Subliminal Clocks: Latent Time Modelling in Diffusion Language Models

Maximo Rulli[1] Thomas Fontanari[1,*] Simone Petruzzi[1,*] Federico Alvetreti[1]
Giorgio Strano[1] Donato Crisostomi[1] Giorgos Nikolaou[2] Tommaso Mencattini[2]
Andrea Santilli[3,†] Emanuele Rodolà[1] Simone Scardapane[1] Alessio Devoto[3,†]

Abstract

Diffusion Language Models (DLMs) have recently emerged as a promising alternative to autoregressive models. Unlike standard diffusion-based approaches, DLMs are not explicitly conditioned on a timestep, raising a natural question: do these models internally represent denoising progress, and how is such information used downstream? In this work, we show that DLMs do in fact encode a latent representation related to the diffusion timestep within their residual streams. We find that this signal can be reliably extracted using probes across layers, indicating that denoising progress is decodable from internal activations. We further demonstrate that steering the model along a low-dimensional subspace associated with the inferred timestep allows us to systematically modulate its notion of denoising progress, leading to predictable changes in model confidence and entropy. Finally, we analyse the geometry of the identified representation, showing that it exhibits structured and interpretable properties in activation space, and shedding light on how such a signal is processed by these models.

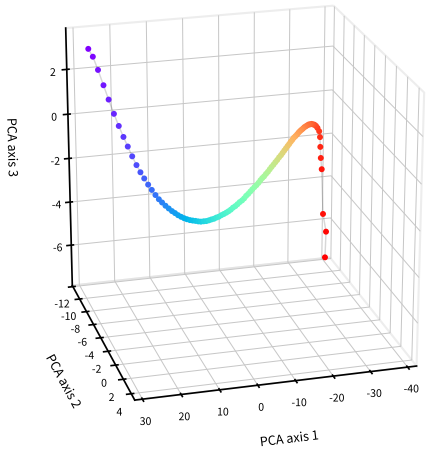


Figure 1: **3D projection of latent denoising step modelling for LLaDA**. LLaDA represents its τ subspace as a low-dimensional manifold-like curve, progressively modelling the denoising progress from all [MASK] (red) to no remaining [MASK] (purple).

1 Introduction

Large Language Models (LLMs) (Anthropic, 2025; Meta AI, 2025; Achiam et al., 2023; DeepSeek-AI, 2024; Gemma Team, 2025) are driving a paradigm shift across various scientific and social domains. Understanding the capabilities of these models remains an active area of research. Many works aim to characterise the manifolds and subspaces spanned by the features of these models' activations (Saglam et al., 2026; Tiblias et al., 2026; Joshi et al., 2025; Shai et al., 2025; Bhalla et al., 2026), helping explain how they represent and transform concepts across their computations. Recently, large-scale Masked Diffusion Language Models (Nie et al., 2025; Zhu et al., 2026; Ye et al., 2025) have emerged as an alternative paradigm for text generation. Rather than generating strictly left-to-right, they progressively denoise [MASK] tokens in a BERT-style manner (Devlin et al., 2019); throughout this work, we call this masked-token family DLMs. Most existing research has primarily focused on improving their efficiency and generation quality (Wu et al., 2025; Arriola et al., 2025; Du et al., 2026; Zekri et al., 2026; Yu et al., 2026). In contrast, their internal mechanisms and representations remain largely unexplored. More recently, a small but growing body of work has begun investigating the interpretability of these models. For instance, Wang et al. (2026) train Sparse Autoencoders (Cunningham et al., 2023) on LLaDA and Dream to enable concept steering at inference time,

*Equal contribution.
[1]Sapienza University of Rome. [2]EPFL. [3]NVIDIA.
†Work performed while at Sapienza University of Rome.
Correspondence: rulli.2154435@studenti.uniroma1.it.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 39904–39927
October 24-29, 2026 ©2026 Association for Computational Linguistics

Rulli et al. (2025) analyse how attention sinks move across denoising steps and how robust DLMs are to pruning them, Wen et al. (2026) demonstrate how [MASK] tokens can be manipulated to induce harmful or deceptive generations, and Piskorz et al. (2025) show that DLM performance is strongly influenced by the number of [MASK] tokens present at each denoising step. While these works provide valuable insights into the behaviour of DLMs, the internal dynamics underlying the denoising process itself remain poorly understood. In particular, it remains unclear whether DLMs internally represent denoising progress during generation. Previous theoretical work suggests that the strong performance of absorbing-state DLMs may be largely attributed to the special role of the [MASK] token: the reverse process only needs to denoise masked positions, and the corresponding score can be expressed as a clean-data conditional distribution up to an analytic time-dependent factor (Ou et al., 2024). Complementarily, masked diffusion objectives can be reformulated in terms of the number of masked tokens, making continuous denoising time closely related to a relaxation of the mask ratio and connecting these models to time-agnostic masked models (Zheng et al., 2025). This makes the fraction of masked tokens a natural observable proxy for denoising progress, but leaves open whether and how such a signal is internally represented by the model.

In this work, we investigate this through the following three research questions:

RQ1

Do Diffusion Language Models internally represent a signal related to the denoising step?

RQ2

Is the identified signal important to the generation process? How does modifying it affect the models' downstream computations?

RQ3

What characteristics does the signal exhibit? Is there a unified model-level representation?

We address these questions in three stages: first by probing for denoising-time information, then by causally steering the recovered signal, and finally by characterising its geometry across layers. We conduct our analysis on LLaDA-1.5 (Zhu et al., 2026) (which we refer to as LLaDA) and Dream (Ye et al., 2025), two representative large-scale masked diffusion language models.

2 Background

DLMs are trained to recover a clean sequence x_0 from a corrupted version of it (Li et al., 2022; Sahoo et al., 2024; Ou et al., 2024; Shi et al., 2025). In absorbing-state models, the corrupted sequence x_s is obtained from x_0 by replacing a subset of its positions with a special [MASK] token. We define:

$$\mathcal{M}(x_s) = \{j : x_s^j = [\text{MASK}]\}$$

to be the set of masked positions. The scalar $s \in [0, 1]$ controls the corruption level: $s = 0$ recovers the clean sequence, while $s = 1$ masks every position. During training, a noise level $s \sim \mathcal{U}(0, 1]$ is sampled and each position is corrupted independently according to the forward process

$$x_s^j \sim \text{Cat}\big(\cdot; p = (1 - s)\, x_0^j + s\, [\text{MASK}]\big)$$

so that position j keeps its original token with probability $1 - s$ and is replaced by [MASK] with probability s. The DLM p_θ takes the corrupted sequence x_s as input and predicts, at every position j, a distribution $p_\theta(x_0^j \mid x_s)$ over the original vocabulary $\mathcal{V}$. The training minimises a cross-entropy loss applied only at the masked positions, reweighted by $1/s$:

$$-\mathbb{E}_{x_0,\, s,\, x_s}\left[\frac{1}{s} \sum_{j \in \mathcal{M}(x_s)} \log p_\theta(x_0^j \mid x_s)\right]. \quad (1)$$

Inference proceeds through a sequence of T denoising steps. Generation is initialised at step $t = 0$ with L [MASK] tokens appended to the prompt, and ends at $t = T$ with all tokens unmasked. At each step t, the model operates on the current partially-masked sequence x_t to obtain $p_\theta(\cdot \mid x_t)$ at every masked position, and a subset of $\mathcal{M}(x_t)$ is selected to be unmasked by sampling according to an unmasking policy. This policy determines how many tokens are revealed per step. At inference time we measure the fraction of unmasked tokens as

$$\tau_t := 1 - \frac{|\mathcal{M}(x_t)|}{L}. \quad (2)$$

In Appendix A, we provide more details of how τ_t is equivalent, in expectation, to the complement of the continuous diffusion-time variable s.

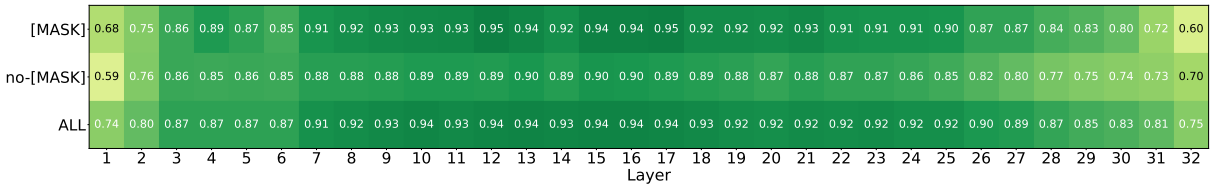


Figure 2: **MLP probes recover the τ signal**. The R^2 coefficient (higher is better) degrades as we probe deeper into the model, to the point where the MSE almost reduces to half the variance of τ. Moreover, both [MASK] and non-[MASK] tokens seem to carry this information, attaining similarly high coefficients.

We therefore interpret τ_t as an empirical measure of denoising progress and use it as a proxy for the notion of denoising time in masked diffusion language models.

3 Recovering τ

In this section we address RQ1 by attempting to recover the τ signal from the model's hidden states. To test whether τ can be recovered from the residual stream of the model, we train MLP probes to predict the current τ_t of a given sequence by using a residual-stream hidden state as input. Specifically, we fit an MLP ($\phi_l(\cdot) : \mathbb{R}^d \to (0, 1)$) for each layer l in the model, where d is the dimensionality of the hidden state (see Appendix B for architectural and training details). Each MLP is trained to minimise the regression loss

$$\mathcal{L}_{\text{MLP}} = \mathbb{E}_{t,n,j}\left[\left(\tau_t - \phi_l(h^j_{t,l,n})\right)^2\right]$$

where n is the example index, j is the token index in the example sequence, and $h^j_{t,l,n}$ denotes the hidden state. Throughout this work, we use $\mathbb{E}_r$ to denote the empirical expectation over the corresponding quantity r. We fit separate probes for every layer in both LLaDA and Dream. Additionally, we vary the subset of tokens used for training, considering only [MASK] tokens, non-[MASK] tokens, or all tokens. This allows us to investigate whether information related to τ is preferentially encoded in specific types of tokens. Figure 2 shows the R^2 obtained for each layer of LLaDA. The probes maintain an $R^2 > 0.5$ across all layers of the model, indicating that information about τ is consistently represented throughout the network depth. Performance slightly degrades in the earliest and latest layers, suggesting that the representation becomes less accessible at the boundaries of the computation. We observe similar trends for Dream (see Appendix C).

We also find that activations corresponding to [MASK] tokens yield marginally better predictions than non-[MASK] tokens, though both token types support accurate recovery. This suggests that τ information is not exclusively localised to masked positions, but is instead distributed across the residual stream.

Importantly, each probe receives a single contextualised hidden state, without direct access to the full sequence or neighbouring activations; hinting that: individual contextualised token representations themselves encode substantial information about τ. Thus, the high performance of the probes indicates that individual token representations **carry a latent representation** of the sequence-level statistic τ.

As a result, we answer affirmatively to RQ1 and conclude that DLMs do internally model a signal related to the current denoising step. In the next section, we show a natural construction of a signal related to τ and use it to steer the model at inference time.

✓ Takeaway for RQ1

Diffusion Language Models do internally encode a denoising-step-related signal, and we can accurately recover it.

4 Assessing the Signal's Importance

We now investigate whether the found signal can be explicitly extracted and used to steer the model. In particular, we aim to identify directions in activation space associated with the denoising-time statistic τ, and study whether steering along these directions changes the model's behaviour during inference.

4.1 Approximating τ

To characterise and approximate the internal representation of τ, we turn to mean activation vectors (Gurnee et al., 2025; Wang et al., 2025; Panickssery et al., 2024). The key intuition behind this approach is that averaging activations over many

samples suppresses instance-specific information while preserving systematic signals shared across examples, such as the one associated with τ.

Following Wang et al. (2025), we compute mean activation vectors over all hidden states in the response window, grouping them according to their denoising-progress bin. More formally, for each denoising-progress bin t and layer l, we define:

$$\mu_{t,l} := \mathbb{E}_{n,j}\left[h^j_{t,l,n}\right], \tag{3}$$

We discretise the denoising process into 100 bins, yielding 100 mean vectors per layer. These bins discretize normalised denoising progress and do not imply that generation uses 100 inference steps; each inference state is assigned to the bin closest to its corresponding τ_t value. This results in 3200 mean vectors for LLaDA and 2800 for Dream.

To verify whether the computed mean vectors capture the same signal identified in Section 3, we apply the same trained probes, without any additional tuning, to $\mu_{t,l}$ and compute the correlation between $\phi_l(\mu_{t,l})$ and the normalised bin index $t/100$. If both methods capture a similar latent direction, then the probe should behave monotonically as the mean vector's index increases. For LLaDA, the correlations reach 0.976 Pearson and 0.980 Spearman, while for Dream they reach 0.962 and 0.974, respectively. Moreover, as shown in Figure 3, the probe predictions align almost perfectly with the denoising bins encoded by each mean vector. These results suggest that the τ information identified by the probes is not merely recoverable at the token level, but also emerges as a robust global structure in the residual stream after averaging across examples. In other words, the τ signal appears to correspond to a coherent latent direction that is consistently represented across tokens and sequences, making it a natural target for the steering interventions we introduce next. Having verified that the mean vectors track the same signal recovered by the probes, we next assess its causal relevance by intervening on it during inference.

4.2 Steering

We use the computed activations to steer the model by swapping the mean vectors for the current step with those of the target $\hat{t}$ (where $\hat{t}$ is the discretised target bin whose denoising progress we steer the model towards):

$$\tilde{h}^j_{t,l} := h^j_{t,l} - \mu_{t,l} + \mu_{\hat{t},l}. \tag{4}$$

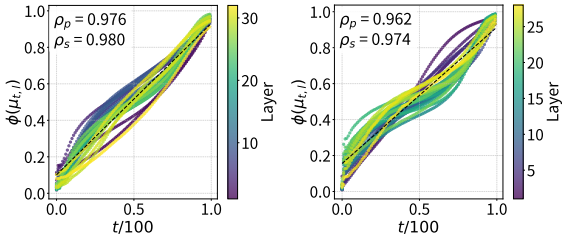


Figure 3: $\phi_l(\mu_{t,l})$ **and** $\mu_{t,l}$ **are highly correlated**. For each of the found $\mu_{t,l}$ we compute $\phi_l(\mu_{t,l})$ and plot it against the corresponding $\tau = t/100$. We observe high correlations for both LLaDA (left) and Dream (right).

We note that, since the perturbation is defined as a difference $\Delta_l^{t\to\hat{t}} := \mu_{\hat{t},l} - \mu_{t,l}$, the shared component $\bar{\mu}_l := \mathbb{E}_t[\mu_{t,l}]$ cancels out. This removes the part of the representation that is independent of denoising progress and intrinsic to the hidden states of layer l. Based on the results shown in Figure 2, we apply steering across all tokens to ensure that no τ-related information is inadvertently preserved when perturbing the hidden states. As a control condition, we construct the empirical covariance matrix $C_{t,l}$ (formed from the activations at step t and layer l), and we sample a random vector that is applied across all hidden states at that step:

$$\tilde{h}^j_{t,l} := h^j_{t,l} + a^{t\to\hat{t}}, \quad a \sim \mathcal{N}(\mathbf{0}, C_{t,l}) \tag{5}$$

We denote by $a^{t\to\hat{t}}$ the normalised version of the sampled vector a, rescaled such that $\left\|a^{t\to\hat{t}}\right\| = \left\|\mu_{\hat{t},l} - \mu_{t,l}\right\|$.

4.3 Experimental results

We now investigate whether steering the internal representation of τ produces consistent downstream changes in the model predictions. Intuitively, if the characterised signal truly relates to a model's mechanism for representing τ, then shifting this representation towards larger or smaller τ values should alter the model's confidence, entropy, and token distributions accordingly. To quantify these effects, we compare the clean token distributions

$$p^j_{t,n} := p_\theta(x^j_{0,n} \mid x_{t,n})$$

against the steered ones

$$\tilde{p}^j_{t,n} := p_\theta(x^j_{0,n} \mid \tilde{x}_{t,n}),$$

where $n \in [N]$ denotes the sequence index.

We evaluate three complementary quantities: the variation in entropy, the variation in confidence,

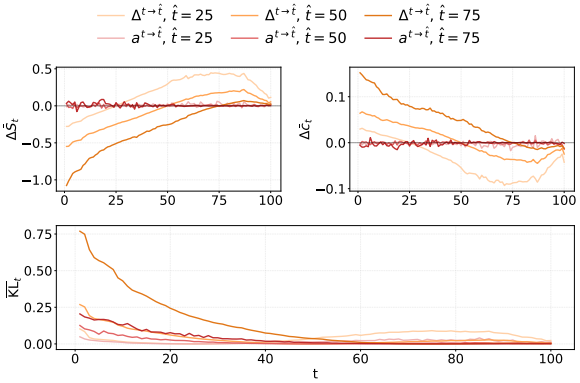


Figure 4: **Steering on layer 29 downstream effects**. We steer the activations of layer 29 using Equation (4) targeting different $\hat{t}$ values (blue), and compare the resulting entropy, confidence, and KL-divergence variations against norm-matched Equation (5) perturbations (red). Steering along the found τ directions produces systematic and interpretable effects.

and the KL-divergence between clean and steered distributions:

$$\begin{aligned} \Delta\bar{S}_t &:= \mathbb{E}_{n,j}\left[\mathrm{H}(\tilde{p}^j_{t,n}) - \mathrm{H}(p^j_{t,n})\right], \\ \Delta\bar{c}_t &:= \mathbb{E}_{n,j}\left[\max(\tilde{p}^j_{t,n}) - \max(p^j_{t,n})\right], \\ \overline{\mathrm{KL}}_t &:= \mathbb{E}_{n,j}\left[D_{\mathrm{KL}}\left(p^j_{t,n} \,\|\, \tilde{p}^j_{t,n}\right)\right]. \end{aligned} \tag{6}$$

We apply the steering vectors at different transformer layers and observe that the strongest downstream effects consistently emerge in the final layers of both LLaDA and Dream. Figure 4 presents the results obtained when steering layer 29 of LLaDA. Similar trends are observed in Dream when steering layer 25. In Appendix C, we show results across various layers for both LLaDA and Dream.

Remarkably, steering with Equation (4) produces a behaviour that closely matches what would be expected from directly modifying the model's internal notion of τ. When steering towards larger values of $\hat{t}$ relative to the current step t, i.e. $\hat{t} - t > 0$, the model becomes more confident and its entropy decreases. Conversely, when $\hat{t} - t < 0$, confidence decreases and entropy increases. At the same time, the KL divergence grows approximately proportionally to the distance $|\hat{t} - t|$, indicating progressively larger deviations from the original distribution as the steering target moves further away from the current denoising step.

To verify that these effects are specific to the identified signal and are not a phenomenon of the denoising mechanics of these models, we compare

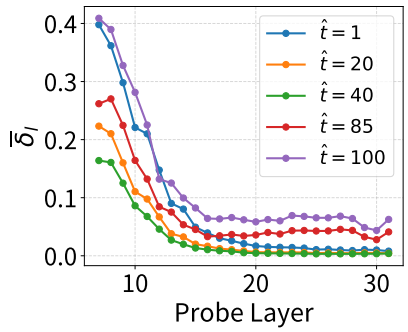
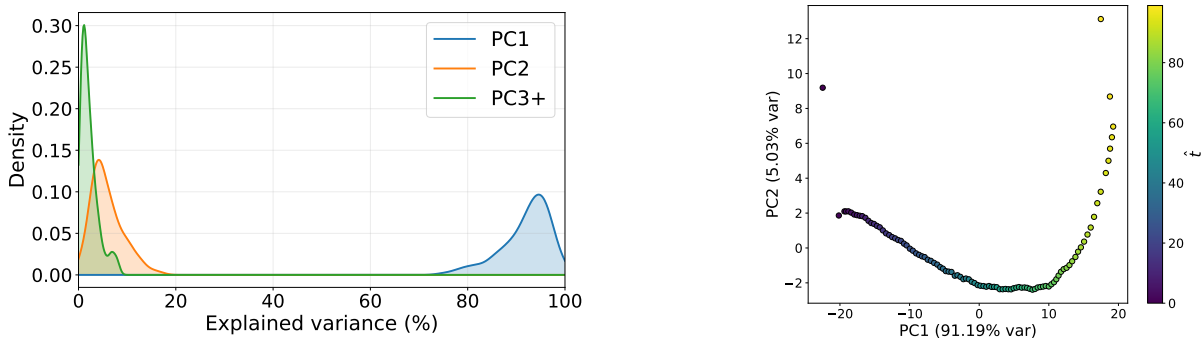


Figure 5: **Average probe drift across layers and steps after steering in LLaDA.** The model progressively compensates for the perturbation, reducing the discrepancy between clean and steered representations of τ.

them against Equation (5) with matched norm. Unlike the τ-based perturbations, random perturbations do not induce coherent trends in entropy or confidence across denoising steps. Moreover, their downstream impact is substantially weaker. For a comparable perturbation norm, the KL divergence induced by random perturbations is approximately half of that produced by τ steering, suggesting that the learned directions correspond to a particularly sensitive and semantically meaningful subspace in the residual stream. Contrary to the behaviour observed in the final layers, steering in shallow layers produces effects that are largely indistinguishable from the random baseline. Representative results for shallow- and mid-layer steering experiments, are reported in Appendix C. Combined with the probe analysis in Figure 2, this suggests that τ information is not merely propagated unchanged across layers: early perturbations can be attenuated by later computation as the model progressively compensates for them. We test this by injecting a steering perturbation at layer 6 and tracking the probe-predicted τ across the remaining layers.

What happens to the signal at early layers?

To explain the null result obtained when steering early layers, we investigate this phenomenon more deeply and measure how the probe predictions evolve after steering. If the model does indeed correct an early perturbation as computation proceeds, then the τ value read by the probes should drift back towards its clean value at deeper layers; conversely, a perturbation that survives unchanged would keep the steered and clean predictions far apart throughout. We therefore track the average deviation between the probe outputs of clean and



(a) The PCA distribution of the mean vectors.

(b) Dream's layer 25 2D projection

Figure 6: **PCA distribution and sampled projection on the top-2 principal components of the mean vectors.** Left: PCA distribution of the mean vectors for LLaDA; notably, across layers, most of the variance of the identified set of mean vectors can be explained by fewer than three dimensions. Right: sampled projections for Dream. We use the mean vectors obtained from layer 25 of Dream. We observe a parabola-like geometry.

steered activations across layers

$$\bar{\delta}_l := \mathbb{E}_t \left[\left| \mathbb{E}_j \left[\phi_l(\tilde{h}^j_{t,l}) - \phi_l(h^j_{t,l}) \right] \right| \right].$$

Starting from a perturbation injected at layer 6, we evaluate how different steering targets $\hat{t}$ propagate through the networks' depth.

As shown in Figure 5, the model is able to correct most of the perturbation introduced in early layers, such that the probe predictions under the perturbation become nearly indistinguishable from the clean ones in the final blocks.

However, steering toward extreme values such as $\hat{t} = 100$ remains harder to compensate for, leaving a persistent discrepancy even in deeper layers. This observation also helps explain the sharp peaks previously observed for shallow-layer steering (see Appendix C): at those denoising steps, the model is temporarily unable to fully recover from the injected perturbation before producing its prediction (additional analysis is provided in Appendix E). Having established that this signal affects the models' output distributions in a predictable way, we now move on to characterising its properties and look for the emergence of model-level patterns in how the signal is organised.

✓ Takeaway for RQ2

The found signal has direct causal implications for modelling dynamics, predictably affecting downstream entropy, confidence and KL divergence. The models also internally recompute it several times across their depth, allowing for correction.

5 Characterising the Signal

Having established how the models react to changes in the signal, we now examine how this signal is represented internally and whether consistent structural patterns emerge.

5.1 μ vectors' low-dimensional structure

We begin by analysing the structural properties of the discovered mean vectors, focusing on the subspace spanned by these representations. Specifically, we examine its effective dimensionality. The resulting distribution of explained variance across principal components is shown in Figure 6a. Surprisingly, a large fraction of the intra-layer variance can be captured by a single principal component, with the distribution heavily concentrated above 90% explained variance. Motivated by this observation, we project the mean vectors from selected layers into two- and three-dimensional PCA spaces to better visualise their structure. We show a sample 2D and 3D projection in Figure 6b and Figure 1, respectively. Interestingly, we find the 3D shape to be closely related to structures that models have been shown to exhibit when operating on counting or time-dependent and sequentially ordered tasks (Gurnee et al., 2025; Engels et al., 2025; Karkada et al., 2026; Modell et al., 2025), characterised by a continuous and ordered representation of the time-axis being analysed (in our case τ).

To test whether the functional effect of τ steering is concentrated in this low-dimensional subspace, we repeat the intervention of Equation (4), but restrict the steering vector $\Delta_l^{t\to\hat{t}} := \mu_{\hat{t},l} - \mu_{t,l}$ to a chosen set of principal directions. Concretely, we isolate the part of $\Delta_l^{t\to\hat{t}}$ lying in the subspace

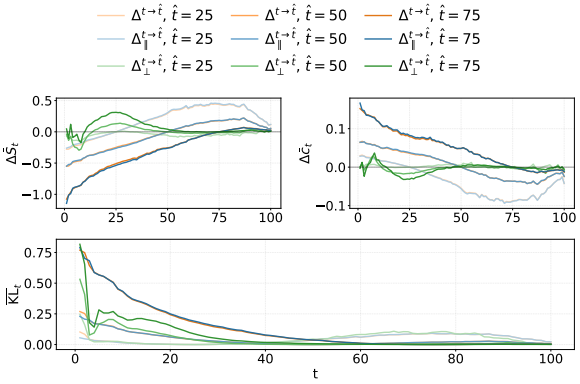


Figure 7: **Low-dimensional subspace steering in LLaDA.** Using Equation (7), we steer the model within the two-dimensional subspace ($k = 2$) spanned by the top principal components of the layer-29 mean vectors. Steering within the subspace closely resembles the unrestricted one, while the orthogonal perturbation produces incoherent effects.

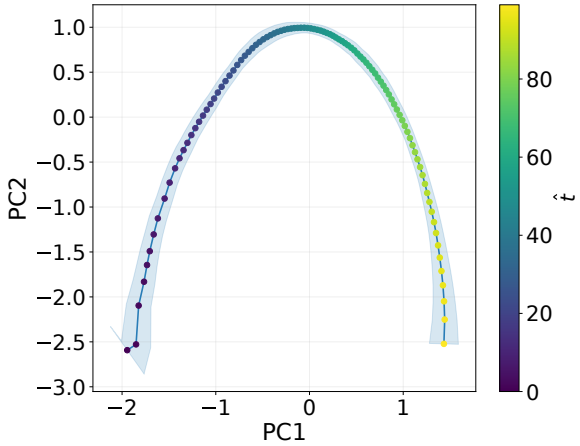


Figure 8: **The shared 2-dimensional geometry of $\hat{t}$ in LLaDA.** We compute an average trajectory by taking the standardised 2D PC projections of the mean vectors across the model. The parabolic geometry is shared across layers. PC1 and PC2 are standardised to unit variance.

spanned by the first k principal components and the part orthogonal to it, rescaling each to match the norm of the original steering vector:

$$\Delta^{t\to\hat{t}}_{\parallel,l} := P_{\parallel,l}\,\Delta^{t\to\hat{t}}_{l}\,\frac{\|\Delta^{t\to\hat{t}}_{l}\|}{\|P_{\parallel,l}\,\Delta^{t\to\hat{t}}_{l}\|}$$
$$\Delta^{t\to\hat{t}}_{\perp,l} := P_{\perp,l}\,\Delta^{t\to\hat{t}}_{l}\,\frac{\|\Delta^{t\to\hat{t}}_{l}\|}{\|P_{\perp,l}\,\Delta^{t\to\hat{t}}_{l}\|} \tag{7}$$

Here $P_{\parallel,l}$ denotes the orthogonal projection onto the subspace spanned by the first k principal components of layer l, and $P_{\perp,l} = I - P_{\parallel,l}$ the projection onto its complement. Because the top components capture most of the variance (Figure 6a), we have $\|P_{\parallel,l}\,\Delta^{t\to\hat{t}}_{l}\| \gg \|P_{\perp,l}\,\Delta^{t\to\hat{t}}_{l}\|$; rescaling both perturbations to $\|\Delta^{t\to\hat{t}}_{l}\|$ therefore isolates the effect of the steering *direction* from that of its *magnitude*. As shown in Figure 7, projecting onto just the first two principal components recovers the same behaviour as the full, unprojected perturbation. Interestingly, the orthogonal perturbation $\Delta^{t\to\hat{t}}_{\perp,l}$ produces incoherent effects at the same norm, implying that those directions carry little information of the τ representation. We present more results on Appendix D. These results motivate a closer examination of whether this geometry can be consistently described across layers and aggregated into a unified, model-level representation.

5.2 A shared representation across layers

From the observation that many layers exhibit a similar parabolic shape, we investigate whether a general model-level 2D representation of the mean vector components exists. In particular, we aim to model a 2D trajectory $f(t) : [100] \to \mathbb{R}^2$ that describes the geometry of the top two principal components across the entire model.

To this end, we collect the 2D projections from all layers and standardise them to have zero mean and unit variance. We then compute, for each $t \in [100]$ (corresponding to the 100 mean vectors projected per layer), the empirical mean of the 2D points associated with the mean vectors binned by t. This yields a mean-standardised projection across layers. From this procedure, we obtain a non-parametric 2D trajectory that represents the standardised expectation of the projections for each τ index. We report the resulting plot in Figure 8 for LLaDA (see Appendix F for similar results on Dream). Remarkably, all points closely follow the proposed shared trajectory, with error bars remaining within approximately 0.1, indicating that the proposed approximation faithfully captures the general model-level geometry of the subspace. This further implies that the principal components of the mean vector distributions follow the same parabolic shape up to scaling and offset factors.

While this construction reveals a stable and coherent 2D structure shared across layers, it remains unclear whether the underlying representation is consistently encoded throughout the depth of the model or whether it emerges from layer-specific implementations. We next address this question by

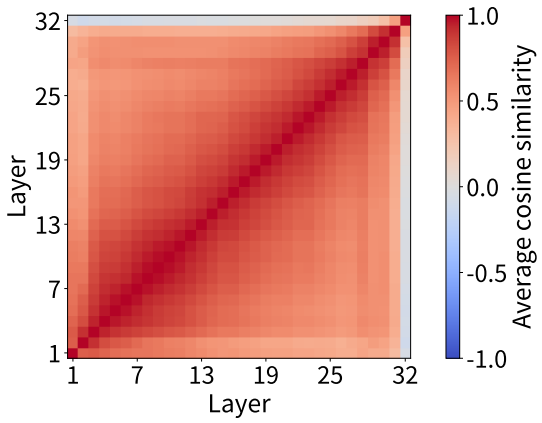


Figure 9: **Average cosine similarity of same-indexed vectors across layers in LLaDA.** We observe that most layers maintain a highly correlated representation of the same index t, and that this relationship degrades with distance. Layer 32 instead maintains a representation that is largely independent of the other layers.

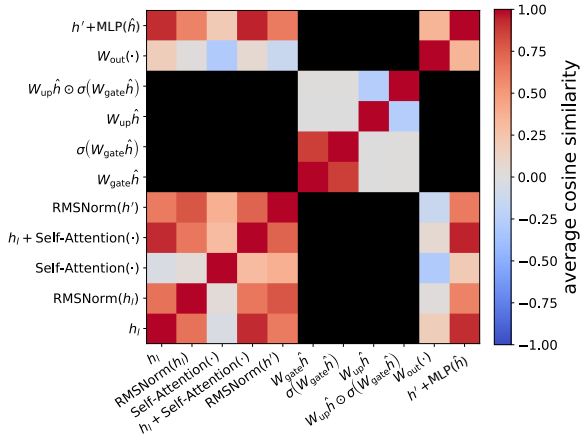


Figure 10: **Average cosine similarity across layer components in LLaDA.** Where $h' := h_l + \text{Self-Attention}(\cdot)$, and $\hat{h} := \text{RMSNorm}(h')$. We clearly see how across most of the internal representations there is high-correlation or almost complete orthogonality. We note how self-attention and the outputs of the out matrix seem to be anticorrelated.

studying the cross-layer alignment of the discovered mean vectors.

5.3 Cross-layer τ representations

Based on these results, we further investigate how the discovered mean vectors are connected across layers. In particular, we ask whether the model maintains a consistent direction-wise representation of a given μ_t vector across depth, or instead develops layer-local representations that do not generalise. To address this question, we compute the average pairwise cosine similarity between all centred vectors encoding a given index t across different layers. We report the results in Figure 9 for LLaDA and in Appendix F for Dream.

Notably, we observe a clear structural difference between the two models in how they preserve representations of the same mean vector across depth. In LLaDA, most layers exhibit strong alignment in their representations, with the exception of layer 32, which appears nearly orthogonal to the others. In contrast, Dream shows a more heterogeneous organisation: representations are only consistently aligned within specific blocks of layers, while layers outside these blocks are largely uncorrelated with those inside them. This suggests that there may be a model-level basis for encoding τ across layers, as having high correlation implies that the downstream effects would be similar if interchanging the mean vectors from those layers. Although the previous sections establish the existence of structured and consistent τ representations across layers, they do not reveal how these representations are formed within the computations of each layer, which we investigate in the next section.

How is τ represented within a layer?

To query the models' internal representation of the mean vectors we employ the same approach as proposed in Equation (3), but now capture activations inside the models' intermediate layer computations. With these mean vectors, we measure the cosine similarity of all same-dimensional representations inside the models' layers (as the MLP has a different hidden dimension compared to the rest of the model). We present the results for LLaDA at Figure 10 (and Appendix F for Dream). As the results suggest, most operations before and after the MLP remain highly correlated; nonetheless, inside the MLP, the mean vectors produce almost completely independent representations of τ, as for both models they are orthogonal in expectation (with the exception of the post up-projection matrix and post activation function, which remain highly correlated). We note a particular pattern in both models between the post-MLP and post-attention similarities, as both are highly anti-correlated, sharing a common direction but pointing in opposite ways. This phenomenon can be better appreciated in Figure 11. Interestingly, the two stages agree on the distance from the centre of a τ value but not on its sign. We hypothesise that this seeming contradiction in the mean τ representation may help the

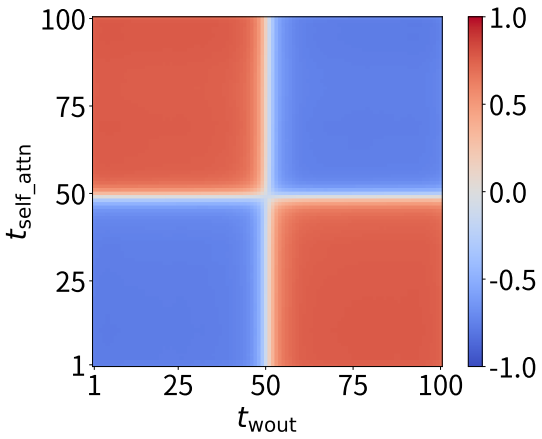


Figure 11: **Average cosine similarity of self-attention and post-W_{out} vectors.** Layer-29 vectors from LLaDA reveal a clear geometric organization over t: representations for distant indices become nearly anti-aligned, and a transition region emerges around $t = 50$, where vectors are approximately perpendicular.

model correct the signal as shown in Section 4.3.

✓ Takeaway for RQ3

The models represent the mean vectors subspace in an ordered and manifold-like manner. Most layers in LLaDA share common semantics for this space, while Dream is organised into representation blocks. Finally, self-attention and the MLP have opposite representations.

6 Related Work

LLM Interpretability. Prior work studies activation subspaces and representation geometry to explain LLM behaviour. Closest to us, Gurnee et al. (2025) use mean activation vectors to analyse and steer a character counting task. We similarly use mean vectors, but study a distributed denoising-time signal and its propagation across layers. Engels et al. (2025), Modell et al. (2025), and Karkada et al. (2026) analyse concept and token-level representations in language models, describing low-dimensional manifold geometries that, in the case of ordered concepts such as historical years, resemble the one we find for τ. In contrast to their work, we study a sequence-level statistic that the model reconstructs as "meta-level" information to track denoising progress, rather than a human-like concept (such as years or months) whose geometry has been tied to statistical co-occurrences.

DLM Denoising Dynamics and Interpretability. Recent DLM work has examined both steering and internal behaviour. Zhou et al. (2026) find that concepts are steerable only at particular denoising stages; in contrast, our τ-related signal persists throughout diffusion and is repeatedly recomputed. Rossi et al. (2026) study DLM length-awareness through downstream performance, whereas we characterise an internal representation tied to denoising progress.

7 Conclusion and Discussion

We characterise a denoising-progress-related signal in the residual streams of DLMs. The signal is probe-decodable across layers and token types, causally steerable through mean vector directions, repeatedly corrected across depth, and represented in a structured low-dimensional subspace. Steering this representation produces predictable changes in entropy, confidence, and KL divergence, indicating that the recovered directions are not merely descriptive but functionally relevant to the model's computation.

A key implication of our findings is that probes or auxiliary models operating on DLMs hidden states need not be explicitly conditioned on a timestep embedding to access denoising-progress information. As the probes show, this information is already intrinsically encoded in the residual stream and can therefore be recovered directly from the activations on which such methods operate.

A natural next question is how this signal is computed. Our results show that DLMs internally track information closely related to the fraction of unmasked tokens, but they do not identify the circuit that constructs or updates this representation. Future work could therefore investigate whether the signal is derived from explicit mask-ratio estimation, from distributed sequence-level statistics, or from interactions between attention and MLP components. Understanding this computation would better clarify how DLMs operate in this subspace and enable better architecture designs for these models.

8 Limitations

Although our work aims to characterise and explain the representation of τ inside DLMs, several open questions remain. First, our analysis is restricted to LLaDA and Dream, two DLMs trained with the same cross-entropy loss over [MASK] tokens; it therefore remains unexplored how the τ

phenomenon emerges in other DLMs that similarly lack explicit timestep conditioning, such as block-diffusion models (Cheng et al., 2025; Arriola et al., 2025; Wu et al., 2025). Moreover, it is unclear whether the identified τ signal can be exploited at inference time to enable more efficient decoding—or even remasking—strategies, which we leave as a promising direction for performance improvements. Another promising direction is studying small DLMs trained from scratch; as it could help determine which properties of the τ representation emerge early in training or at small scale, and which depend on model depth or capacity. Finally, the token-level effects of steering remain unexplored: our analyses focus on sequence-level statistics rather than examining which tokens change, and how, as the signal is steered.

9 Acknowledgements

This work was supported by a grant from Coefficient Giving, administered by the Berkeley Existential Risk Initiative (BERI), and by Sapienza grant RG123188B3EF6A80 (CENTS). We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking and hosted by CINECA (Italy). We thank Fastweb S.p.A. for providing the computational resources used in this paper. We also thank Donatella Genovese and Francesco Piccolo for helpful discussions and valuable feedback.

We used LLMs only for minor grammar, wording, and readability checks. We reviewed and edited the final manuscript and take full responsibility for its content.

10 Ethical considerations

This work analyses the internal representations of publicly released diffusion language models (LLaDA and Dream) and does not involve human subjects, new data collection, or the release of new models. We foresee no ethical concerns arising from this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2025. System card: Claude Opus 4 & Claude Sonnet 4. https://www.anthropic.com/claude-4-system-card.

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, volume 2025, pages 50726–50753.

Usha Bhalla, Thomas Fel, Can Rager, Sheridan Feucht, Tal Haklay, Daniel Wurgaft, Siddharth Boppana, Matthew Kowal, Vasudev Shyam, Jack Merullo, Atticus Geiger, and Ekdeep Singh Lubana. 2026. Do sparse autoencoders capture concept manifolds? *Preprint*, arXiv:2604.28119.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and 1 others. 2025. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *Preprint*, arXiv:2412.19437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Zhenbang Du, Kejing Xia, Xinrui Zhong, Yonggan Fu, Nicolai Oswald, Binfei Ji, Brucek Khailany, Pavlo Molchanov, and Yingyan Lin. 2026. R^2-dllm: Accelerating diffusion large language models via spatio-temporal redundancy reduction. *arXiv preprint arXiv:2604.18995*.

Josh Engels, Eric Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. 2025. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations*, volume 2025, pages 84591–84622.

Gemma Team. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson. 2025. When models manipulate manifolds: The geometry of a counting task. *Transformer Circuits Thread*.

Pengcheng Huang, Shuhao Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Zulong Chen, and Tong Xiao. 2025. Pc-sampler: Position-aware calibration of decoding bias in masked diffusion models. *arXiv preprint arXiv:2508.13021*.

Abhinav Joshi, Divyanshu Bhatt, and Ashutosh Modi. 2025. Geometry of decision making in language models. *Preprint*, arXiv:2511.20315.

Dhruva Karkada, Daniel J Korchinski, Andres Nava, Matthieu Wyart, and Yasaman Bahri. 2026. Symmetry in language statistics shapes the geometry of model representations. *arXiv preprint arXiv:2602.15029*.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

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

Meta AI. 2025. Introducing Llama 4: Advancing multimodal intelligence. `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. 2025. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *Preprint*, arXiv:2502.09992.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2024. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Julian Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Julianna Piskorz, Cristina Pinneri, Alvaro Correia, Motasem Alfarra, Risheek Garrepalli, and Christos Louizos. 2025. Masks can be distracting: On context comprehension in diffusion language models. *Preprint*, arXiv:2511.21338.

Vittorio Rossi, Giacomo Cirò, Davide Beltrame, Luca Gandolfi, Paul Röttger, and Dirk Hovy. 2026. Diffusion language models are natively length-aware. *arXiv preprint arXiv:2603.06123*.

Maximo Eduardo Rulli, Simone Petruzzi, Edoardo Michielon, Fabrizio Silvestri, Simone Scardapane, and Alessio Devoto. 2025. Attention sinks in diffusion language models. *arXiv preprint arXiv:2510.15731*.

Baturay Saglam, Paul Kassianik, Blaine Nelson, Sajana Weerawardhena, Yaron Singer, and Amin Karbasi. 2026. Large language models encode semantics and alignment in linearly separable representations. *Preprint*, arXiv:2507.09709.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *Preprint*, arXiv:2406.07524.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. 2025. Transformers represent belief state geometry in their residual stream. *Preprint*, arXiv:2405.15943.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. 2025. Simplified and generalized masked diffusion for discrete data. *Preprint*, arXiv:2406.04329.

Federico Tiblias, Irina Bigoulaeva, Jingcheng Niu, Simone Balloccu, and Iryna Gurevych. 2026. Hypothesis-driven feature manifold analysis in llms via supervised multi-dimensional scaling. *Preprint*, arXiv:2510.01025.

Weixuan Wang, Jingyuan Yang, and Wei Peng. 2025. Semantics-adaptive activation intervention for llms via dynamic steering vectors. *arXiv preprint arXiv:2410.12299*.

Xu Wang, Bingqing Jiang, Yu Wan, Baosong Yang, Lingpeng Kong, and Difan Zou. 2026. Dlm-scope: Mechanistic interpretability of diffusion language models via sparse autoencoders. *Preprint*, arXiv:2602.05859.

Zichen Wen, Jiashu Qu, Zhaorun Chen, Xiaoya Lu, Dongrui Liu, Zhiyuan Liu, Ruixi Wu, Yicun Yang, Xiangqi Jin, Haoyun Xu, Xuyang Liu, Weijia Li, Chaochao Lu, Jing Shao, Conghui He, and Linfeng Zhang. 2026. The devil behind the mask: An emergent safety vulnerability of diffusion llms. *Preprint*, arXiv:2507.11097.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. 2025. Fast-dllm v2: Efficient block-diffusion llm. *Preprint*, arXiv:2509.26328.

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

Yifan Yu, Yuqing Jian, Junxiong Wang, Zhongzhu Zhou, Donglin Zhuang, Xinyu Fang, Sri Yanamandra, Xiaoxia Wu, Qingyang Wu, Shuaiwen Leon Song, and 1 others. 2026. Introspective diffusion language models. *arXiv preprint arXiv:2604.11035*.

Oussama Zekri, Théo Uscidda, Nicolas Boullé, and Anna Korba. 2026. Generalized discrete diffusion from snapshots. *arXiv preprint arXiv:2603.21342*.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. 2025. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *Preprint*, arXiv:2409.02908.

Hanhan Zhou, Shamik Roy, and Rashmi Gangadharaiah. 2026. Steering without breaking: Mechanistically informed interventions for discrete diffusion language models. *arXiv preprint arXiv:2605.10971*.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2026. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11425–11460.

A Denoising Progress and the Training Noising Level

Here we provide more details about DLMs' training objectives and clarify the relationship between the training noising level s and the empirical denoising-progress variable τ_t used throughout the paper. For more details about these models' formulations we refer the reader to Nie et al. (2025) and Ye et al. (2025). In absorbing-mask diffusion language models, the fraction of response tokens that remain masked is an empirical analogue of the training noising level, while the fraction of response tokens that have been unmasked measures denoising progress.

Absorbing-mask corruption. Let x_0^j denote the clean token at position j, represented as a one-hot vector. We write [MASK] both for the absorbing mask token and, when used inside a categorical parameter, for its corresponding one-hot vector. At noising level $s \in [0, 1]$, each token is independently replaced by [MASK] with probability s. For a single position j,

$$q_{s|0}^j(z \mid x_0^j) = \begin{cases} 1 - s, & z = x_0^j, \\ s, & z = [\text{MASK}], \\ 0, & \text{otherwise.} \end{cases}$$

Equivalently,

$$q_{s|0}^j(\cdot \mid x_0^j) = \text{Cat}\big(\cdot; (1 - s)x_0^j + s[\text{MASK}]\big).$$

Assuming independent corruption across positions, the full corrupted sequence distribution factorises as

$$q_{s|0}(\cdot \mid x_0) = \prod_j q_{s|0}^j(\cdot \mid x_0^j).$$

Thus, s is the probability that a token is masked, and $1 - s$ is the probability that it remains visible.

Training objectives. For a corrupted sequence x_s, define the masked-position set

$$\mathcal{M}(x_s) := \{j : x_s^j = [\text{MASK}]\}.$$

The pretraining objective can then be written compactly as

$$\mathcal{L}_{\text{pre}} = -\mathbb{E}_{x_0, s, x_s} \left[\frac{1}{s} \sum_{j \in \mathcal{M}(x_s)} \log p_\theta(x_0^j \mid x_s) \right]. \tag{8}$$

During supervised fine-tuning, the prompt is kept fixed and only response tokens are corrupted. Let p_0 denote the prompt, r_0 the clean response, and r_s the corrupted response. Define

$$\mathcal{M}(r_s) := \{j : r_s^j = [\text{MASK}]\}.$$

The SFT objective is

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{p_0, r_0, s, r_s} \left[\frac{1}{s} \sum_{j \in \mathcal{M}(r_s)} \log p_\theta(r_0^j \mid p_0, r_s) \right]. \tag{9}$$

Both objectives therefore train the model on partially masked sequences, where the expected fraction of masked tokens is controlled by s.

Inference-time denoising progress. At inference time, the model does not sample a corrupted sequence directly from $q_{s|0}$. Instead, it starts from a fully masked response and progressively unmasks tokens over discrete denoising steps $t \in \{0, \ldots, T\}$.

Let R be the set of response positions, with $L := |R|$. For a realised inference state x_t, define the response-local masked and unmasked sets as

$$\begin{aligned} \mathcal{M}_R(x_t) &:= \{j \in R : x_t^j = [\text{MASK}]\}, \\ \mathcal{U}_R(x_t) &:= \{j \in R : x_t^j \neq [\text{MASK}]\}. \end{aligned} \tag{10}$$

These sets partition the response positions, so

$$|\mathcal{M}_R(x_t)| + |\mathcal{U}_R(x_t)| = L.$$

We define the empirical denoising progress at step t as

$$\tau_t := \frac{|\mathcal{U}_R(x_t)|}{L}. \tag{11}$$

Its complement is the remaining response-local mask ratio:

$$1 - \tau_t = \frac{|\mathcal{M}_R(x_t)|}{L}. \tag{12}$$

Thus, $\tau_t = 0$ corresponds to a fully masked response, while $\tau_t = 1$ corresponds to a fully unmasked response.

Connection to the training noising level. Under the SFT corruption process, each response token is masked independently with probability s. Therefore, for $r_s \sim q_{s|0}(\cdot \mid r_0)$,

$$\mathbb{E}_{r_s}\left[\frac{|\mathcal{M}(r_s)|}{L}\right] = s. \tag{13}$$

Similarly, if $\mathcal{U}(r_s) := \{j : r_s^j \neq [\text{MASK}]\}$, then

$$\mathbb{E}_{r_s}\left[\frac{|\mathcal{U}(r_s)|}{L}\right] = 1 - s. \tag{14}$$

Consequently, the realised inference-time mask ratio $(1 - \tau_t)$ plays the role of an effective noising level, while τ_t measures effective denoising progress.

Interpretation. The equivalence above should be understood at the level of mask ratios, not as an equality between the training and inference distributions. During training, s indexes an independently sampled corruption process. During inference, the sequence of states is generated by the model and its unmasking policy. Nevertheless, both procedures produce partially masked responses, and each such response has a well-defined response-local mask ratio.

We therefore use τ_t as an empirical measure of denoising progress throughout the paper. Its complement, $1 - \tau_t$, is the realised fraction of response tokens that remain masked and is directly analogous to the training noising level s. Recovering τ_t from the residual stream can thus be interpreted as recovering the model's internal representation of its position along the denoising trajectory.

B Probe Architectural and Training Details

We use a 5-layer residual neural network with LayerNorm+GELU blocks, shown in Figure 12, and bound the outputs to the interval $(0, 1)$ by applying a sigmoid function to the MLP logits, since the target τ is itself bounded. We train the probes for 20 epochs using 300 training examples and 100 validation examples. For each example and at each epoch, we dynamically vary the generation length and number of steps to improve the probe's coverage. We implement batch gradient descent by updating the parameters after each full example, i.e., by batching the gradients over the entire denoising stage. We employ the

```python
def block(d_in, d_out):
    return nn.Sequential(
        nn.Linear(d_in, d_out),
        nn.LayerNorm(d_out),
        nn.GELU(),
    )

class MLPProbe(nn.Module):
    def __init__(self, dim, bounded=True):
        super().__init__()
        w = min(dim, 1024)
        self.inp = block(dim, w)
        self.middle_in = block(w, w)
        self.expansion = nn.Sequential(
            block(w, 2 * w),
            block(2 * w, w),
        )
        self.middle_out = block(w, w)
        self.out = nn.Linear(w, 1)
        self.bounded = bounded

    def forward(self, x):
        h = self.middle_in(self.inp(x))
        e = self.expansion(h)
        h = self.middle_out(e) + h   # residual
        out = self.out(h)
        if self.bounded:
            return out.sigmoid()
        return out
```

Figure 12: **MLP probe architecture.** Each `block` is a $\mathrm{Linear} \to \mathrm{LayerNorm} \to \mathrm{GELU}$ unit; the probe stacks five such blocks around a single residual connection, and a sigmoid head bounds the output to $(0, 1)$ to match the range of τ. The hidden width w is capped at 1024.

AdamW (Loshchilov and Hutter, 2017) weight optimiser, setting the learning rate to $\alpha = 10^{-3}$ and the weight decay coefficient to 6×10^{-6}. All experiments introduced earlier were conducted on NVIDIA H100 and A100 GPUs.

C Additional Experiments on Probing and Steering τ

In this section, we present additional results concerning the probes' capabilities to correctly identify the value of τ, followed by further analyses of the mean activation vectors' ability to steer the model. The first part extends the probing analysis to Dream and to linear probes; the second broadens the steering analysis to additional layers, generation lengths, and downstream tasks.

MLP probes on Dream. Figure 13 shows the R^2 obtained using the MLP probe over Dream. Similarly to what was observed with LLaDA, the probes' accuracy is high across the whole model but drops in the deeper layers.

Linear probes. We also train linear probes with the same sigmoid output and regression objective as the MLPs (Figure 12), reporting the results in Figure 15. Both models' linear probes track τ well through the early and middle layers, but their accuracy then falls off more steeply than the MLP probes as we move deeper, and in the final layers it collapses essentially to the baseline of predicting the mean τ. Across the broad mid-to-deep range the MLP probes retain markedly higher R^2 (Figure 2, Figure 13), so for those layers we read the gap not as the τ signal disappearing at the token level, but as it becoming less linearly accessible: the information is still present, yet increasingly encoded in a form a single linear readout cannot capture. Only at the very last layers, where the MLP performance also degrades, does the token-level signal become genuinely hard to recover by either probe. The choice of training tokens also matters more here than for the MLPs: for LLaDA, restricting training to non-[MASK] tokens opens a

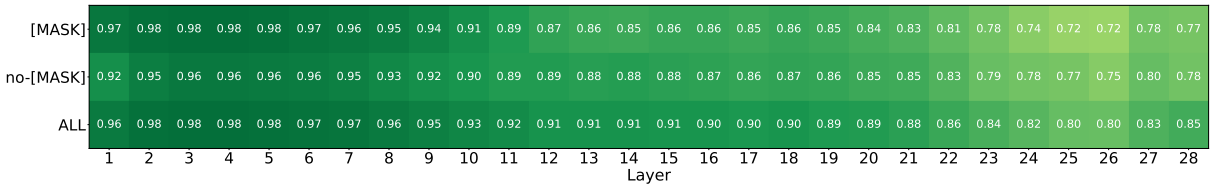


Figure 13: **R^2 coefficients of MLPs in Dream**. The R^2 coefficient slightly degrades as we probe deeper into the model, which remains consistent with the observations of Figure 2.

noticeable gap relative to [MASK] or all tokens—consistent in direction with the smaller [MASK] advantage seen for the MLP probes, but more pronounced—whereas Dream is mostly insensitive to the token subset.

We stress that this does not contradict the low-dimensionality of τ established earlier: that structure is a property of the mean vectors, which average over many tokens and examples and thereby expose sequence-level regularities to which an individual token's activation—and hence a token-level linear probe—has no comparable access.

Mean-vector steering on Dream. Having shown that τ is decodable across both models, we now turn from reading the signal to perturbing it, extending the steering analysis beyond the layers reported in the main text. We begin with Dream: Figure 14 shows the downstream effects of steering at layer 25. We observe a similar trend as in LLaDA, where entropy decreases and confidence increases when we steer towards later denoising steps.

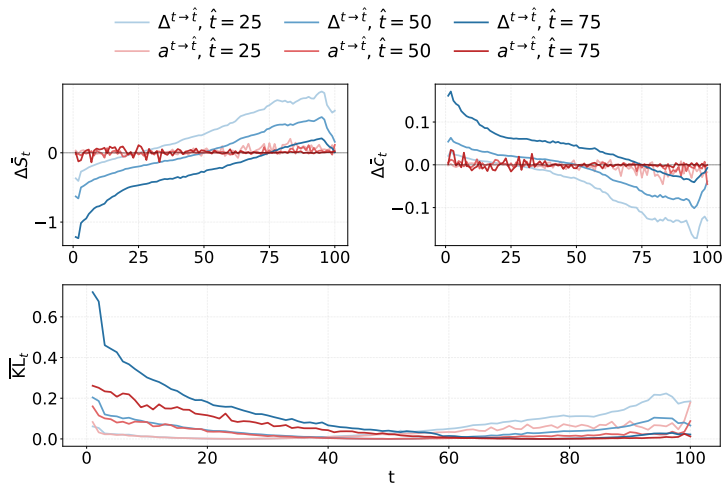


Figure 14: **Mean-steering downstream effects on layer 25 of Dream**. We steered the activations in layer 25 using the mean-ratio vectors (blue) targeting different τ values, and measured the variation in entropy, confidence and the KL divergence. We compared it against random perturbations (red). Steering with the mean vectors has an effect that is consistent with what would be expected.

Steering across intervention layers. Figure 20 and Figure 21 show the downstream results for LLaDA and Dream across a range of intervention layers and target bins $\hat{t}$.

Robustness to generation length. In Figure 19 we ablate the effect of generation length, repeating the mean-steering analysis at $L = 64$ and $L = 128$ (with the number of denoising steps matched to L). The downstream effects follow the same trajectory as in the main results across all steering targets $\hat{t}$, indicating that the previously-observed phenomena are not artifacts of a particular generation length.

Downstream task performance. In Table 1 we show the effect of mean vector steering on downstream task performance. We observe that steering does not lead to a collapse in performance across the range of $\hat{t}$ values, with LLaDA remaining within a few points of its base score on all three benchmarks. Dream is more sensitive, particularly on GSM8K, where performance degrades substantially at $\hat{t} = 100$. A

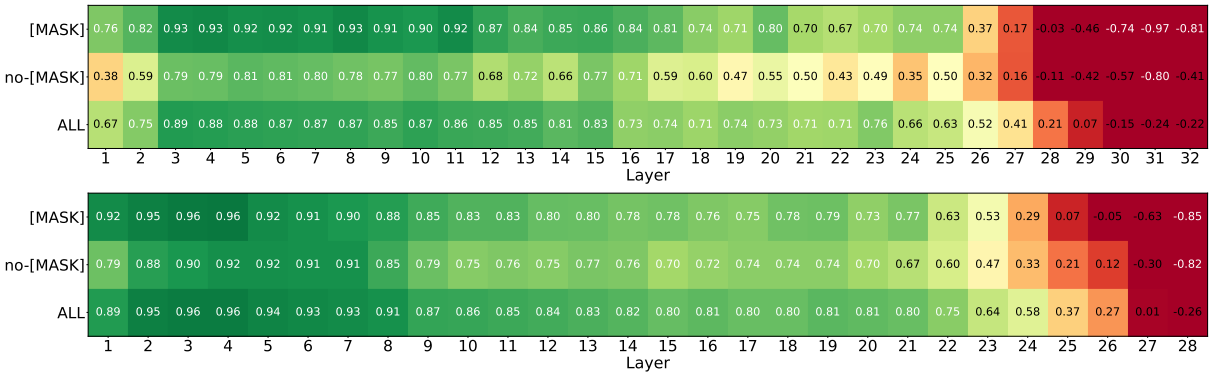


Figure 15: **Linear probe performance**. The linear probes degrade markedly as we probe deeper into the model—more so than the MLP probes shown in Figure 2 and Figure 13—and fall to the mean-τ baseline in the final layers for both LLaDA (top) and Dream (bottom). This indicates that, at the token level, τ information becomes increasingly hard to decode linearly with depth, while an MLP still recovers much of it until the last few layers.

Table 1: Effect of mean-steering on LLaDA (layer 29) and Dream (layer 25) on downstream performance.

			$\hat{t}$				
Model	Dataset	base	1	25	50	75	100
LLaDA-1.5	GSM8K	84.3	81.6	81.0	83.1	84.1	81.0
	HumanEval	44.5	41.6	47.4	46.0	45.3	46.7
	StrategyQA	69.5	63.1	69.5	66.6	65.3	69.2
Dream-7B	GSM8K	68.8	51.6	61.5	59.5	56.9	23.1
	HumanEval	35.0	28.0	32.9	35.0	29.4	23.8
	StrategyQA	68.3	67.2	66.3	67.2	68.5	68.3

qualitative inspection of Dream's incorrect GSM8K generations reveals that this degradation is largely driven by the premature emission of end-of-sequence (EOS) tokens. Steering towards $\hat{t} = 100$ leads the model to unmask EOS tokens both early in the denoising trajectory and at early positions within the response window. Since these models are trained to keep predicting EOS once the first one has been emitted, a single EOS placed near the start of the response causes the generation to collapse into a degenerate, EOS-filled sequence. This failure mode is consistent with the decoding biases reported by Huang et al. (2025), who show that uncertainty-based samplers in DLMs over-select trivial, high-confidence tokens—including EOS, newlines, and punctuation—during the first denoising steps, and tend to commit the two ends of the sequence before its centre (a "U-shaped" decoding trajectory). Our intervention can be read as amplifying this bias from the representation side: by driving the model's internal estimate of denoising progress to its maximum, we further inflate the probability of completion tokens such as EOS, which in turn triggers the collapse described above. Taken together, and while we leave a quantitative analysis to future work, this suggests that the τ signal is not a content-agnostic progress counter, but is instead entangled with the token distribution itself—carrying latent information about which tokens become more or less probable as denoising advances, with sequence-ending tokens such as EOS being a clear example.

D Low-dimensional Steering

In this section we provide further insights on the causal importance of the found low-dimensional subspaces when steering LLaDA and Dream. To do so, we employ the subspace steering method introduced in Equation (7). As Figure 22 and Figure 23 suggest, with just the 1D projection of the mean vectors, we can recover a steering downstream impact similar to the one we attain when performing the steering with the

untouched mean vectors (Equation (4)). Furthermore, as we increase the subspaces' dimensions from one to two, the results resemble with high precision the phenomena induced by the untouched mean vectors, implying that the 2D parabolas described at Figure 8, do encode the majority of the τ-relevant geometry. Finally, we observe how the effects caused by $\Delta_{l,\perp}^{t\to\hat{t}}$ remain of low-impact and do not follow any clear trend, implying that those dimensions have little relevance to modelling τ.

E Depth Correction

Here we provide a finer-grained view of the depth-correction phenomenon discussed in Section 4.3. Figure 16 shows the per-step unrolled version of Figure 5 for different $\hat{t}$: instead of averaging the probe drift $\bar{\delta}_l$ over denoising steps, we plot one column per step, so that averaging the columns recovers the Figure 5 curve. Early denoising steps are markedly more sensitive to the injected perturbation: the absolute probe drift remains substantial across nearly the entire depth of the network, propagating up to layer 31, while at later steps it is suppressed within the first few layers above the injection site. This is consistent with the sharp peaks observed for shallow-layer steering (see Appendix C), and indicates that the model's ability to correct an injected $\hat{t}$ depends strongly on the denoising step at which the intervention occurs. Moreover, the residual drift decreases monotonically as the steered step becomes more compatible with $\hat{t} = 100$, i.e. as t approaches the end of the denoising trajectory.

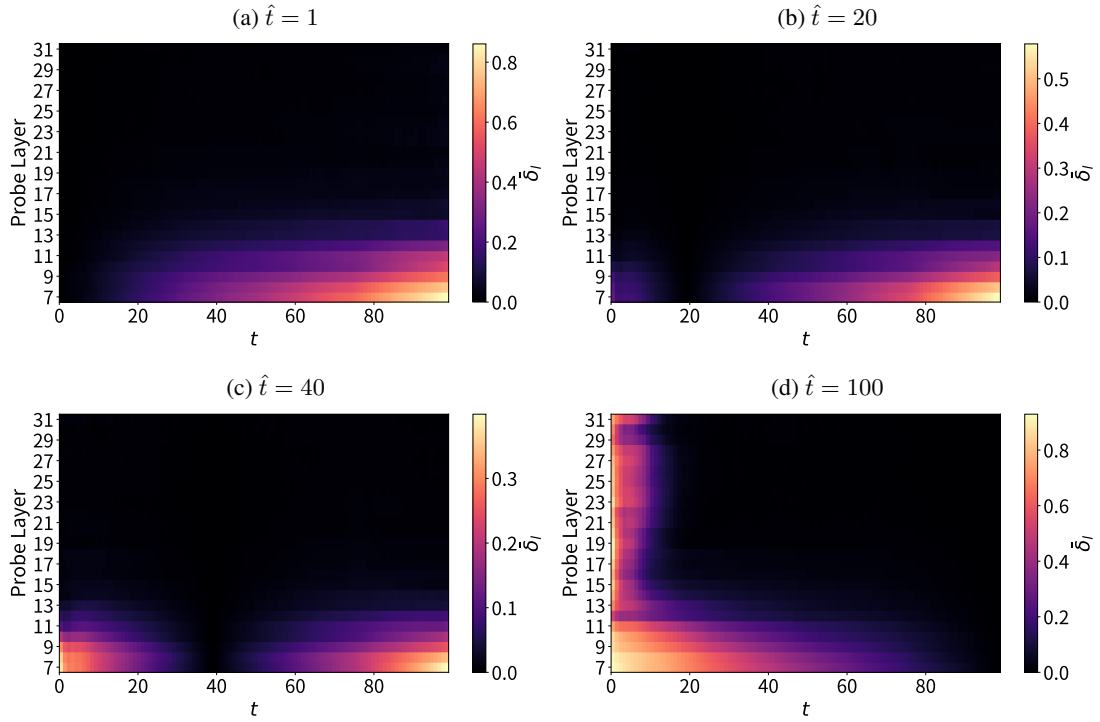
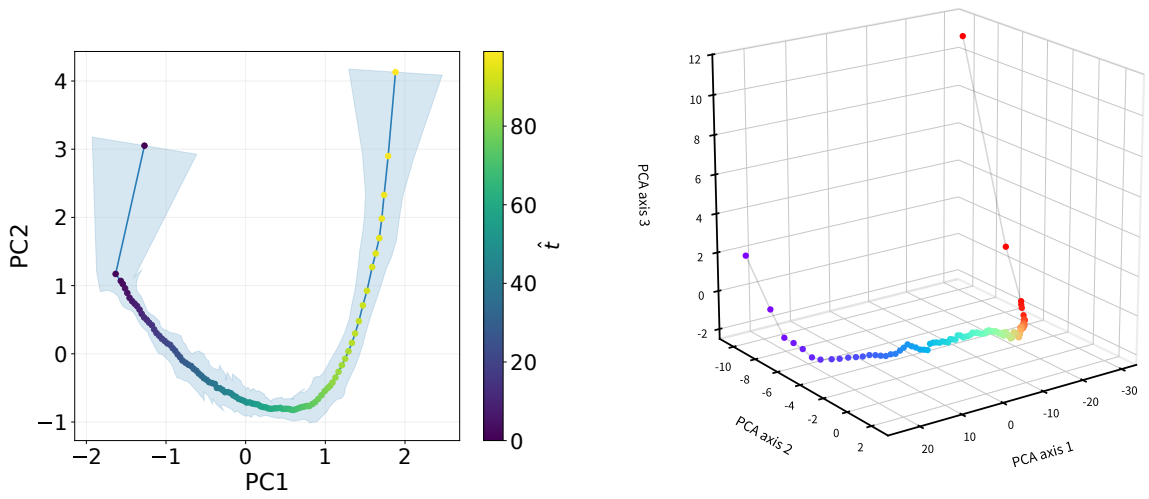


Figure 16: **Probe drift across layers and denoising steps when steering at layer 6.** Each column shows the absolute drift $\bar{\delta}_l$ between clean and steered probe predictions at a given denoising step t, resolved across probe layers. We can observe how only for $\hat{t} = 100$ the perturbation at some steps penetrates all across the model and achieves a distortion at the last layers.

F Dream's Mean Activation Analysis

In this section, we present additional results regarding Dream's mean activations, mirroring our LLaDA analysis from the main text. Similarly to the behaviour observed in LLaDA, we find that the representations associated with τ concentrate in a structured low-dimensional subspace, exhibiting a smooth and ordered trajectory across denoising time.

Shared 2D geometry. Figure 17a shows the shared 2D geometry obtained by averaging the standardised PCA projections across layers for Dream. As in LLaDA, the resulting geometry follows a clear parabola-



(a) **The shared 2-dimensional geometry of $\hat{t}$ in Dream.** We compute an average trajectory by taking the standardised 2D PC projections of the mean vectors across the model. The parabolic geometry is shared across layers, similarly to LLaDA. PC1 and PC2 are standardised to unit variance.

(b) **3D visualisation of the PCA projection of the mean vectors for layer 25 of Dream.** As in LLaDA, we again observe a low-dimensional ordered structure. Notably, two points, corresponding to $t = 1$ and $t = 100$, lie far from the rest of the geometry, suggesting a special representation.

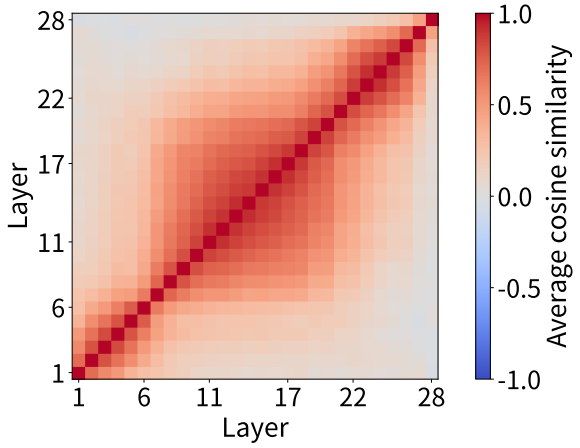
Figure 17: **Low-dimensional geometry of τ's mean-vectors in Dream.** Left: shared 2D mean-vector trajectory across layers. Right: 3D PCA projection for layer 25.

like trajectory, supporting the idea of a shared geometry of τ across the whole model. Nonetheless, differently from LLaDA, Dream's parabola has a bigger spread at the two endpoints $t = 1$ and $t = 100$, implying that, at the boundaries, the geometrical organisation significantly changes across layers.

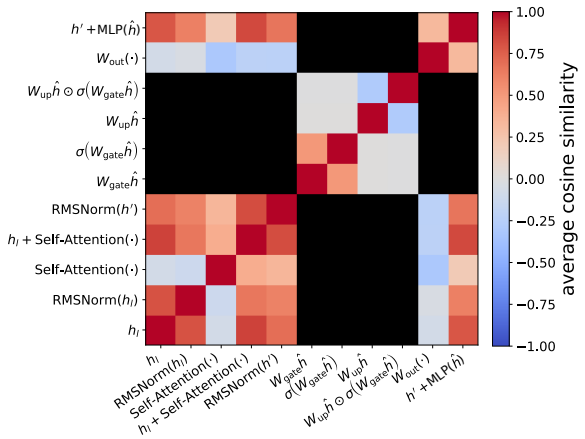
3D geometry. Figure 17b presents a 3D PCA projection of the mean vectors from layer 25 of Dream. Again, the resulting trajectory reveals a low-dimensional, ordered structure. Notably, the representations associated with the extreme denoising steps, namely the initial and final steps, appear as outliers, suggesting that the model may behave qualitatively differently at these extremes, which helps explain the big boundary deviations of Figure 17a.

Cross-layer alignment. Mirroring our LLaDA analysis in Figure 9, we investigate how the τ representations align across layers for Dream. In contrast to LLaDA, which exhibits strong global alignment across most layers, Figure 18a shows that Dream presents a substantially more heterogeneous organisation. Interestingly, the model appears to partition its layers into three main computational stages, with high cosine similarity within each block and weaker alignment across blocks.

Within-layer components. We present the cosine similarity across layer components for Dream (as for LLaDA in Figure 10). In Figure 18b we can appreciate how Dream follows an internal geometric organisation of the mean vectors similar to LLaDA.

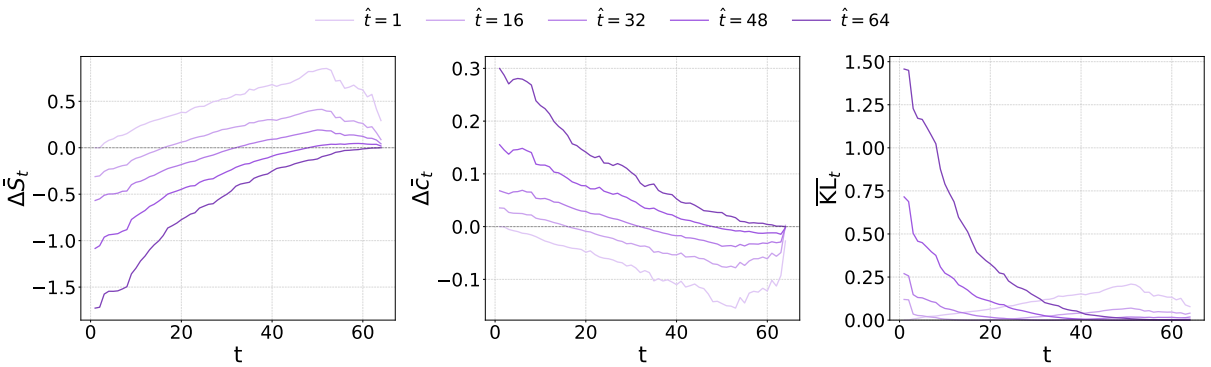


(a) **Average mean vector cosine similarity for Dream.** Dream presents a more orthogonal relationship than LLaDA, indicating that layers are more independent in their representation of τ. We also observe three structured high-similarity blocks aligned with the model's computation stages: layers 1–6, 7–20, and 21–27. Note again the last layer's orthogonality.

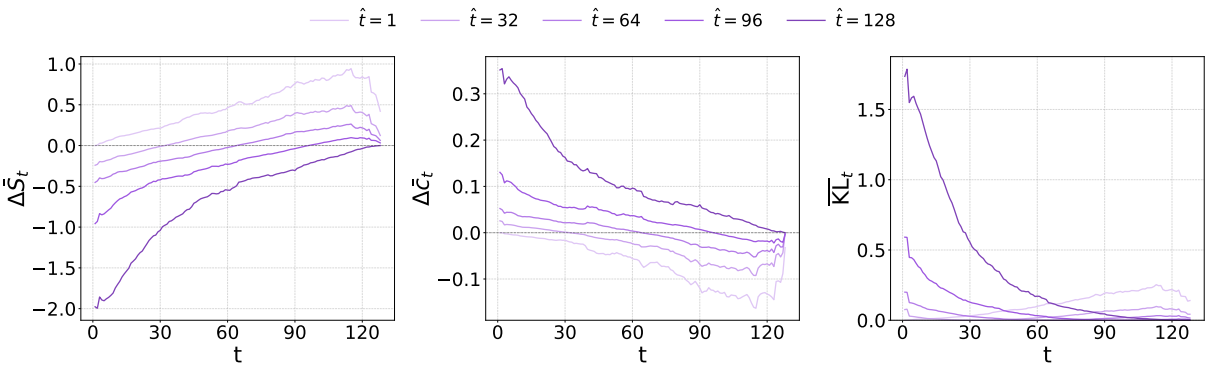


(b) **Average cosine similarity across layer components in Dream.** The pre-MLP components remain highly correlated, while the MLP representations split into two groups: the gate path, and the up-projection together with the Hadamard product. As in LLaDA, self-attention and MLP vectors are anticorrelated.

Figure 18: **Cosine-similarity structure of mean vectors in Dream.** Left: average similarity across layers. Right: average similarity across intermediate layer components.



(a) $L = 64, \quad \hat{t} \in \{1, 16, 32, 48, 64\}$.



(b) $L = 128, \quad \hat{t} \in \{1, 32, 64, 96, 128\}$.

Figure 19: **Mean-steering downstream effects across generation lengths for LLaDA**. Ablation over generation length $L \in \{64, 128\}$, with the number of denoising steps matched to L. Each row shows, left to right, $\Delta\bar{S}_t$, $\Delta\bar{c}_t$ and $\overline{\mathrm{KL}}_t$ versus the denoising step t, for five steering targets $\hat{t}$. Vertical axes use a symmetric-log scale to resolve the convergence near 0.

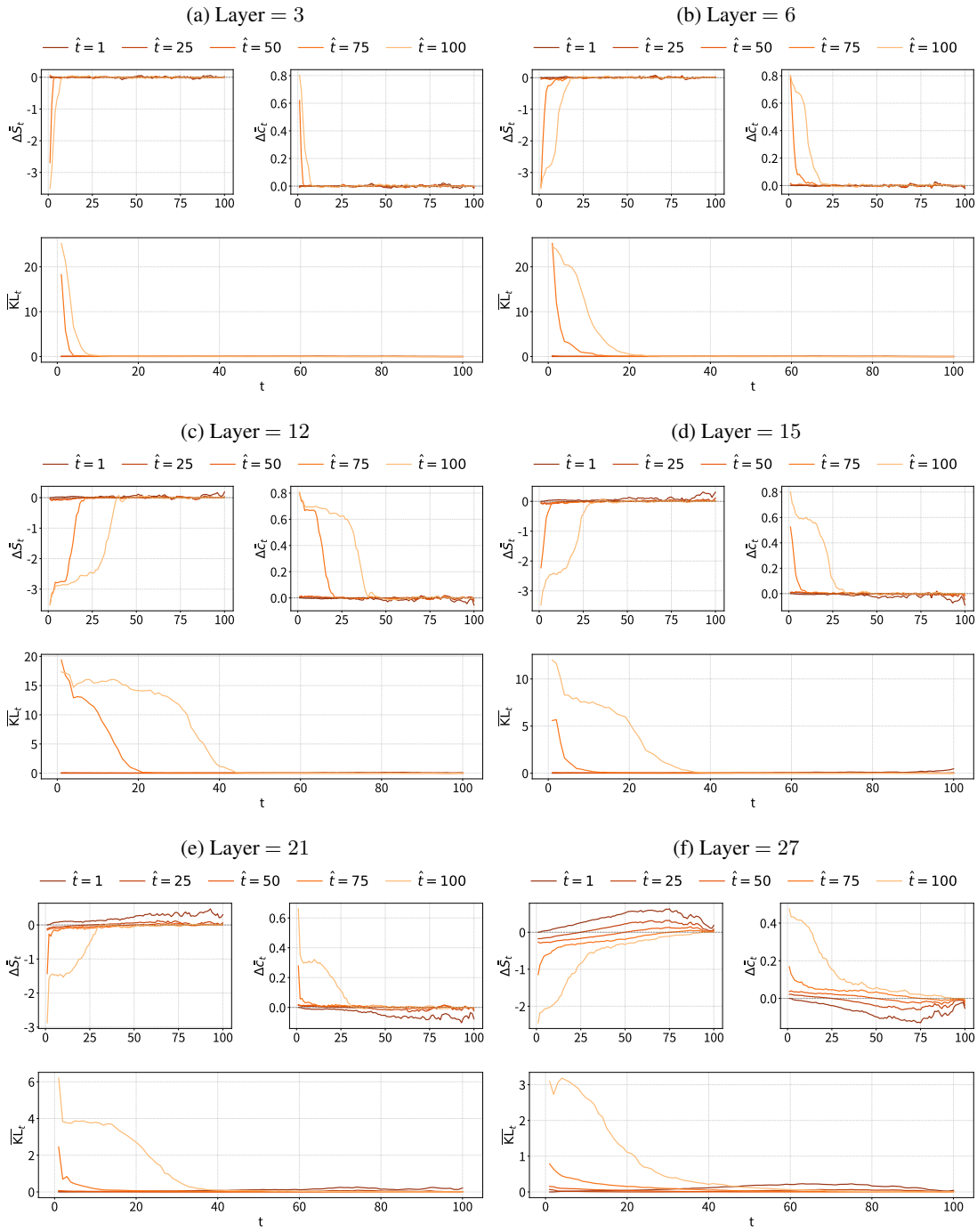


Figure 20: **Layer-wise mean-steering effects in LLaDA.** We apply mean vector steering at different intervention layers and target denoising-progress bins $\hat{t} \in \{1, 25, 50, 75, 100\}$. Each panel reports the downstream effect on entropy drift, confidence drift and KL divergence across denoising steps.

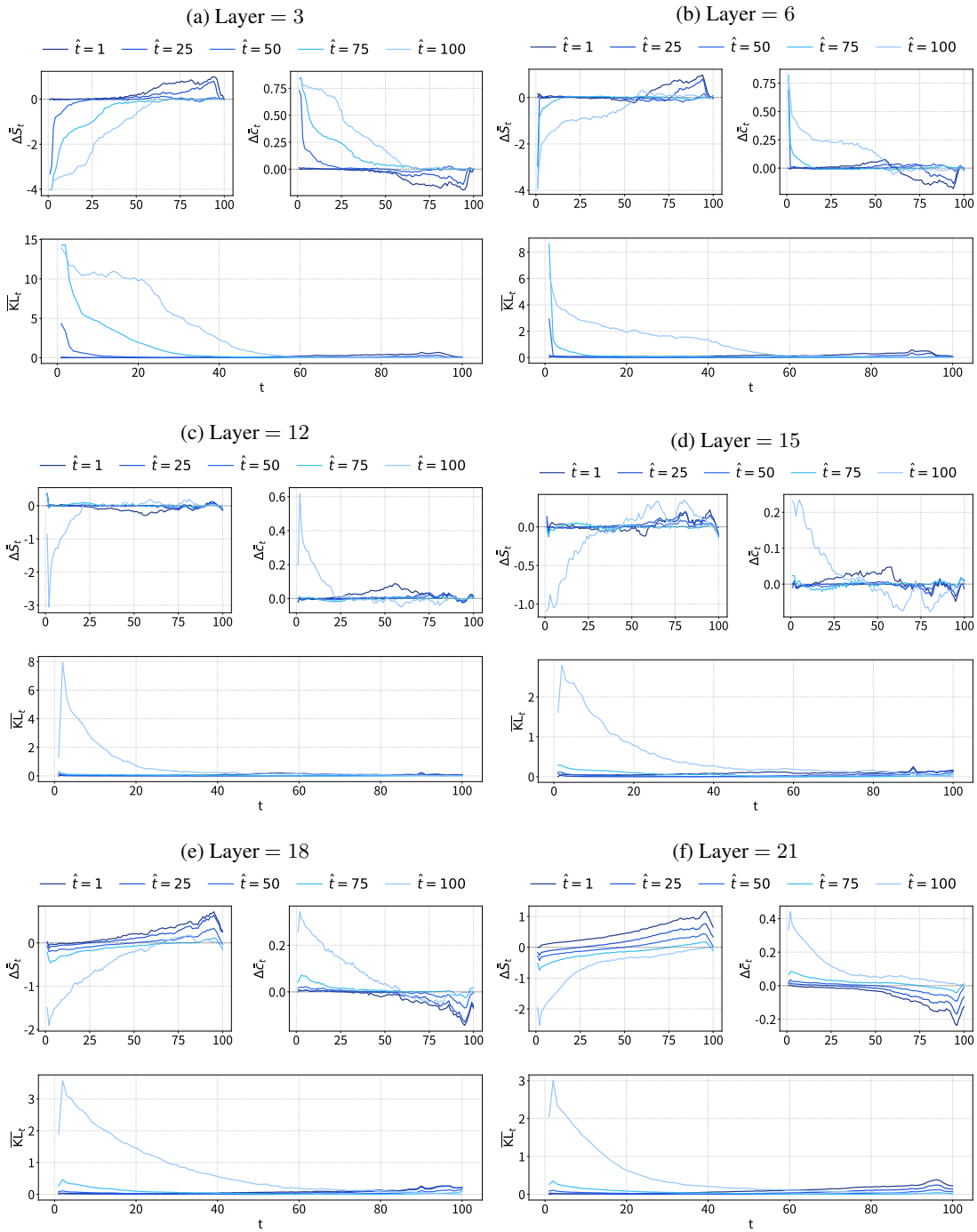


Figure 21: **Layer-wise mean-steering effects in Dream.** We apply mean vector steering at different intervention layers and target denoising-progress bins $\hat{t} \in \{1, 25, 50, 75, 100\}$. Each panel reports the downstream effect on entropy drift, confidence drift and KL divergence across denoising steps.

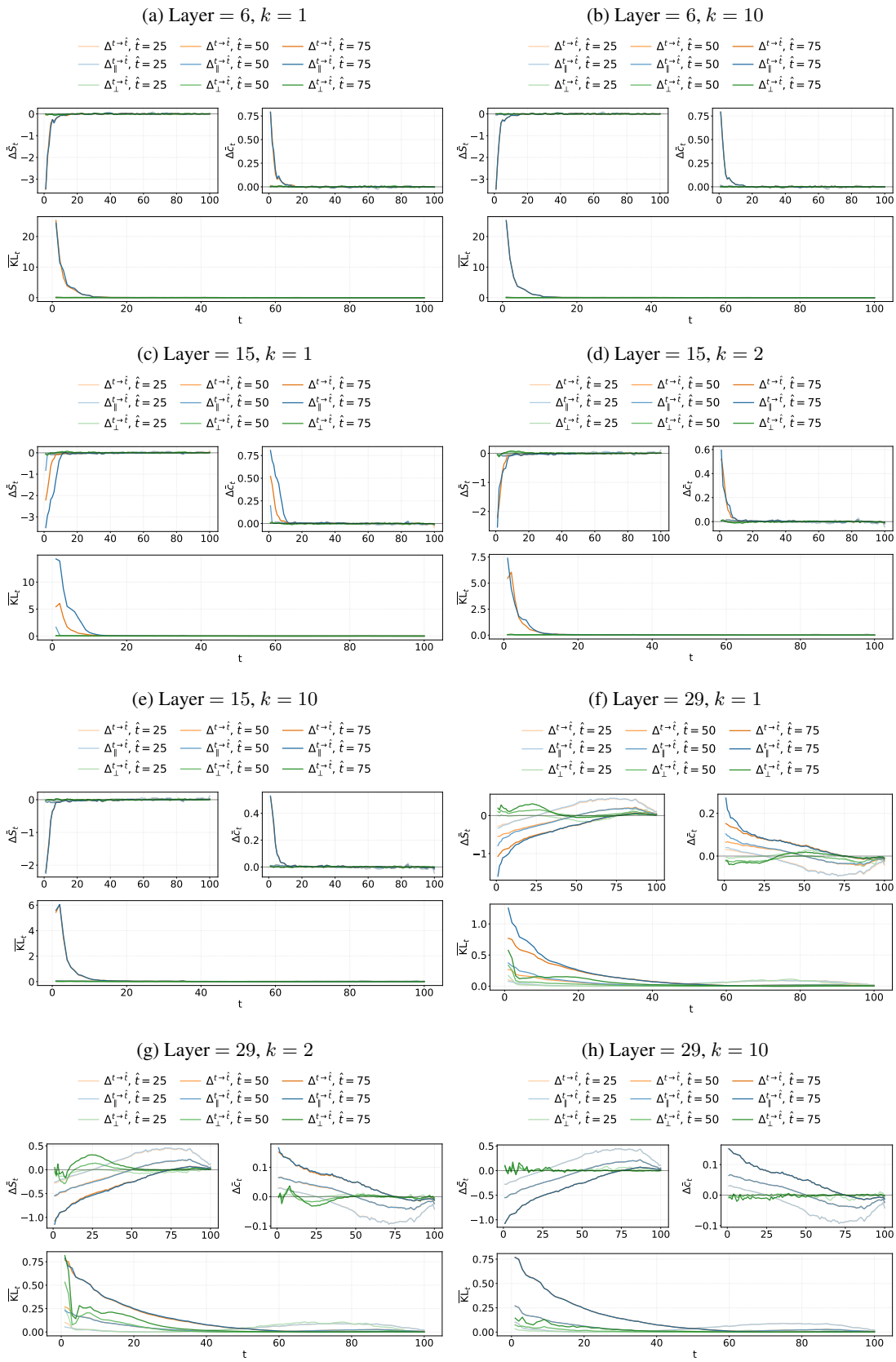


Figure 22: **Low-dimensional steering on LLaDA.** LLaDA's mean vectors concentrate around a low-dimensional subspace. Steering across the top-1, top-2, and top-10 principal components yields results similar to using the unprojected mean vector.

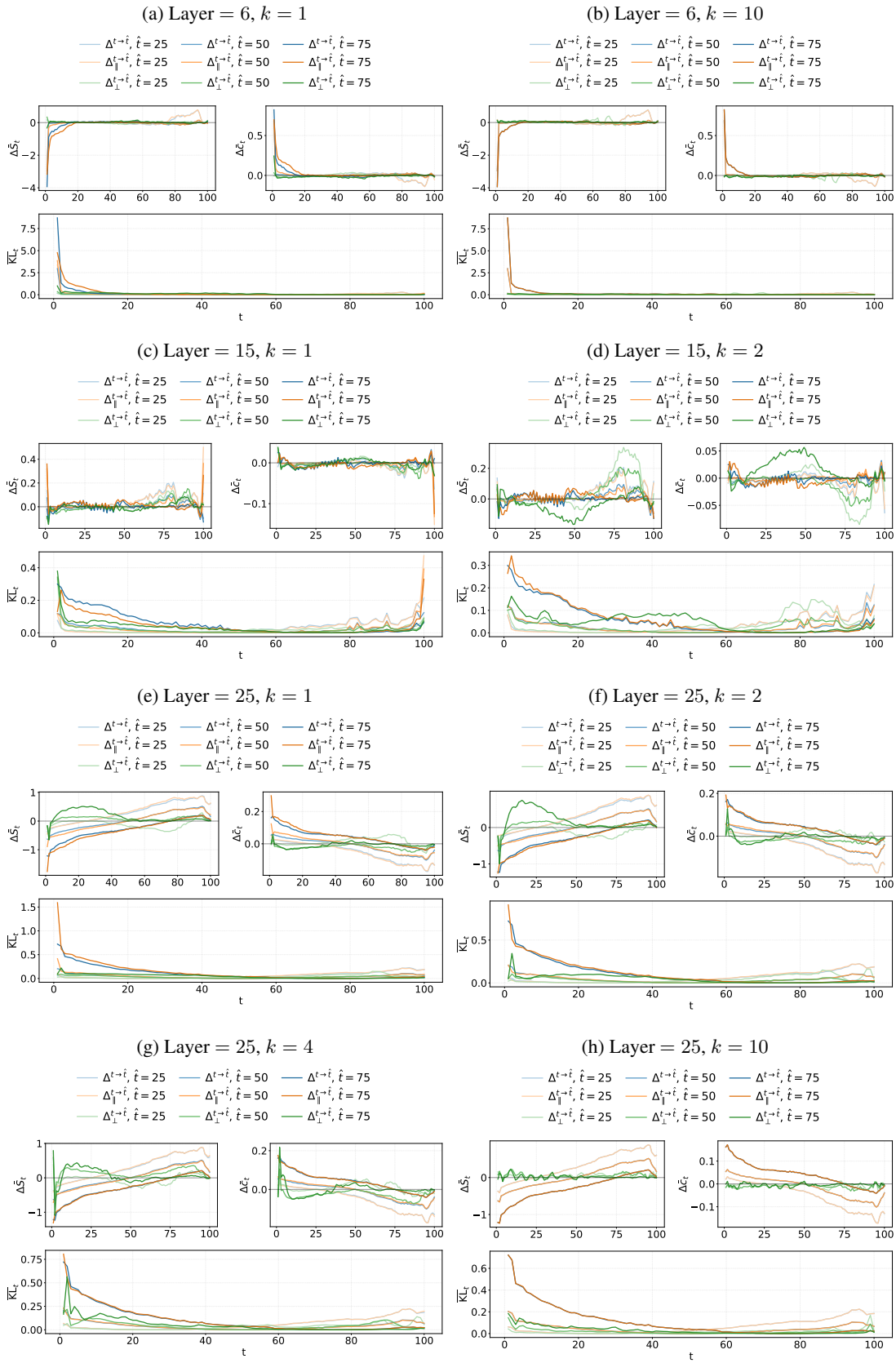


Figure 23: **Low-dimensional steering on Dream.** Dream's mean vectors concentrate around a low-dimensional subspace as in LLaDA. Similar to Figure 22, steering across the top-1, top-2, top-4, and top-10 principal components yields results similar to using the unprojected mean vector.