that transformer hidden states do not form a uniform sequence of equally suitable diffusion spaces. Large $\hat{m}_{\text{curv}}$ indicates locally compact neighborhoods, but local compactness does not necessarily imply globally coherent stiffness, as measured by $\hat{m}_{\text{mono}}$. Likewise, low effective rank alone does not guarantee strong curvature-related structure. Thus, layer selection reflects a curvature–dimension trade-off rather than optimization of a single proxy.

The fixed geometry score combines local curvature, global monotonicity, and effective rank. It selects layer 3 for Llama-3.1-8B and layer 2 for Qwen3-8B, rather than defaulting to the largest single proxy. Both selected layers are close to the embedding interface, suggesting that continuous diffusion may be suitable for hidden spaces that retain embedding-like geometric structure while remaining easier to denoise than token embeddings themselves. Since geometry alone does not guarantee bridgeability, we next test whether this ranking predicts bridgeability under a matched budget.

4.3. Fixed-Budget Layer Sweep

We perform a fixed-budget layer sweep to test whether the geometric pattern in Figure 2 translates into bridgeability. For each candidate layer, we train one bridge for one epoch on 150K examples and measure validation bridge loss; the sweep evaluates the geometry score but does not fit it. We compare against single-proxy baselines using only $\hat{m}_{\text{curv}}$

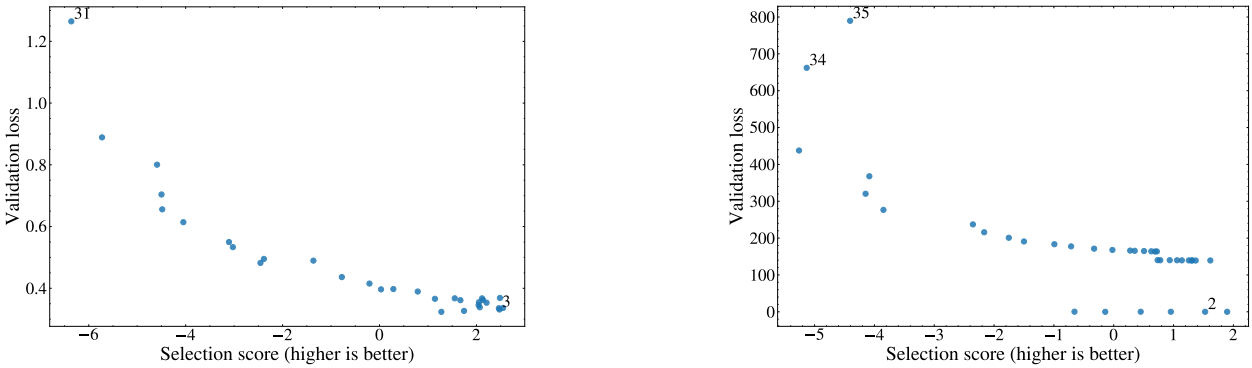

Figure 3. Fixed geometry score versus validation bridge loss for Llama-3.1-8B-Instruct (left) and Qwen3-8B (right). Each point is one candidate layer, labeled by index. Higher scores generally correspond to lower validation loss, indicating that the fixed geometry score predicts bridgeability.

Table 1. Fixed-budget layer sweep results. We compare the geometry-selected layer against representative baselines under the same bridge-training budget. Validation bridge loss measures bridgeability.

Model	Layer type	Layer	Selection score ↑	$\hat{m}_{\mathrm{curv}}$ ↑	$\hat{m}_{\mathrm{mono}}$ ↑	$\hat{k}$ ↓	Val. bridge loss ↓
Llama-3.1-8B	Geometry-selected	3	**2.360**	553.166	227.324	1.332	0.331
	Curvature-only	1	1.271	**780.707**	45.976	**1.310**	**0.324**
	Dimension-only	2	1.857	701.159	102.836	1.314	0.327
	Early baseline	7	2.324	258.717	**374.482**	1.472	0.362
	Middle baseline	17	0.265	51.230	134.336	2.679	0.397
	Late baseline	27	-4.094	7.643	23.769	8.305	0.656
Qwen3-8B	Geometry-selected	2	**2.044**	168.963	317.682	4.544	0.060
	Curvature-only	1	1.661	**292.434**	**450.606**	8.267	**0.059**
	Dimension-only	6	1.492	4.965	7.484	**1.008**	139.092
	Early baseline	7	1.430	4.602	7.237	1.012	139.161
	Middle baseline	18	0.616	1.036	3.535	1.097	164.087
	Late baseline	30	-3.734	0.034	0.112	4.089	276.584

or $\hat{k}$, and depth-based Early/Middle/Late baselines. Figure 2 suggests that embedding-adjacent layers form a distinct geometric regime, with stronger curvature-related proxies near the input and less favorable effective rank in later layers.

If the score is meaningful, higher-scoring layers should achieve lower validation loss. We therefore correlate the geometry score with negative validation bridge loss, where higher is better. Figure 3 visualizes this trend, and Table 1 shows that the score is not intended to select the exact minimum-loss layer but still selects layers close to the best observed loss and far below middle and late baselines. For Llama-3.1-8B, compared with selected layer 3, layer 17 has much lower curvature ($\hat{m}_{\mathrm{curv}} = 51.230$ vs. 553.166) and higher effective rank ($\hat{k} = 2.679$ vs. 1.332); layer 27 further decreases in curvature to 7.643 and increases in effective rank to 8.305. Qwen3-8B shows a similar pattern. These results suggest that deeper, more task-specialized representations are less favorable for diffusion-based reconstruction under the matched bridge-training protocol.

Across all candidate layers, the fixed score shows strong agreement with bridgeability. Table 2 reports correlations over 30 repeated 500-example score-estimation runs. The score achieves Spearman $\rho = 0.9143 \pm 0.0069$ on Llama-3.1-8B and $\rho = 0.9267 \pm 0.0157$ on Qwen3-8B, with rank gaps of 2 and 1 between the best predicted and best observed layers. These results show that the fixed geometry score is useful for cheaply identifying bridgeable layers under a matched training budget.

4.4. Diagnostic Matched-Budget Comparison of Continuous Diffusion Methods

We compare continuous diffusion methods under the same budget for training diffusion/recovery components in the Qwen3-8B setting. Baselines cover token-embedding denoising (Diffusion-LM (Li et al., 2022), SED (Strudel et al.,

Table 2. Agreement between fixed geometry score and bridgeability under the fixed-budget sweep. Correlations use negative validation bridge loss; parentheses report standard deviations over 30 repeated 500-example score runs. Geometry proxies use 100 repeated 3K-example subsamples.

Model	Spearman $\rho \uparrow$	Kendall $\tau \uparrow$	Pearson $r \uparrow$	Best pred.	Best obs.	Rank gap $\downarrow$
Llama-3.1-8B	0.9143 ($\pm$0.0069)	0.8011 ($\pm$0.0086)	0.8687 ($\pm$0.0027)	3	1	2
Qwen3-8B	0.9267 ($\pm$0.0157)	0.8208 ($\pm$0.0256)	0.8605 ($\pm$0.0098)	2	1	1

Table 3. Diagnostic same-budget comparison of continuous diffusion target spaces and recovery interfaces. This is not a pretraining-matched comparison: DiHAL reuses a pretrained transformer suffix and LM head, while some baselines use smaller standalone recovery modules.

Method	Diffusion target	Recovery interface	Gen.PPL $\downarrow$	Div. $\uparrow$
DiHAL	Hidden state h_{ℓ^*}	Retained upper layers + LM head	**136.02**	**0.5913**
Diffusion-LM	Token embeddings	Embedding-to-token recovery	683.43	0.4324
SED	Self-conditioned embeddings	Embedding-to-token recovery	778.82	0.4942
LD4LG	Learned text latent	Frozen BART + latent decoder	166.11	0.5797
CoDAR	Continuous token/latent states	Continuous-to-discrete recovery	144.83	0.4777

2023)), learned-latent denoising (LD4LG (Lovelace et al., 2023)), and continuous-to-discrete recovery (CoDAR (Shen et al., 2026)). In contrast, DiHAL denoises an internal hidden state and delegates token prediction to retained Qwen3-8B suffix and LM head. This is a diagnostic, not pretraining-matched, comparison: all methods use the same 300K-example corpus and 40 H100-hour budget for diffusion/recovery training but differ in how they reuse pretrained components.

We evaluate generated text using Gen.PPL, the perplexity assigned by a GPT-2 evaluator, and diversity, defined as the product of Distinct-1 through Distinct-4, where Distinct-n is the ratio of unique n-grams to generated n-grams. Table 3 shows that DiHAL achieves the lowest Gen.PPL and highest diversity in this setting. Compared with CoDAR, DiHAL improves Gen.PPL from 144.83 to 136.02 and diversity from 0.4777 to 0.5913, suggesting that moving diffusion from token recovery to hidden-state recovery can be beneficial in this diagnostic setting. Additional details are in Appendix E.

4.5. Top-Layer Full Training and Evaluation

We fully train the highest-ranked layer to test whether the short-budget signal transfers to a larger optimization setting. We compare it against two controls: a validation-loss oracle, defined as the layer with the lowest validation bridge loss in the fixed-budget sweep, and a worst-layer control, which tests whether poor insertion points degrade the hybrid model. Since identifying the oracle requires explicitly training bridges across candidate layers, this comparison tests whether the geometry score can select a competitive layer without using validation bridge loss as the criterion.

Table 4 reports the final evaluation. We also include CoDAR, a recent continuous-diffusion baseline, re-evaluated under the same pipeline. The geometry-selected layer improves over the worst-layer control and remains competitive with the validation-loss oracle. On Llama-3.1-8B, it outperforms the oracle in NLL and PPL, showing that the fixed-budget oracle is not always optimal for final language-modeling metrics. On Qwen3-8B, the oracle is slightly better, but the geometry-selected layer remains comparable without layer-wise bridge training and substantially improves over CoDAR. Thus, this evaluation supports geometry-based layer selection and hidden-state recovery, while the remaining gap to the original autoregressive teacher clarifies DiHAL's scope.

5. Related Work

Diffusion language models adapt diffusion to text, where discreteness makes generation less direct than in images (Hoogeboom et al., 2022). Prior work includes discrete token diffusion (Austin et al., 2021; Gong et al., 2023), masked diffusion with iterative refinement (Sahoo et al., 2024; Nie et al., 2025; Ye et al., 2025), and continuous diffusion over embeddings or learned latents (Li et al., 2022; Lovelace et al., 2023). Continuous methods avoid token corruption but

Table 4. Final evaluation of DiHAL insertion layers against CoDAR on the combined WikiText-103 and held-out Dolma evaluation sets. CoDAR is a recent strong continuous-diffusion baseline re-evaluated in the Qwen3-8B setting under the same evaluation pipeline.

Model	Method	Layer	NLL ↓	PPL ↓	KL ↓
Llama-3.1-8B	DiHAL, geometry-selected	3	**4.91**	**135.64**	0.73
	Validation-loss oracle	1	5.11	165.67	**0.62**
	Worst layer	31	5.17	175.91	1.32
Qwen3-8B	DiHAL, geometry-selected	2	4.97	144.03	**0.53**
	Validation-loss oracle	1	**4.94**	**139.77**	0.54
	Worst layer	35	5.23	186.79	1.46
Recent continuous diffusion	CoDAR	N/A	5.18	177.87	N/A

require recovering tokens from denoised vectors, which can introduce projection or decoding errors (Wang et al., 2022; Li et al., 2022). We instead study transformer hidden states as a diffusion-friendly denoising space.

Diffusion behavior depends on representation geometry, including curvature, conditioning, and intrinsic dimension (Pidstrigach, 2022; Rombach et al., 2022). In parallel, efficient generation has been studied through transformer compression, distillation, layer reduction, early exiting, and hybrid modules (Fan et al., 2020; Sanh et al., 2020; Lenz et al., 2025). DiHAL connects these directions by identifying internal transformer representations suitable for diffusion-based replacement.

6. Limitations

Continuous diffusion language modeling still faces important limitations at scale, especially because it must learn both continuous denoising and reliable recovery back to discrete language. `DiHAL` mitigates this difficulty by moving diffusion to an internal hidden-state interface, but it is not a standalone diffusion language model: token prediction still depends on the retained transformer suffix and LM head. Due to compute constraints, we do not fully explore larger bridges, longer training, or deeper replacement. Future work could incorporate the proposed geometric proxies directly into bridge training, making deeper hidden states more bridgeable and potentially enabling replacement of larger transformer prefixes. We therefore view DiHAL as a step toward understanding where continuous diffusion can operate inside pretrained language models.

7. Conclusion

We introduced `DiHAL`, a geometry-guided diffusion–transformer hybrid that moves continuous diffusion from token-level recovery to internal hidden-state reconstruction. Instead of treating the continuous-to-discrete interface as an unavoidable bottleneck, DiHAL asks where inside a pretrained language model diffusion should operate. It locates diffusion-friendly layers using curvature, monotonicity, and effective-rank proxies, then replaces the lower transformer prefix with a conditional diffusion bridge while preserving the upper layers and original LM head. This yields a simple but important reframing: continuous diffusion need not generate language by directly recovering tokens; it can reconstruct a representation that the pretrained transformer already knows how to decode.

Our experiments show that interface choice is not incidental. Across two 8B-scale backbones, geometry-selected insertion points are embedding-adjacent, predict bridgeability under fixed-budget training, and remain competitive with validation-loss oracles after training. Middle and late hidden states are harder to reconstruct, suggesting that diffusion failures in language reflect not only discreteness but a mismatch between denoising and representation-space geometry.

DiHAL is not yet a standalone diffusion language model and still relies on retained pretrained layers. However, this limitation sharpens the paper's main lesson: successful continuous diffusion for language may require choosing or learning the right internal interface, rather than applying diffusion uniformly to arbitrary continuous spaces. By identifying where diffusion can effectively enter an existing language model, DiHAL provides a step toward principled diffusion–transformer hybrids and future models that use geometry not only to locate, but also to train, more bridgeable representations.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and Geometry of Markov Diffusion Operators*, volume 348 of *Grundlehren der mathematischen Wissenschaften*. Springer Cham, 2014. ISBN 978-3-319-00227-9. doi: 10.1007/978-3-319-00227-9.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylO2yStDr.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq-v2: Bridging discrete and continuous text spaces for accelerated seq2seq diffusion models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=OLcDbSRjbx.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh

Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In

Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=6nbpPqUCIi7.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Lm8T39vLDTE.

Jaehyeong Jo and Sung Ju Hwang. Continuous diffusion model for language modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=VGv5y60sXC.

M. Ledoux. *The Concentration of Measure Phenomenon*. Mathematical surveys and monographs. American Mathematical Society, 2001. ISBN 9780821837924. URL https://books.google.com.sg/books?id=mCX_cWL6rqwC.

Barak Lenz, Opher Lieber, Alan Arazi, Amir Bergman, Avshalom Manevich, Barak Peleg, Ben Aviram, Chen Almagor, Clara Fridman, Dan Padnos, Daniel Gissin, Daniel Jannai, Dor Muhlgay, Dor Zimberg, Edden M. Gerber, Elad Dolev, Eran Krakovsky, Erez Safahi, Erez Schwartz, Gal Cohen, Gal Shachaf, Haim Rozenblum, Hofit Bata, Ido Blass, Inbal Magar, Itay Dalmedigos, Jhonathan Osin, Julie Fadlon, Maria Rozman, Matan Danos, Michael Gokhman, Mor Zusman, Naama Gidron, Nir Ratner, Noam Gat, Noam Rozen, Oded Fried, Ohad Leshno, Omer Antverg, Omri Abend, Or Dagan, Orit Cohavi, Raz Alon, Ro'i Belson, Roi Cohen, Rom Gilad, Roman Glozman, Shahar Lev, Shai Shalev-Shwartz, Shaked Haim Meirom, Tal Delbari, Tal Ness, Tomer Asida, Tom Ben Gal, Tom Braude, Uriya Pumerantz, Josh Cohen, Yonatan Belinkov, Yuval Globerson, Yuval Peleg Levy, and Yoav Shoham. Jamba: Hybrid transformer-mamba language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JFPaD7lpBD.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=3s9IrEsjLyk.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Seo Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=NKdtztladR.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

Jakiw Pidstrigach. Score-based generative models detect manifolds, 2022. URL https://proceedings.nips.cc/paper_files/paper/2022/file/e8fb575e3ede31f9b8c05d53514eb7c6-Paper-Conference.pdf.

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 (CVPR)*, pages 10684–10695, June 2022.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://www.emc2-ai.org/assets/docs/neurips-19/emc2-neurips19-paper-33.pdf.

Junzhe Shen, Jieru Zhao, Ziwei He, and Zhouhan Lin. Codar: Continuous diffusion language models are more powerful than you think, 2026. URL https://arxiv.org/abs/2603.02547.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024. URL https://arxiv.org/abs/2402.00159.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Robin Strudel, Corentin Tallec, Florent Altché, Yilun Du, Yaroslav Ganin, Arthur Mensch, Will Sussman Grathwohl, Nikolay Savinov, Sander Dieleman, Laurent Sifre, and Rémi Leblond. Self-conditioned embedding diffusion for text generation, 2023. URL https://openreview.net/forum?id=OpzV3lp3IMC.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer Berlin, Heidelberg, 2009. ISBN 978-3-540-71050-9. doi: 10.1007/978-3-540-71050-9.

Rose E Wang, Esin Durmus, Noah Goodman, and Tatsunori Hashimoto. Language modeling via stochastic processes. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=pMQwKL1yctf.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Siyue Zhang, Yilun Zhao, Liyuan Geng, Arman Cohan, Anh Tuan Luu, and Chen Zhao. Diffusion vs. autoregressive language models: A text embedding perspective. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4273–4303, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.213. URL https://aclanthology.org/2025.emnlp-main.213/.

A. Proofs of Theorems

Preliminaries

Law of a random variable. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $X : \Omega \to \mathbb{R}^d$ be a random variable. The law (distribution) of X, denoted by $\mathcal{L}(X)$, is the probability measure defined by

$$\mathcal{L}(X)(A) := \mathbb{P}(X \in A), \quad A \in \mathcal{B}(\mathbb{R}^d).$$

2-Wasserstein distance. Let $\mathcal{P}_2(\mathbb{R}^d)$ denote the set of probability measures with finite second moment. For $\nu, \mu \in \mathcal{P}_2(\mathbb{R}^d)$,

$$W_2^2(\nu, \mu) := \inf_{\pi \in \Pi(\nu,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, \pi(dx, dy),$$

where $\Pi(\nu, \mu)$ denotes the set of couplings of ν and μ. Equivalently,

$$W_2^2(\nu, \mu) = \inf \left\{ \mathbb{E}\|X - Y\|^2 : \mathcal{L}(X) = \nu, \mathcal{L}(Y) = \mu \right\}.$$

In $\mathbb{R}^d$, an optimal coupling attaining the infimum exists. If preferred, one may instead work with ε-optimal couplings and let $\varepsilon \downarrow 0$ at the end.

Lemma A.1 (Strong convexity implies gradient monotonicity)**.** *Let $U : \mathbb{R}^d \to \mathbb{R}$ be differentiable and m-strongly convex, i.e.*

$$U(y) \geq U(x) + \langle \nabla U(x), y - x \rangle + \frac{m}{2}\|y - x\|^2 \quad \textit{for all } x, y.$$

Then

$$\langle \nabla U(x) - \nabla U(y), x - y \rangle \geq m\|x - y\|^2 \quad \textit{for all } x, y.$$

Proof. Apply strong convexity twice, swapping x and y:

$$U(y) \geq U(x) + \langle \nabla U(x), y - x \rangle + \frac{m}{2}\|y - x\|^2,$$

$$U(x) \geq U(y) + \langle \nabla U(y), x - y \rangle + \frac{m}{2}\|x - y\|^2.$$

Summing the two inequalities yields

$$\langle \nabla U(x) - \nabla U(y), x - y \rangle \geq m\|x - y\|^2.$$

□

Lemma A.2 (Gibbs invariance and uniqueness)**.** *Assume that $U \in C^2(\mathbb{R}^d)$, ∇U is globally Lipschitz, and*

$$\nabla^2 U(x) \succeq mI \qquad \textit{for all } x \in \mathbb{R}^d$$

for some $m > 0$. Let

$$\mu(dx) = Z^{-1} e^{-U(x)} \, dx, \qquad Z := \int_{\mathbb{R}^d} e^{-U(x)} \, dx.$$

Then $Z < \infty$, $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, and μ is invariant for

$$dX_t = -\nabla U(X_t) \, dt + \sqrt{2} \, dW_t.$$

Moreover, if for every initial law $\nu_0 \in \mathcal{P}_2(\mathbb{R}^d)$ the law $\nu_t := \mathcal{L}(X_t)$ satisfies

$$W_2(\nu_t, \mu) \leq e^{-mt} W_2(\nu_0, \mu),$$

then μ is the unique invariant distribution in $\mathcal{P}_2(\mathbb{R}^d)$.

Proof. By strong convexity, for every $x \in \mathbb{R}^d$,

$$U(x) \geq U(0) + \langle \nabla U(0), x \rangle + \frac{m}{2}\|x\|^2.$$

Completing the square, this implies

$$e^{-U(x)} \leq C \exp\Big(-\frac{m}{4}\|x\|^2\Big)$$

for some constant $C < \infty$. Hence $Z < \infty$ and μ has finite second moment.

Let

$$L\varphi(x) := \Delta\varphi(x) - \langle \nabla U(x), \nabla\varphi(x) \rangle$$

be the generator. For $\varphi \in C_c^\infty(\mathbb{R}^d)$, integration by parts gives

$$\int_{\mathbb{R}^d} L\varphi(x)\,\mu(dx) = Z^{-1} \int_{\mathbb{R}^d} \big(\Delta\varphi(x) - \langle \nabla U(x), \nabla\varphi(x) \rangle\big) e^{-U(x)}\,dx = 0.$$

Since this holds for all $\varphi \in C_c^\infty(\mathbb{R}^d)$, the stationary equation $L^*\mu = 0$ holds in the weak sense. Therefore μ is invariant.

Finally, if $\nu \in \mathcal{P}_2(\mathbb{R}^d)$ is any invariant distribution, then applying the contraction estimate with $\nu_0 = \nu$ yields

$$W_2(\nu, \mu) = W_2(\nu_t, \mu) \leq e^{-mt} W_2(\nu, \mu) \qquad \text{for all } t \geq 0.$$

Fix any $t > 0$. Since $e^{-mt} < 1$, this implies $W_2(\nu, \mu) = 0$, hence $\nu = \mu$. □

Proof of Theorem 3.1

By Lemma A.2, μ is an invariant distribution for

$$dX_t = -\nabla U(X_t)\,dt + \sqrt{2}\,dW_t.$$

Let $\nu_0 := \mathcal{L}(X_0)$. Choose an optimal coupling (X_0, Y_0) between ν_0 and μ such that

$$\mathbb{E}\|X_0 - Y_0\|^2 = W_2^2(\nu_0, \mu).$$

(Alternatively, choose an ε-optimal coupling and let $\varepsilon \downarrow 0$.) We take this initial coupling to be independent of the Brownian motion below.

Define the synchronous coupling:

$$dX_t = -\nabla U(X_t)\,dt + \sqrt{2}\,dW_t,$$
$$dY_t = -\nabla U(Y_t)\,dt + \sqrt{2}\,dW_t,$$

driven by the same Brownian motion W_t.

Since $Y_0 \sim \mu$ and μ is invariant, we have

$$\mathcal{L}(Y_t) = \mu \qquad \text{for all } t \geq 0.$$

Let $Z_t := X_t - Y_t$. Subtracting the SDEs gives

$$dZ_t = -(\nabla U(X_t) - \nabla U(Y_t))\,dt.$$

Since the stochastic terms cancel, Z_t has absolutely continuous paths and satisfies the pathwise ODE

$$\dot{Z}_t = -(\nabla U(X_t) - \nabla U(Y_t)) \qquad \text{for a.e. } t \geq 0.$$

Therefore the ordinary chain rule applies, and

$$\frac{d}{dt}\|Z_t\|^2 = 2\langle Z_t, \dot{Z}_t \rangle = -2\langle Z_t, \nabla U(X_t) - \nabla U(Y_t) \rangle.$$

By Lemma A.1,

$$\frac{d}{dt}\|Z_t\|^2 \le -2m\|Z_t\|^2.$$

By Grönwall's inequality,

$$\|Z_t\|^2 \le e^{-2mt}\|Z_0\|^2.$$

Taking expectation yields

$$\mathbb{E}\|X_t - Y_t\|^2 \le e^{-2mt}\mathbb{E}\|X_0 - Y_0\|^2.$$

Since (X_t, Y_t) is a coupling of $\mathcal{L}(X_t)$ and μ, by definition of W_2,

$$W_2^2(\mathcal{L}(X_t), \mu) \le \mathbb{E}\|X_t - Y_t\|^2.$$

Therefore,

$$W_2^2(\mathcal{L}(X_t), \mu) \le e^{-2mt}W_2^2(\nu_0, \mu).$$

Taking square roots gives

$$W_2(\mathcal{L}(X_t), \mu) \le e^{-mt}W_2(\nu_0, \mu).$$

It remains to prove uniqueness of the invariant distribution in $P_2(\mathbb{R}^d)$. Let $\pi \in P_2(\mathbb{R}^d)$ be any invariant distribution of the same Langevin SDE. Applying the contraction estimate above with $\nu_0 = \pi$ gives, for every $t > 0$,

$$W_2(\pi, \mu) = W_2(\mathcal{L}(X_t), \mu) \le e^{-mt}W_2(\pi, \mu),$$

where we used the invariance of π to obtain $\mathcal{L}(X_t) = \pi$. Since $m > 0$ and $e^{-mt} < 1$ for $t > 0$, this implies

$$W_2(\pi, \mu) = 0.$$

Hence $\pi = \mu$. Therefore μ is the unique invariant distribution in $P_2(\mathbb{R}^d)$. □

Proof of Theorem 3.2

We prove stability of an invariant measure under a uniformly bounded perturbation of the score function.

Let $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ have density p and define

$$U(x) := -\log p(x).$$

Assume that $U \in C^2(\mathbb{R}^d)$ is m-strongly convex, i.e.

$$\nabla^2 U(x) \succeq mI \quad \text{for all } x \in \mathbb{R}^d,$$

for some $m > 0$, and that ∇U is globally Lipschitz.

Define the true score

$$s(x) := \nabla \log p(x) = -\nabla U(x).$$

Let $\hat{s}$ be a globally Lipschitz function satisfying

$$\sup_{x\in\mathbb{R}^d} \|\hat{s}(x) - s(x)\| \le \varepsilon.$$

Consider the SDEs

$$dX_t = s(X_t)\,dt + \sqrt{2}\,dW_t, \qquad d\hat{X}_t = \hat{s}(\hat{X}_t)\,dt + \sqrt{2}\,dW_t,$$

and assume that the second SDE admits an invariant distribution $\hat{\mu}$. By Lemma A.2, μ is an invariant distribution of the first SDE.

We first show that $\hat{\mu} \in \mathcal{P}_2(\mathbb{R}^d)$. Since $s = -\nabla U$ and U is m-strongly convex, Lemma A.1 applied with $y = 0$ gives

$$\langle x, s(x) - s(0)\rangle \le -m\|x\|^2 \qquad \text{for all } x \in \mathbb{R}^d.$$

Hence

$$\langle x, s(x)\rangle = \langle x, s(x) - s(0)\rangle + \langle x, s(0)\rangle \le -m\|x\|^2 + \|s(0)\|\,\|x\|.$$

Using $\|\hat{s}(x) - s(x)\| \le \varepsilon$, we obtain

$$\langle x, \hat{s}(x)\rangle \le \langle x, s(x)\rangle + \varepsilon\|x\| \le -m\|x\|^2 + (\|s(0)\| + \varepsilon)\|x\|.$$

By Young's inequality, there exists a constant $C_0 < \infty$ such that

$$\langle x, \hat{s}(x)\rangle \le -\frac{m}{2}\|x\|^2 + C_0 \qquad \text{for all } x \in \mathbb{R}^d.$$

Let

$$\hat{L}f(x) := \langle \hat{s}(x), \nabla f(x)\rangle + \Delta f(x)$$

be the generator of the perturbed SDE, and set

$$V(x) := \|x\|^2.$$

Then

$$\hat{L}V(x) = 2\langle x, \hat{s}(x)\rangle + 2d \le -m\|x\|^2 + C_1 = -mV(x) + C_1$$

for some constant $C_1 < \infty$.

We now justify the use of this unbounded Lyapunov function by truncation. Let $\psi_R \in C^2([0,\infty))$ be nondecreasing and concave, with

$$0 \le \psi_R' \le 1, \qquad \psi_R'' \le 0,$$

such that

$$\psi_R(r) = r \quad \text{for } r \le R, \qquad \psi_R(r) = \text{constant} \quad \text{for } r \ge 2R.$$

Define

$$V_R(x) := \psi_R(V(x)).$$

Since $V_R \in C_b^2(\mathbb{R}^d)$ and the perturbed SDE has globally Lipschitz drift, the associated Markov semigroup $(\widehat{P}_t)_{t\ge 0}$ has generator $\hat{L}$. Hence

$$\frac{d}{dt}\widehat{P}_t V_R\Big|_{t=0} = \hat{L}V_R.$$

By invariance of $\hat{\mu}$,

$$\int_{\mathbb{R}^d} \widehat{P}_t V_R(x)\,\hat{\mu}(dx) = \int_{\mathbb{R}^d} V_R(x)\,\hat{\mu}(dx) \quad \text{for all } t \ge 0.$$

Differentiating at $t = 0$ gives

$$\int_{\mathbb{R}^d} \hat{L}V_R(x)\,\hat{\mu}(dx) = 0.$$

By the chain rule for $\hat{L}$,

$$\hat{L}V_R = \psi_R'(V)\hat{L}V + \psi_R''(V)\|\nabla V\|^2.$$

Since $\psi_R'' \le 0$, we have

$$\hat{L}V_R \le \psi_R'(V)\hat{L}V \le \psi_R'(V)(-mV + C_1).$$

Therefore

$$0 = \int \hat{L}V_R\,d\hat{\mu} \le -m\int \psi_R'(V)V\,d\hat{\mu} + C_1\int \psi_R'(V)\,d\hat{\mu}.$$

Since $0 \le \psi_R' \le 1$, this implies

$$m\int \psi_R'(V)V\,d\hat{\mu} \le C_1.$$

Moreover, $\psi_R'(V) = 1$ whenever $V \le R$, and hence

$$m\int_{\{V\le R\}} V\,d\hat{\mu} \le C_1.$$

Letting $R \to \infty$ and using monotone convergence gives

$$\int \|x\|^2 \, \hat{\mu}(dx) = \int V \, d\hat{\mu} \leq \frac{C_1}{m} < \infty.$$

Thus $\hat{\mu} \in \mathcal{P}_2(\mathbb{R}^d)$.

Now choose an optimal coupling $(X_0, \hat{X}_0)$ of μ and $\hat{\mu}$ such that

$$\mathbb{E}\|X_0 - \hat{X}_0\|^2 = W_2^2(\mu, \hat{\mu}).$$

We take this initial coupling to be independent of the Brownian motion below. Drive both SDEs by the same Brownian motion $(W_t)_{t \geq 0}$ and define

$$Z_t := \hat{X}_t - X_t.$$

Subtracting the two SDEs yields

$$dZ_t = \big(\hat{s}(\hat{X}_t) - s(X_t)\big) \, dt.$$

Since the stochastic terms cancel, Z_t has absolutely continuous paths and satisfies the pathwise ODE

$$\dot{Z}_t = \hat{s}(\hat{X}_t) - s(X_t) \qquad \text{for a.e. } t \geq 0.$$

Therefore the ordinary chain rule applies, and

$$\frac{d}{dt}\|Z_t\|^2 = 2\langle Z_t, \hat{s}(\hat{X}_t) - s(X_t)\rangle.$$

We decompose

$$\hat{s}(\hat{X}_t) - s(X_t) = \big(\hat{s}(\hat{X}_t) - s(\hat{X}_t)\big) + \big(s(\hat{X}_t) - s(X_t)\big).$$

Hence

$$\frac{d}{dt}\|Z_t\|^2 = 2\langle Z_t, \hat{s}(\hat{X}_t) - s(\hat{X}_t)\rangle + 2\langle Z_t, s(\hat{X}_t) - s(X_t)\rangle.$$

For the first term, using the uniform bound on the score error,

$$2\langle Z_t, \hat{s}(\hat{X}_t) - s(\hat{X}_t)\rangle \leq 2\varepsilon\|Z_t\|.$$

For the second term, Lemma A.1 implies

$$\langle \hat{X}_t - X_t, \nabla U(\hat{X}_t) - \nabla U(X_t)\rangle \geq m\|\hat{X}_t - X_t\|^2.$$

Since $s = -\nabla U$, we get

$$\langle Z_t, s(\hat{X}_t) - s(X_t)\rangle \leq -m\|Z_t\|^2.$$

Combining the two bounds gives

$$\frac{d}{dt}\|Z_t\|^2 \leq -2m\|Z_t\|^2 + 2\varepsilon\|Z_t\|.$$

By Young's inequality,

$$2\varepsilon\|Z_t\| \leq m\|Z_t\|^2 + \frac{\varepsilon^2}{m}.$$

Substituting this into the differential inequality, we obtain

$$\frac{d}{dt}\|Z_t\|^2 \leq -m\|Z_t\|^2 + \frac{\varepsilon^2}{m}.$$

Multiplying both sides by e^{mt} and integrating from 0 to t yields

$$\|Z_t\|^2 \leq e^{-mt}\|Z_0\|^2 + \frac{\varepsilon^2}{m^2}\big(1 - e^{-mt}\big) \qquad \text{a.s.}$$

Taking expectations gives

$$\mathbb{E}\|Z_t\|^2 \le e^{-mt}\mathbb{E}\|Z_0\|^2 + \frac{\varepsilon^2}{m^2}\big(1 - e^{-mt}\big).$$

Since $X_0 \sim \mu$ and $\hat{X}_0 \sim \hat{\mu}$ are invariant initial laws, we have

$$X_t \sim \mu, \qquad \hat{X}_t \sim \hat{\mu} \qquad \text{for all } t \ge 0.$$

Therefore $(X_t, \hat{X}_t)$ is a coupling of μ and $\hat{\mu}$, so

$$W_2^2(\hat{\mu}, \mu) \le \mathbb{E}\|\hat{X}_t - X_t\|^2 = \mathbb{E}\|Z_t\|^2.$$

Combining this with the previous estimate gives

$$W_2^2(\hat{\mu}, \mu) \le e^{-mt} W_2^2(\hat{\mu}, \mu) + \frac{\varepsilon^2}{m^2}\big(1 - e^{-mt}\big).$$

Letting $t \to \infty$, we obtain

$$W_2^2(\hat{\mu}, \mu) \le \frac{\varepsilon^2}{m^2}.$$

Taking square roots gives

$$W_2(\hat{\mu}, \mu) \le \frac{\varepsilon}{m}.$$

□

Proof of Lemma 3.3

Let

$$\Sigma := \mathrm{Cov}(X).$$

Define

$$R := X - \Pi(X).$$

Since

$$\mathbb{E}X = \mathbb{E}\Pi(X) + \mathbb{E}R,$$

we may write

$$X - \mathbb{E}X = \big(\Pi(X) - \mathbb{E}\Pi(X)\big) + \big(R - \mathbb{E}R\big).$$

Therefore, using the inequality $\|u + v\|^2 \le 2\|u\|^2 + 2\|v\|^2$, we obtain

$$\mathrm{tr}(\Sigma) = \mathbb{E}\|X - \mathbb{E}X\|^2 \le 2\,\mathbb{E}\|\Pi(X) - \mathbb{E}\Pi(X)\|^2 + 2\,\mathbb{E}\|R - \mathbb{E}R\|^2.$$

For the first term, by assumption,

$$\mathbb{E}\|\Pi(X) - \mathbb{E}\Pi(X)\|^2 = \mathrm{tr}(\mathrm{Cov}(\Pi(X))) \le C_1 k.$$

For the second term, since covariance is dominated by the second moment,

$$\mathbb{E}\|R - \mathbb{E}R\|^2 = \mathrm{tr}(\mathrm{Cov}(R)) \le \mathbb{E}\|R\|^2.$$

Using the approximation assumption,

$$\mathbb{E}\|R\|^2 = \mathbb{E}\|X - \Pi(X)\|^2 \le C_2(\delta^2 + \eta).$$

Combining the two bounds yields

$$\mathrm{tr}(\Sigma) \le 2C_1 k + 2C_2(\delta^2 + \eta).$$

By definition,

$$r_{\text{eff}}(\Sigma) = \frac{\operatorname{tr}(\Sigma)}{\|\Sigma\|}.$$

Using the non-degeneracy assumption $\|\Sigma\| \geq c > 0$, we obtain

$$r_{\text{eff}}(\Sigma) \leq \frac{2C_1 k + 2C_2(\delta^2 + \eta)}{c} = \frac{2C_1}{c}\, k + \frac{2C_2}{c}\,(\delta^2 + \eta).$$

In particular, if $\delta, \eta = O(1)$ and $\|\Sigma\|$ is bounded below by a positive constant, then

$$r_{\text{eff}}(\Sigma) = O(k + 1).$$

If additionally $k \geq 1$, this simplifies to

$$r_{\text{eff}}(\Sigma) = O(k).$$

□

Proof of Theorem 3.4

Let $X \sim \mu$, and denote

$$\bar{x} := \mathbb{E}X, \qquad \Sigma := \operatorname{Cov}(X) = \mathbb{E}\big[(X - \bar{x})(X - \bar{x})^\top\big].$$

By the trace identity,

$$\mathbb{E}\|X - \bar{x}\|^2 = \mathbb{E}\operatorname{tr}\big((X - \bar{x})(X - \bar{x})^\top\big) = \operatorname{tr}\Big(\mathbb{E}(X - \bar{x})(X - \bar{x})^\top\Big) = \operatorname{tr}(\Sigma).$$

Next, by the assumptions of Lemma 3.3, we have

$$\operatorname{tr}(\operatorname{Cov}(\Pi(X))) \leq C_1 k, \qquad \mathbb{E}\|X - \Pi(X)\|^2 \leq C_2(\delta^2 + \eta), \qquad \|\Sigma\| \geq c_0 > 0.$$

Therefore Lemma 3.3 yields

$$r_{\text{eff}}(\Sigma) = \frac{\operatorname{tr}(\Sigma)}{\|\Sigma\|} \leq \frac{2C_1}{c_0}\, k + \frac{2C_2}{c_0}\,(\delta^2 + \eta).$$

Since

$$\operatorname{tr}(\Sigma) = \|\Sigma\|\, r_{\text{eff}}(\Sigma),$$

it follows that

$$\operatorname{tr}(\Sigma) \leq \|\Sigma\| \left(\frac{2C_1}{c_0}\, k + \frac{2C_2}{c_0}\,(\delta^2 + \eta)\right).$$

Using $\mathbb{E}\|X - \bar{x}\|^2 = \operatorname{tr}(\Sigma)$, we obtain

$$\big(\mathbb{E}\|X - \bar{x}\|^2\big)^{1/2} = \sqrt{\operatorname{tr}(\Sigma)} \leq \|\Sigma\|^{1/2} \left(\frac{2C_1}{c_0}\, k + \frac{2C_2}{c_0}\,(\delta^2 + \eta)\right)^{1/2}.$$

It remains to prove the concentration statement. Since

$$p(x) = Z^{-1} e^{-U(x)}$$

and

$$\nabla^2 U(x) \succeq mI \qquad \text{for all } x \in \mathbb{R}^d,$$

the Bakry–Émery criterion implies that μ satisfies a logarithmic Sobolev inequality with constant of order $1/m$, up to the standard normalization convention. Consequently, by the Herbst argument, every 1-Lipschitz function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies a Gaussian concentration bound of the form

$$\mathbb{P}(|f(X) - \mathbb{E}f(X)| \geq t) \leq 2\exp(-cmt^2) \qquad \text{for all } t \geq 0,$$

where $c > 0$ is an absolute constant.

Now define

$$f(x) := \|x - \bar{x}\|.$$

For any $x, y \in \mathbb{R}^d$,

$$|f(x) - f(y)| = \big|\|x - \bar{x}\| - \|y - \bar{x}\|\big| \le \|x - y\|,$$

so f is 1-Lipschitz. Applying the preceding concentration bound to this f gives

$$\mathbb{P}(|\|X - \bar{x}\| - \mathbb{E}\|X - \bar{x}\|| \ge t) \le 2\exp(-cmt^2) \qquad \text{for all } t \ge 0.$$

This proves the claimed concentration inequality and completes the proof. □

Corollary A.3. *By Jensen's inequality,*

$$\mathbb{E}\|X - \mathbb{E}X\| \le \sqrt{\mathbb{E}\|X - \mathbb{E}X\|^2} = \sqrt{\mathrm{tr}(\Sigma)}.$$

Hence

$$\mathbb{P}\Big(\|X - \mathbb{E}X\| \ge \sqrt{\mathrm{tr}(\Sigma)} + t\Big) \le 2\exp(-cmt^2).$$

In particular, if δ, η are bounded by constants and $\|\Sigma\| \asymp 1$, then there exists a constant $C > 0$ such that

$$\sqrt{\mathrm{tr}(\Sigma)} \le C\sqrt{k},$$

and therefore

$$\mathbb{P}\Big(\|X - \mathbb{E}X\| \ge C\sqrt{k} + t\Big) \le 2\exp(-cmt^2).$$

B. Interpretation of the Geometric Proxies

In this appendix, we clarify why the empirical quantities used in our layer-selection procedure can be interpreted as proxies for the theoretical geometric terms introduced in Section 3.1. Our goal is not to recover the exact strong-convexity constant or the exact manifold dimension of the layerwise representation distribution. Rather, we seek observable quantities that capture the same *functional roles* in diffusion-friendly geometry.

Theoretical role of m. In our theory, the quantity m appears as the strong-convexity constant of the potential $U(x) = -\log p(x)$, namely

$$\nabla^2 U(x) \succeq mI.$$

This means that, in every direction, the potential has at least curvature m. A larger m implies stronger contraction of the corresponding Langevin dynamics and greater robustness to score perturbations. Therefore, from the perspective of diffusion, m measures how strongly the representation distribution exhibits restoring geometry toward high-density regions.

In practice, however, the exact density $p(x)$ of layer activations is unknown, and only a finite sample of activation vectors is available. As a result, neither $U(x)$ nor its Hessian $\nabla^2 U(x)$ can be computed exactly. This motivates the use of empirical curvature-related proxies.

Why $\hat{m}_{\mathrm{mono}}$ is an m-like quantity. Our monotonicity proxy is defined from the empirical covariance Σ and its precision matrix $P = \Sigma^{-1}$. For sampled pairs (x_i, x_j), we compute

$$m_{ij} = \frac{(x_i - x_j)^\top P (x_i - x_j)}{\|x_i - x_j\|^2},$$

and summarize these values by a robust statistic such as the median.

This quantity is closely related to the role of m in the Gaussian case. Indeed, if the layerwise representation distribution were exactly Gaussian with mean μ and precision matrix P, then

$$U(x) = \frac{1}{2}(x - \mu)^\top P (x - \mu) + \mathrm{const},$$

and hence

$$\nabla^2 U(x) = P.$$

In that setting, the strong-convexity constant is precisely

$$m = \lambda_{\min}(P).$$

Moreover, for any displacement vector δ, the Rayleigh quotient

$$\frac{\delta^\top P \delta}{\|\delta\|^2}$$

measures directional curvature under the quadratic potential. Therefore, $\hat{m}_{\mathrm{mono}}$ can be interpreted as an empirical summary of the typical directional stiffness of the representation space. Although it is not an exact lower bound on the Hessian, it captures the same global geometric intuition: layers with larger $\hat{m}_{\mathrm{mono}}$ behave as if they are embedded in a more strongly restoring global geometry.

Why $\hat{m}_{\mathrm{curv}}$ is also an m-like quantity. Our local curvature proxy is based on the covariance of local neighborhoods. For each anchor point, we compute the covariance matrix Σ_{local} of its k-nearest neighbors and define a local score proportional to

$$\frac{1}{\lambda_{\max}(\Sigma_{\mathrm{local}})}.$$

The layer-level statistic $\hat{m}_{\mathrm{curv}}$ is then obtained by aggregating these local values across anchors.

This proxy is motivated by a local quadratic approximation. If a small neighborhood of the representation distribution is approximately Gaussian, then its local density may be written as

$$p_{\mathrm{local}}(x) \propto \exp\left(-\frac{1}{2}(x-\mu_{\mathrm{local}})^\top P_{\mathrm{local}}(x-\mu_{\mathrm{local}})\right),$$

with $P_{\mathrm{local}} \approx \Sigma_{\mathrm{local}}^{-1}$. Under this approximation, the local Hessian of the negative log-density is given by P_{local}, and the corresponding local strong-convexity scale is

$$\lambda_{\min}(P_{\mathrm{local}}) = \lambda_{\min}(\Sigma_{\mathrm{local}}^{-1}) = \frac{1}{\lambda_{\max}(\Sigma_{\mathrm{local}})}.$$

Thus, $\hat{m}_{\mathrm{curv}}$ measures how compact and sharply curved the local representation neighborhoods are. Larger values indicate smaller local spread along the most variable direction, which is consistent with the intuition of stronger local restoring geometry.

Why $\hat{k}$ reflects intrinsic dimension. The theoretical quantity k is intended to capture the intrinsic complexity of the representation distribution, rather than its ambient dimension d. In our empirical procedure, we measure this using the effective rank of the covariance:

$$\hat{k} = r_{\mathrm{eff}}(\Sigma) = \frac{\mathrm{tr}(\Sigma)}{\|\Sigma\|}.$$

This quantity has a direct interpretation in idealized low-dimensional settings. Suppose the data are supported exactly on a k-dimensional isotropic subspace with covariance eigenvalues

$$\lambda_1 = \cdots = \lambda_k = \sigma^2, \qquad \lambda_{k+1} = \cdots = \lambda_d = 0.$$

Then

$$r_{\mathrm{eff}}(\Sigma) = \frac{k\sigma^2}{\sigma^2} = k.$$

Hence, in this ideal case, the effective rank exactly recovers the intrinsic dimension.

More generally, when the representation distribution is concentrated near a low-dimensional manifold or subspace, the covariance spectrum typically contains a small number of dominant eigenvalues and many small residual ones. In such cases, $r_{\mathrm{eff}}(\Sigma)$ measures the effective number of active variation directions. For our purposes, this is the relevant notion of intrinsic dimension, since diffusion complexity depends not on the nominal ambient dimension but on how many directions carry meaningful variation.

Interpretation and limitation. Taken together, $\hat{m}_{\mathrm{mono}}$ and $\hat{m}_{\mathrm{curv}}$ serve as global and local proxies for curvature-related restoring geometry, while $\hat{k}$ serves as an operational measure of intrinsic dimension. These quantities are not exact estimators of the theoretical constants in Section 3.1. Instead, they are empirical surrogates designed to preserve the same geometric design principle: diffusion-friendly layers should be locally compact, globally stable, and effectively low-dimensional.

This distinction is important. Our empirical layer-selection score should therefore be interpreted as an operational criterion motivated by theory, rather than as a direct numerical estimate of the exact constants appearing in the theorems.

C. Geometric Proxy Estimation and Layer Selection

This appendix describes the implementation details of the empirical geometric proxies used in the *Locate* stage, together with the practical layer-selection procedure. Where Appendix B explains why these quantities can be interpreted as theory-motivated surrogates for curvature and intrinsic dimension, the present appendix focuses on how they are computed in practice and how they are combined into a robust empirical selection rule.

C.1. Representation Extraction

For each transformer layer, we begin from hidden activations of shape

$$\mathrm{acts} \in \mathbb{R}^{N \times S \times H},$$

where N denotes the number of sequences, S the sequence length, and H the hidden dimension. For example, a layer output may have shape $[B, S, H] = [32, 1024, 4096]$.

To compute geometric statistics, we convert these activations into a set of representation vectors

$$x \in \mathbb{R}^{M \times D},$$

using an extraction function that supports several pooling modes.

Mean pooling. In the `mean` setting, each sequence is mapped to a single vector by masked averaging over valid tokens:

$$x_n = \frac{\sum_{t=1}^{S} a_{nt}\, h_{nt}}{\sum_{t=1}^{S} a_{nt}},$$

where $a_{nt} \in \{0, 1\}$ is the attention mask and $h_{nt} \in \mathbb{R}^H$ is the hidden state at token position t. This yields one representation vector per sequence.

Last-token pooling. In the `last` setting, we take the hidden state of the final valid token in each sequence. This again yields one vector per sequence, while preserving a token-level representation associated with the sequence endpoint.

Tokenwise extraction. In the `token` setting, all valid token representations are flattened across sequences and positions. If necessary, a random subset is retained to control memory and computational cost. This yields a larger collection of vectors and allows the layer geometry to be estimated directly from tokenwise activations.

Mask handling. Padding positions with `attention_mask`=0 are excluded from all computations. Special tokens are retained whenever they are marked as valid by the attention mask. Thus, the extracted representation set reflects the actual active tokenwise computation of the model.

Recorded representation statistics. For transparency and reproducibility, we store layerwise summary statistics describing the extracted representation set, including the number of sequences, sequence length, hidden dimension, number of vectors before and after projection, and the corresponding feature dimensions. These quantities make it possible to verify that proxy estimation is based on comparable representations across layers. In the main experiments, we use `mean` pooling and retain at most `max_seqs`=2,000 sequences and `max_tokens`=200,000 valid tokens per layerwise geometry run.

C.2. Preprocessing and Random Projection

Because hidden representations can be high-dimensional, we optionally apply random projection before estimating the geometric proxies. This serves two purposes: it improves numerical stability in covariance-based calculations, and it reduces the computational cost of repeated layerwise estimation.

Given vectors $x \in \mathbb{R}^{M \times D}$, we construct a Gaussian random matrix

$$R \in \mathbb{R}^{D \times d_{\mathrm{proj}}},$$

followed by QR orthonormalization to obtain an approximately orthonormal projection basis. The projected representations are then

$$x_{\mathrm{proj}} = xR \in \mathbb{R}^{M \times d_{\mathrm{proj}}}.$$

If the projection dimension satisfies $d_{\mathrm{proj}} \leq 0$ or $d_{\mathrm{proj}} \geq D$, projection is skipped and the original vectors are used. This design ensures that the projection acts only as a computational device and does not alter the pipeline when dimensionality reduction is unnecessary.

The main hyperparameters governing this step are the projection dimension, the pooling mode, the maximum number of sequences or tokens retained, and the ridge regularization used in subsequent covariance inversion.

C.3. Local Curvature Proxy

To capture local geometric compactness, we estimate a curvature-inspired proxy from neighborhoods in representation space. For a given layer representation set $x = \{x_1, \ldots, x_M\} \subset \mathbb{R}^D$, we sample a set of anchor points and, for each anchor, identify its k nearest neighbors under Euclidean distance. Distances are computed on the same representation matrix used for proxy estimation, i.e., after the optional projection step when projection is enabled.

Let $\mathcal{N}_i$ denote the neighborhood of anchor x_i. We compute the local covariance matrix

$$\Sigma^{(i)}_{\mathrm{local}} = \mathrm{Cov}(\mathcal{N}_i) + \lambda I,$$

where $\lambda > 0$ is a small ridge term added for numerical stability. The local curvature score is then defined as

$$m^{(i)}_{\mathrm{curv}} = \frac{1}{\lambda_{\max}\left(\Sigma^{(i)}_{\mathrm{local}}\right)}.$$

Intuitively, this quantity is large when the local neighborhood is compact even along its most variable direction, which is consistent with the notion of strong local restoring geometry.

To obtain a layer-level statistic, we aggregate the anchorwise values using a robust summary:

$$\hat{m}_{\mathrm{curv}} = \mathrm{median}_i\big(m^{(i)}_{\mathrm{curv}}\big).$$

In addition, we record quantiles such as the 25th, 50th, and 75th percentiles in order to characterize variation across local neighborhoods and to support uncertainty-aware visualization.

C.4. Monotonicity Proxy

To capture a global notion of directional stiffness, we compute a monotonicity-inspired proxy from the empirical covariance of the layer representations. Let

$$\Sigma = \mathrm{Cov}(x) + \lambda I$$

be the ridge-regularized empirical covariance and let

$$P = \Sigma^{-1}$$

denote its precision matrix.

We then sample random pairs (x_i, x_j) and compute

$$m_{ij} = \frac{(x_i - x_j)^\top P(x_i - x_j)}{\|x_i - x_j\|^2}.$$

This is a Rayleigh-quotient-type quantity that measures how strongly a pairwise displacement is penalized by the global precision geometry.

The layer-level monotonicity proxy is defined by a robust summary over sampled pairs:

$$\hat{m}_{\mathrm{mono}} = \mathrm{median}_{(i,j)}(m_{ij}).$$

As with the local curvature proxy, we additionally record quantiles and confidence intervals obtained by bootstrap resampling. In the main experiments, we report 95% bootstrap percentile intervals for the median monotonicity estimate. These summaries allow us to assess not only the central tendency of global directional stiffness, but also its stability under finite-sample variation.

C.5. Effective Rank

To estimate the effective intrinsic dimension of the layerwise representation distribution, we compute the effective rank of the empirical covariance:

$$\hat{k} = r_{\mathrm{eff}}(\Sigma) = \frac{\mathrm{tr}(\Sigma)}{\|\Sigma\|},$$

where $\|\Sigma\|$ denotes the spectral norm, i.e., the largest eigenvalue of Σ.

This quantity measures the effective number of active directions of variation in the representation space. A smaller value indicates that the representation is concentrated along fewer dominant directions, which is favorable from the perspective of diffusion complexity.

In the main text, we use effective rank as the primary intrinsic-dimension statistic. In addition to the central estimate $\hat{k}$, we optionally record related quantities such as the participation ratio and bootstrap summaries as supplementary diagnostics. To reflect finite-sample uncertainty, we also store quantiles such as

$$\hat{k}_{\mathrm{q25}}, \quad \hat{k}_{\mathrm{q50}}, \quad \hat{k}_{\mathrm{q75}}.$$

C.6. Selection Score Construction

After computing $\hat{m}_{\mathrm{curv}}$, $\hat{m}_{\mathrm{mono}}$, and $\hat{k}$ for each layer, we combine them into a single empirical layer-selection score.

Baseline score. A simple baseline motivated by the curvature–dimension principle is

$$\mathrm{selection\_score}_{\mathrm{base}}(\ell) = z(\log \hat{m}_{\mathrm{curv}}(\ell)) - z(\log \hat{k}(\ell)),$$

where $z(\cdot)$ denotes z-score normalization across layers within the same model. Intuitively, the baseline rewards curvature-related structure while penalizing excessive effective dimension.

Final score used in practice. In the final implementation, we use a fixed score based on log-transformed layerwise statistics:

$$\mathrm{selection\_score}(\ell) = \alpha_1 z(\log \hat{m}_{\mathrm{curv}}(\ell)) + \alpha_2 z(\log \hat{m}_{\mathrm{mono}}(\ell)) + \alpha_3 z(\log \hat{k}(\ell)).$$

We use the fixed coefficients

$$(\alpha_1, \alpha_2, \alpha_3) = (1.0,\ 1.0,\ -1.0).$$

Thus, the final score becomes

$$\mathrm{selection\_score}(\ell) = z(\log \hat{m}_{\mathrm{curv}}(\ell)) + z(\log \hat{m}_{\mathrm{mono}}(\ell)) - z(\log \hat{k}(\ell)).$$

This score reflects the curvature–dimension principle. First, both local and global curvature-related quantities are rewarded through $\hat{m}_{\mathrm{curv}}$ and $\hat{m}_{\mathrm{mono}}$. Second, effective intrinsic dimension is penalized through $\hat{k}$. We use only a

Preset	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	ℓ^*	$\mathcal{L}(\ell^*)$	oracle ℓ	gap	Spearman
final, no k^2	(1.0,1.0,-1.0,0.0)	2	0.059	1	0.028	0.940
baseline w/ k^2	(1.0,1.0,-1.0,-0.5)	2	0.087	1	0.028	0.859
curv×0.5	(0.5,1.0,-1.0,-0.5)	6	139.092	1	139.033	0.729
curv×1.5	(1.5,1.0,-1.0,-0.5)	2	0.087	1	0.028	0.916
mono×0.5	(1.0,0.5,-1.0,-0.5)	6	139.092	1	139.033	0.733
mono×1.5	(1.0,1.5,-1.0,-0.5)	2	0.087	1	0.028	0.916
k_{lin}×0.5	(1.0,1.0,-0.5,-0.5)	2	0.087	1	0.028	0.950
k_{lin}×1.5	(1.0,1.0,-1.5,-0.5)	6	139.092	1	139.033	0.725
k_{sq}×0.5	(1.0,1.0,-1.0,-0.25)	2	0.087	1	0.028	0.898
k_{sq}×2.0	(1.0,1.0,-1.0,-1.0)	6	139.092	1	139.033	0.733

Table 5. Sensitivity of layer selection to coefficient perturbations. Numbers are produced by our analysis script. The final score removes the quadratic effective-rank penalty and uses only a linear curvature–dimension trade-off.

linear effective-rank penalty because the sensitivity analysis below shows that an additional quadratic penalty can over-penalize some low-rank layers and lead to unstable layer selection.

The selected target layer is then defined by

$$\ell^* = \arg\max_{\ell} \text{selection\_score}(\ell).$$

For convenience, we also report

$$\text{predicted\_loss}(\ell) = -\,\text{selection\_score}(\ell),$$

so that lower predicted loss corresponds to a more diffusion-friendly layer.

Sensitivity to coefficient choice. To address concerns that the fixed coefficients might implicitly overfit, we evaluate a small set of coefficient perturbations around the final score and report (i) the selected layer ℓ^*, (ii) the validation loss at ℓ^*, and (iii) rank correlations between the score and the validation loss. Table 5 summarizes the results computed from `layerwise_geometry.json` and `teacher_loss_frozen.csv` (excluding layer 0). The final linear score selects the oracle-best layer in this sweep while maintaining a strong monotonic relationship with validation bridge loss. By contrast, adding or strengthening the quadratic effective-rank penalty can shift the selected layer from early layers to layer 6, which yields a large degradation in validation loss in this setting. This suggests that effective rank is useful as a soft complexity penalty, but overly strong rank penalties can hurt top-layer selection.

C.7. Implementation Details of the Diffusion Bridge

In the embedding-conditioned setting used in our experiments, the conditioning signal $c(x)$ is the dumped activation tensor $\text{embedding\_out} \in \mathbb{R}^{B\times S\times H}$, i.e., the hidden state immediately before the first transformer block of the source language model. This tensor is passed through a learned condition projection and mapped to the 768-dimensional $\text{encoder\_hidden\_states}$ interface expected by the pretrained UNet. Thus, the diffusion backbone is used as a conditional latent denoiser over language-model hidden states rather than as a text-to-image model.

To instantiate the bridge, we adopt a Stable-Diffusion-style latent denoising architecture built on a pretrained UNet. The target hidden state h_{ℓ^*} is reshaped and projected into an image-like tensor, encoded by a frozen VAE into a latent z_{ℓ^*}, noised to z_t at timestep t using a fixed diffusion scheduler, and denoised by the UNet to predict $\hat{\epsilon}_\theta(z_t, t, c)$. The denoised latent is then decoded and projected back to the original hidden space to obtain the reconstructed hidden state $\hat{h}_{\ell^*}$.

Hidden-to-image layout. Given a selected-layer hidden state $h_{\ell^*} \in \mathbb{R}^{B\times S\times H}$, we first reshape the sequence dimension into a fixed spatial grid and apply a learned 1×1 projection to map the hidden dimension to the channel dimension required by the latent denoising backbone. This produces an image-like tensor of shape $(C, H_{\text{img}}, W_{\text{img}})$. The reshape is deterministic and does not use image supervision. After denoising, an inverse learned projection maps the output back to $\mathbb{R}^{B\times S\times H}$.

In our experiments with $S = 1024$ and hidden size $H = 4096$, we use $H_{\text{img}} = 32$, $W_{\text{img}} = 32$, and $C = 3$, matching

the VAE input-channel interface of SD-v1.5. This channel projection should be understood as an architectural interface to the pretrained latent denoising stack, not as an assumption that language hidden states are naturally RGB images.

We do not assume that language hidden states have natural 2D image semantics. The Stable-Diffusion-style backbone is used only as a high-capacity conditional denoiser. The 2D layout is therefore an architectural interface rather than a semantic image representation. To test this design choice, we compare it against sequence-native deterministic and denoising backbones in Appendix D.1.

For the late-stage alignment losses, the language-modeling term is defined as $\mathcal{L}_{\mathrm{LM}} = \mathrm{CE}(p_{\mathrm{bridge}}(x_{i+1} \mid x_{\leq i}), x_{i+1})$, and the temperature-scaled distillation term is $\mathcal{L}_{\mathrm{KD}} = T^2\,\mathrm{KL}(\mathrm{softmax}(o_{\mathrm{teacher}}/T) \,\|\, \mathrm{softmax}(o_{\mathrm{bridge}}/T))$, where o_{teacher} and o_{bridge} denote the teacher and bridge logits, respectively.

During training, we update the UNet denoiser together with the bridge-specific projection modules, including the hidden-to-image mapping, the image-to-hidden mapping, and the condition projection. By contrast, the VAE is kept frozen and the diffusion scheduler is fixed. Under the embedding-conditioned setting described above, no Stable-Diffusion text-conditioning pathway is used. This isolates the Replace stage as a hidden-space learning problem: the bridge is trained to reproduce the selected-layer hidden state while remaining compatible with the retained upper transformer stack.

D. Experimental Setup Details

This appendix provides experimental and implementation details supplementing Section 4. It describes the data sampling pipeline, representation extraction, geometric proxy estimation, and optimization hyperparameters used for the layer-sweep and full-training experiments.

Data sampling and activation dumps. We construct the activation corpus from a local raw-text copy of *Dolma v1.7*. The dump pipeline samples up to 300,000 text sequences using stratified round-robin sampling over 700 source files, with up to 500 samples per file and random seed 42. Each sequence is tokenized with the corresponding source-model tokenizer and passed through the backbone model to save the embedding output, all decoder-layer outputs, input IDs, and attention masks. Activations are computed in reduced precision and stored as float16 shards for subsequent geometry estimation and bridge training.

Representation extraction. Sequences are tokenized with the source tokenizer under its default special-token handling, so special tokens are retained in the stored input IDs and activation tensors. Padding positions are tracked by the attention mask and excluded whenever masked averaging is used. When token IDs are decoded back into text for prompt reconstruction, special tokens are removed. For geometry estimation, the implementation supports last-token pooling and token-level sampling, but we use mean pooling by default.

Geometry estimation. The default geometry pipeline uses no random projection ($d_{\mathrm{proj}} = 0$), ridge coefficient 10^{-3}, $k = 64$ nearest neighbors for local curvature, 512 anchor points, 200,000 sampled pairs for monotonicity, and bootstrap resampling with 95% confidence intervals. In the main experiments, layerwise geometry proxies are estimated from repeated subsamples as described in Section 4.1.

Bridge training. Bridge training uses a deterministic shard-level split with validation ratio 0.1 and split seed 42. In the fixed-budget layer sweep, each candidate layer is trained for one epoch with up to 150,000 training examples, batch size 4, learning rate 3×10^{-5}, AdamW optimization, mixed-precision FP16, and a maximum of 37,500 optimization steps. We use bridgeability to mean how easily a layer's hidden state can be reconstructed by the diffusion bridge under this fixed budget, and measure it by validation bridge loss. Training losses are reported as auxiliary optimization diagnostics. The codebase also supports applying the auxiliary next-token language-modeling loss $\mathcal{L}_{\mathrm{LM}}$ and teacher-student distillation loss $\mathcal{L}_{\mathrm{KD}}$ only in the final epoch.

Compute resources. The fixed-budget layer-sweep experiments were conducted on NVIDIA H100 GPUs with a matched 40 GPU-hour budget. Each candidate layer was trained for one epoch on 150K examples using batch size 4 and mixed-precision FP16. The final full-training experiments, which are used for the main model-quality evaluation, were conducted on NVIDIA B200 GPUs for 40 hours per main run, training the selected bridge for four epochs on

Table 6. Validation bridge loss on layer 16 (S=512, N=500).

Bridge	Layer	Split	S	N	Mean loss ↓
stable_diffusion_UNet	16	valid	512	500	1163.587321
ddpm_hidden	16	valid	512	500	1411.363999
ddpm_hidden_transformer	16	valid	512	500	2799.283554
ddpm_hidden_conv1d	16	valid	512	500	3672.299440

Method	Trainable params	Active bridge/module params
DiHAL	862.7M	946.3M bridge
Diffusion-LM	93.7M	93.7M
SED	73.7M	73.7M
LD4LG	25.6M AE + 188M diffusion	25.6M AE + 188M diffusion
CoDAR	130M diffusion + 230.9M AR decoder	360.9M

Table 7. Trainable and active parameter counts for the same-budget representation-space comparison. Counts are measured or estimated under our experimental configurations. Trainable parameters are updated by the optimizer, while active bridge/module parameters are used in the forward pass, including frozen modules where applicable. For CoDAR, the count is estimated from the reported MDLM-style diffusion backbone and GPT-2-small-style autoregressive decoder with cross-attention under the Qwen tokenizer vocabulary.

the 300K-example corpus. We additionally report peak GPU memory, throughput, and latency in the final evaluation table. Activation dumping stores float16 hidden states for the evaluated 8B-scale backbones, and therefore storage requirements scale approximately linearly with the number of layers and sampled sequences.

D.1. Diffusion Bridge Architecture and Backbone Choices

Backbone choice. We choose a Stable-Diffusion-style UNet bridge because the replacement module must solve a high-dimensional conditional denoising problem over continuous representations, rather than a standard next-token prediction task. The target is an internal boundary hidden state h_{ℓ^*}, and the conditioning signal is the embedding-derived representation of the same input. We therefore require a backbone that can denoise structured, high-dimensional activations while exploiting the conditioning signal effectively.

Table 6 compares several bridge backbones under the same validation setting on layer 16 (S=512, N=500). The Stable-Diffusion-style UNet bridge achieves the lowest validation loss among the evaluated trainable alternatives. Its mean loss (1163.59) is lower than the MLP-based hidden DDPM bridge (1411.36), corresponding to a 17.6% reduction. It also substantially outperforms the Transformer-based and 1D-convolutional bridges, reducing validation loss by 58.4% relative to `ddpm_hidden_transformer` and 68.3% relative to `ddpm_hidden_conv1d`.

These results suggest that reusing the Stable-Diffusion UNet as a conditional denoising backbone is a practical choice in our setting. We do not claim that this architecture is optimal in general; rather, it provides the strongest validation performance among the tested backbones under the same small-scale ablation. Importantly, the model is not used as an image generator and does not rely on image supervision; we only reuse the latent denoising structure as a conditional backbone for reconstructing language-model hidden states.

E. Baseline Details

Parameter accounting. Table 7 reports parameter counts for the same-budget representation-space comparison. These counts are based on the current implementation and notebook configuration used in our experiments. Trainable parameters are those updated by the optimizer. Active bridge/module parameters are parameters used in the forward pass for the corresponding diffusion or recovery module, including frozen modules when applicable. For DiHAL, the retained pretrained upper transformer layers and original LM head are not counted as trainable bridge parameters, but they are part of the active language-modeling interface.

Table 8. End-to-end inference cost. DiHAL cost includes bridge denoising, retained upper transformer layers, and the LM head.

Model	Method	Insert. layer	NFEs	Latency/token ↓	Throughput ↑	Peak mem. ↓
Llama-3.1-8B	Original	–	–	0.043	23509	16.2
Llama-3.1-8B	DiHAL	3	1	0.074	13481	18.0
Llama-3.1-8B	DiHAL	3	4	0.150	6681	18.0
Llama-3.1-8B	DiHAL	3	20	0.503	1989	18.0
Llama-3.1-8B	DiHAL	10	1	0.065	15314	18.0
Llama-3.1-8B	DiHAL	10	4	0.146	6844	18.0
Llama-3.1-8B	DiHAL	10	20	0.515	1941	18.0
Llama-3.1-8B	DiHAL	20	1	0.051	19467	18.0
Llama-3.1-8B	DiHAL	20	4	0.116	8597	18.0
Llama-3.1-8B	DiHAL	20	20	0.503	1988	18.0
Llama-3.1-8B	DiHAL	30	1	0.035	28676	18.0
Llama-3.1-8B	DiHAL	30	4	0.114	8798	18.0
Llama-3.1-8B	DiHAL	30	20	0.502	1993	18.0
Qwen3-8B	Original	–	–	0.059	17033	15.9
Qwen3-8B	DiHAL	2	1	0.081	12375	17.6
Qwen3-8B	DiHAL	2	4	0.130	7668	17.6
Qwen3-8B	DiHAL	2	20	0.503	1989	17.6
Qwen3-8B	DiHAL	12	1	0.066	15222	17.6
Qwen3-8B	DiHAL	12	4	0.132	7554	17.6
Qwen3-8B	DiHAL	12	20	0.506	1977	17.6
Qwen3-8B	DiHAL	22	1	0.047	21156	17.6
Qwen3-8B	DiHAL	22	4	0.119	8428	17.6
Qwen3-8B	DiHAL	22	20	0.500	2002	17.6
Qwen3-8B	DiHAL	32	1	0.036	27581	17.6
Qwen3-8B	DiHAL	32	4	0.093	10769	17.6
Qwen3-8B	DiHAL	32	20	0.500	2001	17.6

F. Inference Cost Detail

Because `DiHAL` replaces a transformer prefix with a diffusion bridge, its end-to-end inference cost depends on two factors: the insertion depth and the number of denoising steps. We therefore measure latency, throughput, peak memory, and the number of function evaluations (NFEs) across several insertion layers. The measurement includes the full hybrid inference path: bridge denoising, retained upper transformer layers, and the original LM head.

Table 8 shows the resulting cost trade-off. For the geometry-selected shallow insertion layers, `DiHAL` is slower than the original backbone even with a single denoising step, and latency increases substantially as NFEs grow. This indicates that bridge denoising is the dominant overhead in the current implementation. Deeper insertion layers can reduce latency at NFE=1 by skipping more transformer layers, but these layers are not selected by the geometry criterion and are less reliable as hidden-state reconstruction targets. Peak memory also increases because the diffusion bridge adds extra active modules.

Overall, these results support our limitation claim that the current `DiHAL` implementation should not be interpreted as an end-to-end acceleration method. Instead, the table clarifies the cost-quality trade-off: deeper replacement can reduce retained-transformer computation, but diffusion denoising overhead and hidden-state reconstruction difficulty limit practical acceleration in the present setting.

G. Existing assets and licenses.

Our experiments use publicly available pretrained backbones, datasets, and model components. We use Llama-3.1-8B-Instruct under the Llama 3.1 Community License and Qwen3-8B under the Apache 2.0 license. We use Dolma v1.7 for activation extraction and bridge training under the ODC-BY license, and WikiText-103 for evaluation under its Creative Commons/GFDL licensing terms. The diffusion bridge reuses Stable Diffusion v1.5 components under the CreativeML OpenRAIL-M license, and generative perplexity is computed using a GPT-2 evaluator under the GPT-2 license terms. We cite the original papers and model or dataset sources for all existing assets and use them only for research evaluation consistent with their respective terms.