Subliminal Clocks:
Latent Time Modelling in Diffusion Language Models

Anonymous ACL submission

Abstract

Masked Diffusion Language Models (MDLMs) have recently emerged as a promising alternative to autoregressive models. Unlike standard diffusion-based approaches, MDLMs 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 MDLMs 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 controlled and 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 manipulated by these models.

1 Introduction

Large Language Models (LLMs) (Anthropic, 2025; MetaAI, 2024; Achiam et al., 2023; DeepSeek-AI, 2024; Team, 2025) are driving a paradigm shift across various scientific and social domains. Understanding the capabilities of these models is still an active area. 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, a new way of generating text has emerged with large-scale Diffusion Language Models (DLMs) (Nie et al., 2025; Ye et al., 2025). These models operate in a BERT-style (Devlin et al., 2019) way by progressively unmasking tokens as generation progresses. Despite the growing interest in 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, 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

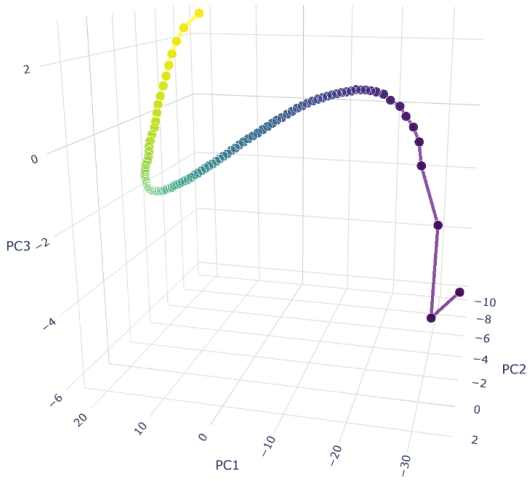


Figure 1: **3D projection of latent denoising step modelling for LLaDA**. LLaDA represents its τ subspace as a low-dimensional manifold-like curve, progressively modelling the denoising progress from all [MASK] (purple) to no remaining [MASK] (yellow).

itself remain poorly understood. In particular, it is still unclear whether DLMs internally represent denoising time during generation. In absorbing-state DLMs (Shi et al., 2025; Ou et al., 2024; Sahoo et al., 2024), denoising time is closely tied to the fraction of masked tokens in the input sequence—a signal that is directly available to the model, yet has not been mechanistically characterised. Understanding whether and how models encode this information is important for explaining how they organise computation across denoising steps.

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.

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). A 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 finite 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 (e.g. low-confidence for Nie et al. (2025)). This policy determines how many tokens are revealed per step. At inference time we can measure the fraction of already unmasked tokens as

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

In Appendix A, we show that τ_t is equivalent to the complement of the continuous diffusion-time variable s. 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

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

where n is the example index, j is the token index in the example sequence, and $h_{t,l,n}^j$ denotes

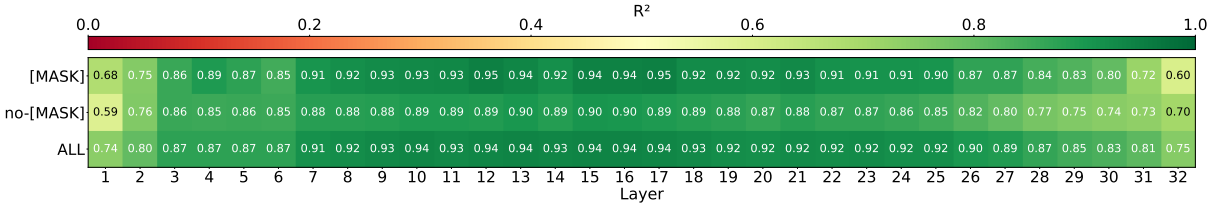


Figure 2: **MLP probes recover the τ signal**. The R^2 coefficient (higher is better) degrades as we probe deeper into the model, to the point where the MSE almost reduces to half the variance of τ. Moreover, both [MASK] and non-[MASK] tokens seem to carry this information, attaining similarly high coefficients.

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, although the gap remains relatively small. This suggests that τ information is not exclusively localised to masked positions, but is instead distributed across the residual stream.

Importantly, each MLP probe operates on a single hidden state independently, without access to any sequence-level context or neighbouring activations, hinting that individual token representations themselves encode substantial information about τ. Thus, the high performance of the probes indicates that the tokens **carry a latent representation** of a sequence-level statistic related to the τ signal we are probing for. 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

Having established that DLMs encode information about the denoising process in their residual stream, we now investigate whether this signal can be explicitly extracted and manipulated. 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 τ statistic.

Following Wang et al. (2025), we compute mean activation vectors over all hidden states in the response window, grouping them according to their denoising step. More formally, for each denoising step 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. This results in 3200 mean vectors for LLaDA and 2800 for Dream.

To verify whether the computed mean vectors capture the same signal identified by the probes from Section 3, we evaluate the correlation between the probe predictions $\phi_l(\mu_{t,l})$ and the normalised

timestep 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, but that can also be recovered at a token-level with a probe. Having verified that the mean vectors track the same signal recovered by the probes, we next test whether this structure is causally relevant 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 $\hat{\tau}$ (where $\hat{\tau}$ represents the discretised target τ to which we want to steer the model):

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

We note that, since the perturbation is defined as a difference $\Delta_l^{t\to\hat{\tau}}$, the shared component $\bar{\mu}_l$ 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 additionally introduce a perturbation obtained by randomly sampling directions from the empirical covariance matrix. Specifically, 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:

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

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

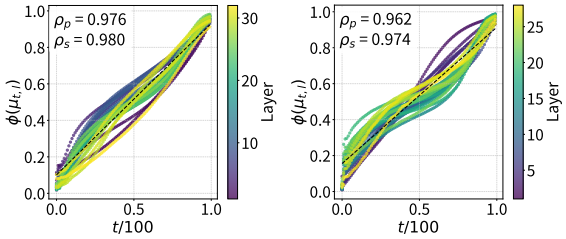


Figure 3: ϕ_l **and** $\mu_{t,l}$ **are highly correlated**. For each of the found $\mu_{t,l}$ we compute $\phi_l(\mu_{t,l})$ and plot it against the corresponding $\tau = t/100$. We observe high correlations for both LLaDA (left) and Dream (right).

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 presumably 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 $t \in [T]$ denotes the denoising step, $n \in [N]$ the sequence index, and $j \in [L]$ the token position.

We evaluate three complementary quantities: the variation in entropy, the variation in confidence, 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 $\hat{\tau}$ relative to the current step t, i.e. $\hat{\tau} - t > 0$, the model becomes more confident and its entropy decreases. Conversely, when $\hat{\tau} - t < 0$, confidence decreases and entropy increases. At the same time, the KL divergence grows approximately proportionally to the distance $|\hat{\tau} - t|$, indicating progressively larger deviations from the original distribution as the steering target moves further away from the current denoising step.

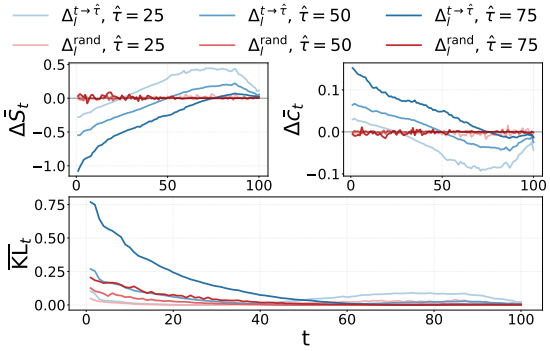


Figure 4: **Steering on layer 29 downstream effects**. We steer the activations of layer 29 using Equation (4) targeting different $\hat{\tau}$ values (blue), and compare the resulting entropy, confidence, and KL-divergence variations against norm-matched Equation (5) perturbations (red). Steering along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

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 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. This suggests that the models may be able to progressively compensate

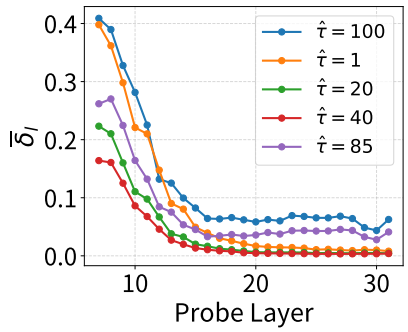


Figure 5: **Average probe drift across layers and steps after steering.** The model progressively compensates for the perturbation, reducing the discrepancy between clean and steered representations of τ.

for the injected perturbation as computation proceeds through subsequent layers. Combined with the probe analysis in Figure 2, these results suggest that τ information is not merely propagated unchanged across layers: early perturbations can be attenuated by later computation. 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?

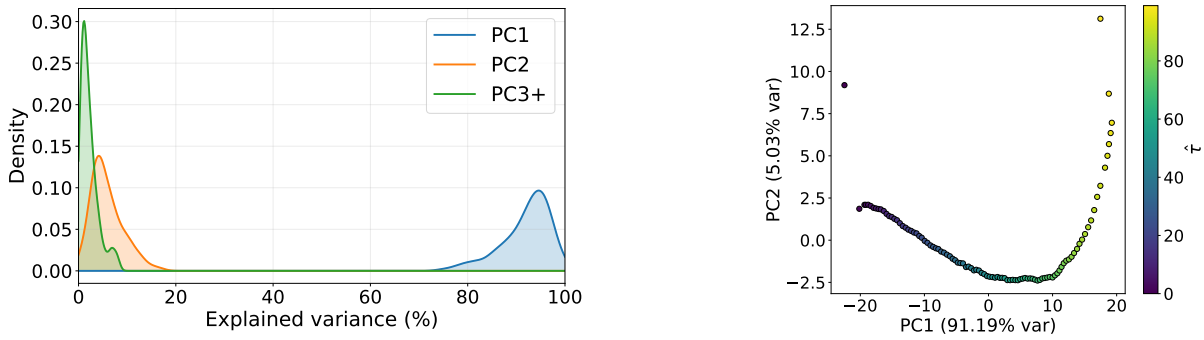
To explain the null result obtained when steering early layers, we investigate this phenomenon more deeply and measure how the probe predictions evolve after steering. Specifically, we track the average deviation between the probe outputs of clean and steered activations across layers

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

Starting from a perturbation injected at layer 6, we evaluate how different steering targets $\hat{\tau}$ 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 $\hat{\tau} = 100$ remains harder to compensate for, leaving a persistent discrepancy even in deeper layers. This observation also explains the sharp peaks previously observed for shallow-layer steering: 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 D). Having established that this



(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 PCs 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.

signal affects the models' output distributions in a predictable way, we now move on to characterising its properties.

✓ 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 low-dimensional 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 7 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). These results suggest that the mean vectors concentrate in a highly structured low-dimensional subspace, motivating a closer examination of whether this geometry can be consistently described across layers and aggregated into a unified representation of the model-level structure.

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 faithfully captures 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 $\hat{\tau}$. This yields a mean-standardised projection across layers. From this procedure, we obtain a new non-parametric 2D common trajectory that represents the standardised expectation of the projections for each τ index. We report the resulting plot in Figure 7 for LLaDA (see Appendix E for similar results on Dream). Remarkably, all points

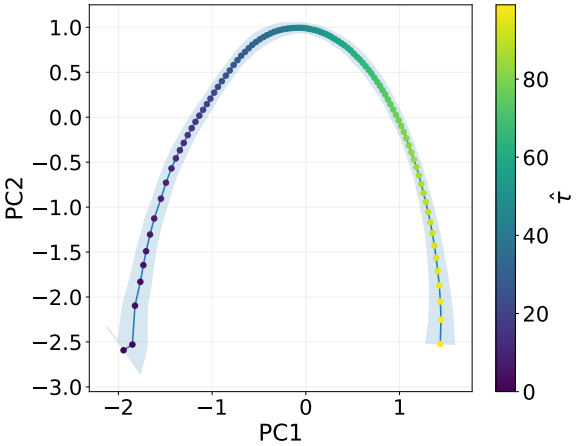


Figure 7: **The shared 2-dimensional geometry of $\hat{\tau}$ in LLaDA.** We compute an average trajectory by taking the standardised 2D PC projections of the mean vectors across the model. The parabolic geometry is shared across layers. PC1 and PC2 are standardised to unit variance.

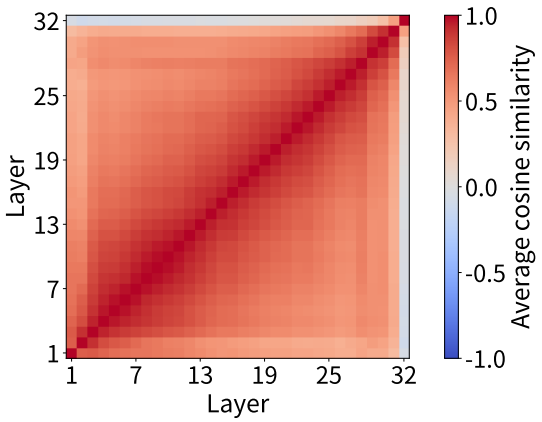


Figure 8: **Average cosine similarity of same-indexed vectors across layers in LLaDA.** We observe that most layers maintain a highly correlated representation of the same index t, and that this relationship degrades with distance. Layer 32 instead maintains a representation that is largely independent of the other layers.

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 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 8 for LLaDA and in Appendix E 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. 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 9. 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,

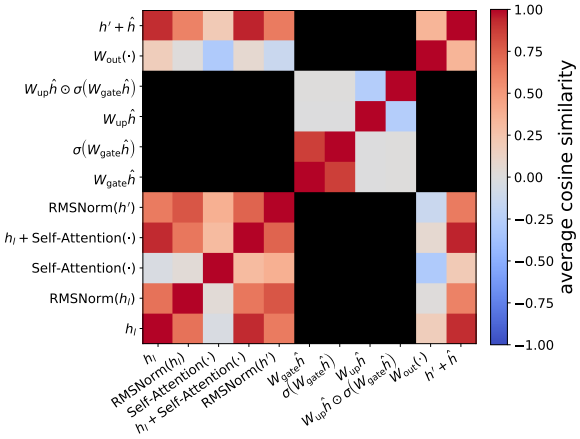


Figure 9: **Average cosine similarity across layer components in LLaDA.** We clearly see how across most of the internal representations there is high-correlation or almost complete orthogonality. We note how self-attention and the outputs of the out matrix seem to be anticorrelated.

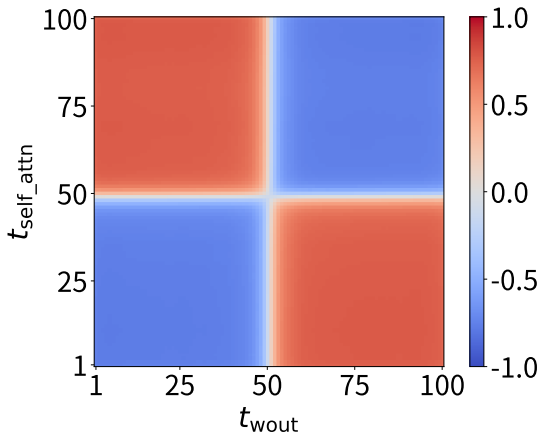


Figure 10: **Average cosine similarity of self-attention and post-W_{out} vectors.** The vectors exhibit strong structural behaviour, with representations for different indices t becoming nearly opposite. Moreover, there is a clear structural region around $t = 50$ where all vectors are approximately perpendicular.

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 10. 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 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 counting task. We similarly use mean vectors, but study a distributed denoising-time signal and its propagation across layers.

DLM Steering 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 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.

8 Limitations

Although our work aims to characterise and explain the representation of τ inside DLMs, several open questions remain. We first note that our analysis is restricted to LLaDA and Dream, two DLMs trained with the same cross-entropy loss function over [MASK] tokens. Thus, it remains unexplored how the τ phenomenon emerges in other types of DLMs that also do not receive timestep conditioning, such as block-diffusion models Cheng et al. (2025); Arriola et al. (2025); Wu et al. (2025).

9 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. *arxiv*.

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.

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.

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*.

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*.

MetaAI. 2024. Introducing llama 4: Advancing multimodal intelligence. *arXiv*.

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*.

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.

Gemma Team. 2025. Gemma 3. *ArXiV*.

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*.

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

A Denoising Progress and the Training Noising Level

This appendix clarifies the relationship between the training noising level s and the empirical denoising-progress variable τ_t used throughout the paper. 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

$$q_{s|0}(\cdot \mid x_0) = \prod_{j=1}^{L} q_{s|0}^j(\cdot \mid x_0^j).$$

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

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

The pretraining objective can then be written compactly as

$$\mathcal{L}_{\text{pre}} = -\mathbb{E}_{x_0,s,x_s}\left[\frac{1}{s}\sum_{j\in M(x_s)} \log p_\theta(x_0^j \mid x_s)\right]. \tag{7}$$

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

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

The SFT objective is

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{p_0,r_0,s,r_s}\left[\frac{1}{s}\sum_{j\in M(r_s)} \log p_\theta(r_0^j \mid p_0, r_s)\right]. \tag{8}$$

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

$$\begin{aligned} M_R(x_t) &:= \{j \in R : x_t^j = [\text{MASK}]\}, \\ U_R(x_t) &:= \{j \in R : x_t^j \neq [\text{MASK}]\}. \end{aligned} \tag{9}$$

These sets partition the response positions, so

$$|M_R(x_t)| + |U_R(x_t)| = L.$$

We define the empirical denoising progress at step t as

$$\tau_t := \frac{|U_R(x_t)|}{L}. \tag{10}$$

Its complement is the remaining response-local mask ratio:

$$1 - \tau_t = \frac{|M_R(x_t)|}{L}. \tag{11}$$

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 $r_s \sim q_{s|0}(\cdot \mid r_0)$,

$$\mathbb{E}_{r_s}\left[\frac{|M(r_s)|}{L}\right] = s. \tag{12}$$

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

$$\mathbb{E}_{r_s}\left[\frac{|U(r_s)|}{L}\right] = 1 - s. \tag{13}$$

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 used a 5-layer residual neural network with LayerNorm+GELU blocks and bounded the outputs to the interval $(0, 1)$ by applying a sigmoid function to the MLP logits, since the target τ is itself bounded. We trained the probes for 20 epochs using 300 training examples and 100 validation examples. For each example and at each epoch, we dynamically varied the generation length and

number of steps to improve the probe's coverage. We implemented batch gradient descent by updating the parameters after each full example, i.e., by batching the gradients over the entire denoising stage. We employed the 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 previously-introduced experiments were conducted on a H100 Nvidia chip.

C Additional Experiments on Probing and Steering τ

In this section, we present additional results concerning the ability of the MLP probes to correctly identify the value of τ and the ability of the mean activation vectors to steer the model.

Figure 11 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.

We also study the performance of the mean-vector approach on Dream. Figure 12 shows the downstream effects of steering Dream in layer 25. We observe a similar trend as in LLaDA, where entropy decreases and confidence increases when we steer towards later denoising steps.

Figures 18 and 19 show the downstream results for other layers of LLaDA and Dream, for multiple layers and targets $\hat{\tau}$. Besides the previously introduced quantities $\Delta\bar{S}_{t,e}$, $\Delta\bar{c}_{t,e}$ and $\overline{\mathrm{KL}}_{t,e}$ (here, e indexes the experiment conducted), we also show the top-K token overlap, defined as

$$\overline{\mathrm{TK}}_{t,e} := \mathbb{E}_n \left[\mathbb{E}_j \left[\frac{\left| T_K(\tilde{p}_{t,e,n,j}) \cap T_K(p_{t,n,j}) \right|}{K} \right] \right].$$

The top-K token overlap measures the similarity between the sets of highest-probability tokens predicted by two distributions. We compute the fraction of shared tokens in their top-K predictions and average it across samples and positions. We observe that the effects are weaker when steering is applied in the initial and middle layers of the model.

In Figure 17 we ablate the effect of generation length. We repeat 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 $\hat{\tau}$, indicating that our findings are not an artefact of a particular generation length.

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

			$\hat{\tau}$				
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

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 $\hat{\tau}$ 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 $\hat{\tau} = 100$.

D Depth Correction

Here we provide a finer-grained view of the depth-correction phenomenon discussed in Section 4.3.

Figure 13 shows the per-step version of Figure 5 for $\hat{\tau} = 100$: instead of averaging the probe drift $\bar{\delta}_k$ over denoising steps, we plot one column per step, so that averaging the columns recovers the $\hat{\tau} = 100$ 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 E), and indicates that the model's ability to correct an injected $\hat{\tau}$ 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 $\hat{\tau} = 100$, i.e. as t approaches the end of the denoising trajectory.

E Mean vector subspace extra experiments

In this section, we present additional results regarding the mean activations obtained for Dream. Similarly to the behaviour observed in LLaDA, we find that the representations associated with τ concentrate in structured low-dimensional subspace,

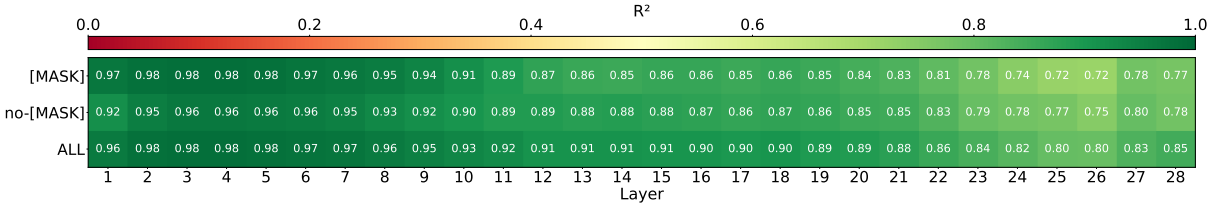


Figure 11: **R^2 coefficients of MLPs in Dream**. The R^2 coefficient slightly degrades as we probe deeper into the models, which remains consistent with the observations of Figure 2.

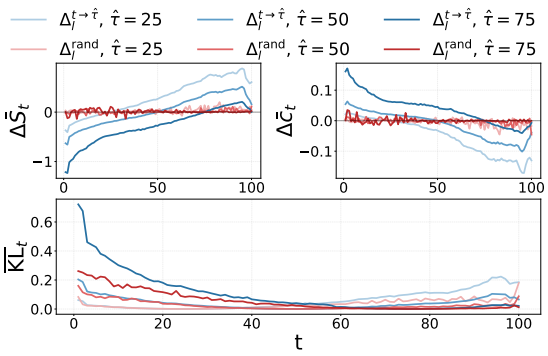


Figure 12: **Mean-steering downstream effects on layer 25 of Dream**. We steered the activations in layer 25 using the mean-ratio vectors (blue) targeting different τ values, and measured the variation in entropy, confidence and the KL divergence. We compared it against random perturbations (red). Steering with the mean vectors has an effect that is consistent with what would be expected.

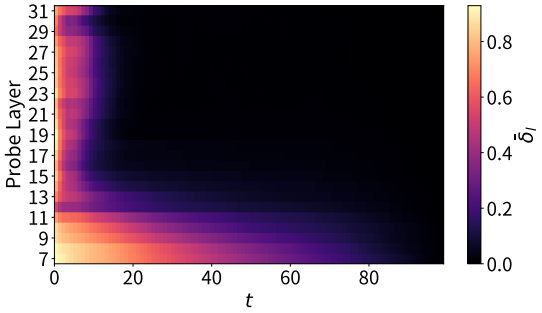


Figure 13: **Probe drift across layers and denoising steps after steering toward $\hat\tau = 100$ at layer 6.** Each column shows the absolute drift $\bar\delta_l$ between clean and steered probe predictions at a given denoising step t, resolved across probe layers.

exhibiting a smooth and ordered trajectory across denoising time.

Figure 14 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-like trajectory, supporting our claim of a shared geometry of τ across the whole model.

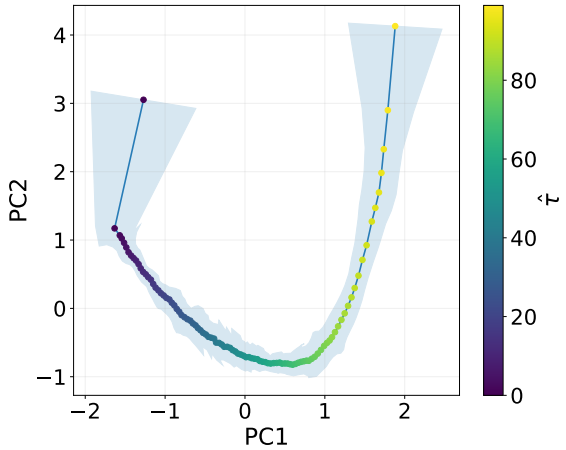


Figure 14: **The shared 2-dimensional geometry of $\hat\tau$ in Dream.** We compute an average trajectory by taking the standardised 2D PC projections of the mean vectors across the model. The parabolic geometry is shared across layers, similarly to LLaDA. PC1 and PC2 are standardised to unit variance.

We also investigated how the $\hat\tau$ representations align across layers for Dream, following the analysis conducted for LLaDA in Figure 8. In contrast to LLaDA, which exhibits strong global alignment across most layers, Figure 15 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.

Finally, Figure 16 presents a 3D PCA projection of the mean vectors from layer 25 of Dream. Again, the resulting trajectory reveals a low-dimensional, ordered structure. Interestingly, however, 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.

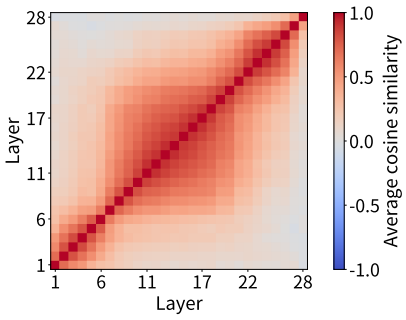


Figure 15: **Average mean vector cosine similarity for Dream.** Noticeably, Dream presents a much more orthogonal relationship than LLaDA, showing how layers are more independent in their representation of τ. Moreover, we see three structured blocks of high-similarity density which align with the model's computation stages 1-6 (early), 7-20 (middle), 21-27 late).

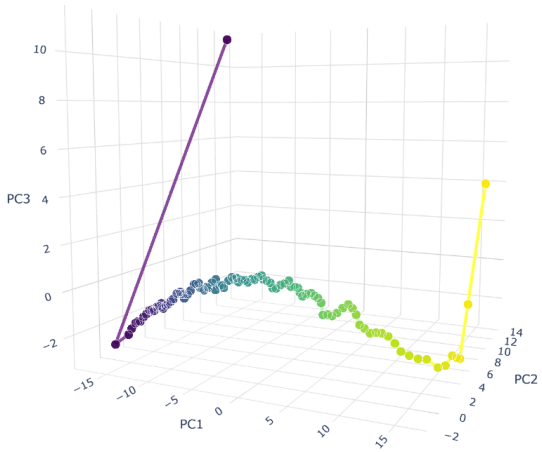
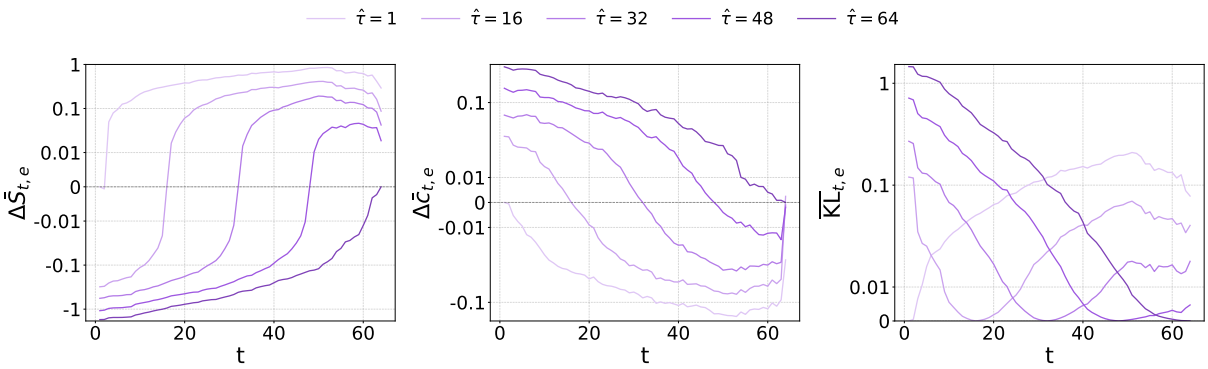
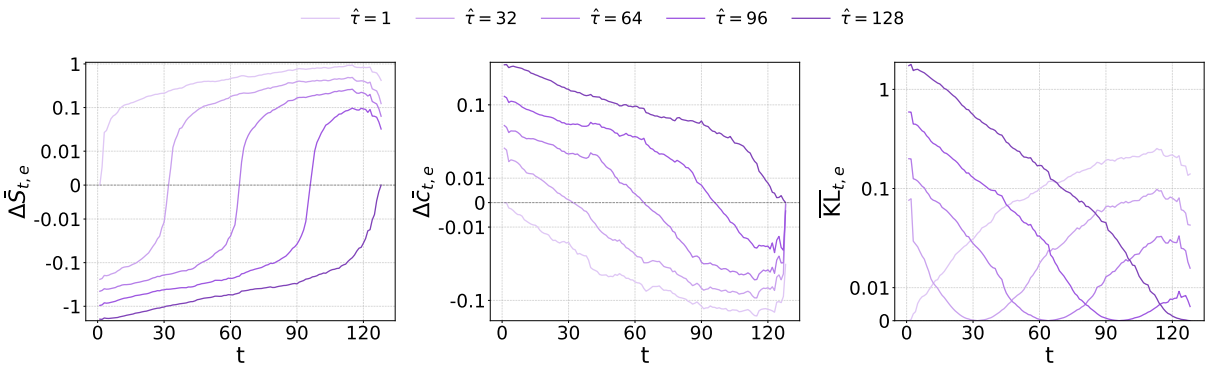


Figure 16: **3D visualisation of the PCA projection of the mean vectors for layer 25 of Dream.** Brighter colours indicate values closer to $\hat{\tau} = 100$. As in LLaDA, we again observe a low-dimensional ordered structure. Notably, two points, corresponding to $t = 1$ and $t = 100$, lie far from the rest of the geometry, suggesting a characteristic representation of these extreme denoising moments.



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



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

Figure 17: **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, $\Delta\bar{S}_{t,e}$, $\Delta\bar{c}_{t,e}$ and $\overline{\mathrm{KL}}_{t,e}$ versus the denoising step t, for five steering targets $\hat{\tau}$. Vertical axes use a symmetric-log scale to resolve the convergence near 0.

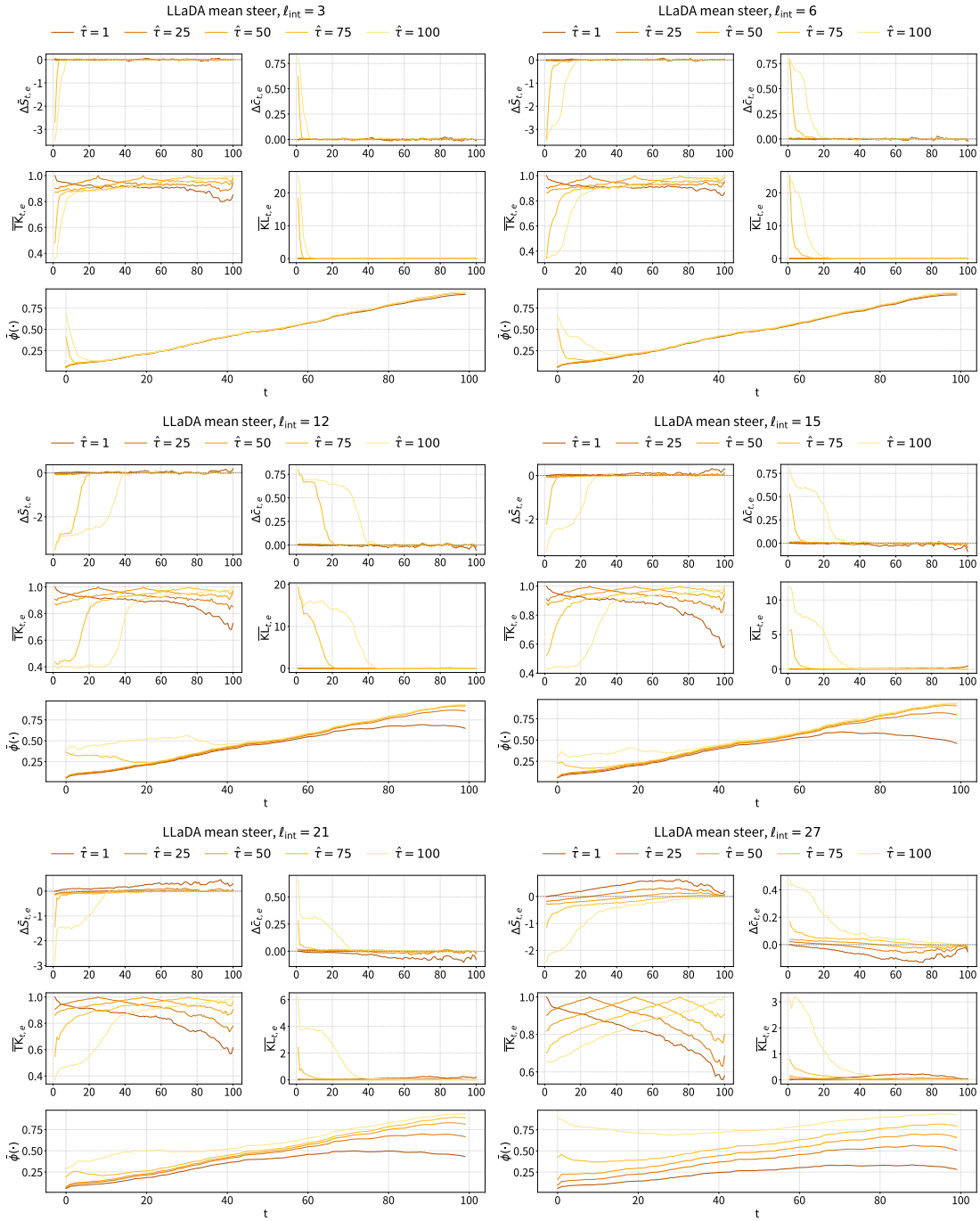


Figure 18: **Layer-wise mean-steering effects in LLaDA.** We apply mean-vector steering at different intervention layers l_{int} and target denoising-progress bins $\hat{\tau} \in \{1, 25, 50, 75, 100\}$. Each panel reports the downstream effect on entropy drift, confidence drift, top-K overlap, and KL divergence across denoising steps.

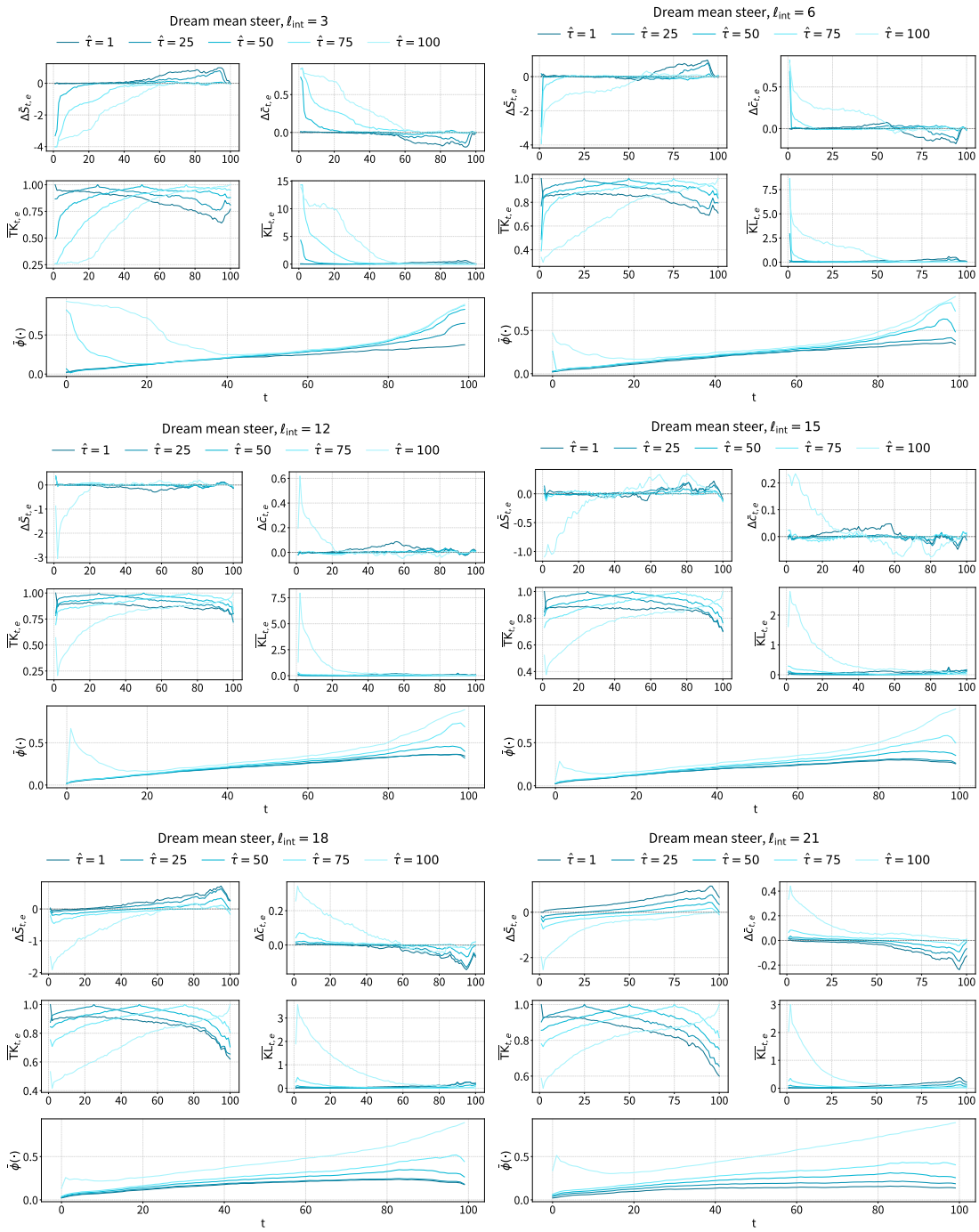


Figure 19: **Layer-wise mean-steering effects in Dream.** We apply mean-vector steering at different intervention layers l_{int} and target denoising-progress bins $\hat{\tau} \in \{1, 25, 50, 75, 100\}$. Each panel reports the downstream effect on entropy drift, confidence drift, top-K overlap, and KL divergence across denoising steps.