How Long Does a Model Need to See? When Vision Becomes State

Anonymous authors
Paper under double-blind review

Abstract

Diffusion vision-language models (DVLMs) keep the image in the computation throughout denoising, tying seeing to answer formation by design. We ask a simple question: *how long does a model actually need to see?* By terminating visual computation at different stages, we find that seeing and thinking have different lifetimes: visually grounded answers can continue to form after the image has left the computation. Visual influence changes substantially over denoising and can re-emerge after becoming weak, while visual attention largely misses this transition. Meanwhile, visual information becomes increasingly reflected in the evolving generation state, whose structure reveals whether vision will matter again later. This signal is surprisingly simple to read out: a 305-parameter predictor can forecast future visual influence, determine when visual computation can end, and transfer across DVLMs. It preserves approximately full task performance while substantially reducing decoding attention and visual KV usage. Together, these results reveal a simple visual lifecycle in diffusion multimodal generation: seeing can end once vision becomes state.

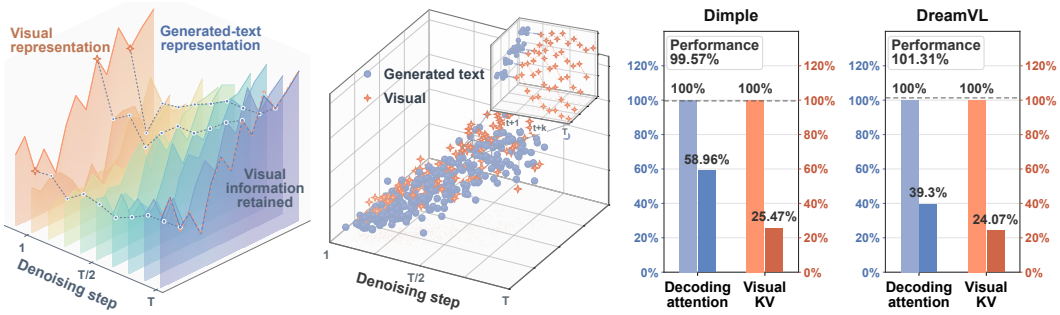

Figure 1: **Visual information moves into the generation state, which reveals future visual need and enables early visual exit.**

1 Introduction

Diffusion vision-language models (DVLMs) (Xin et al., 2025; Rojas et al., 2025; Zeng et al., 2026; Li et al., 2026b; You et al., 2026) keep the image in the computation throughout answer formation. At every denoising step, the model can revisit the same visual evidence while progressively resolving the answer. Seeing and thinking are therefore coupled by design: as long as generation continues, visual computation continues as well (Kim et al., 2026; Chang et al., 2025; Lee et al., 2026). This raises a simple question:

> **Research Question:** *Must a model keep seeing in order to keep thinking visually?*

We begin with a direct intervention. At different denoising stages, we remove visual keys and values and allow the remaining generation process to continue. In many cases, the model still completes the visually grounded answer after the image has already left the computation, while extending visual

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

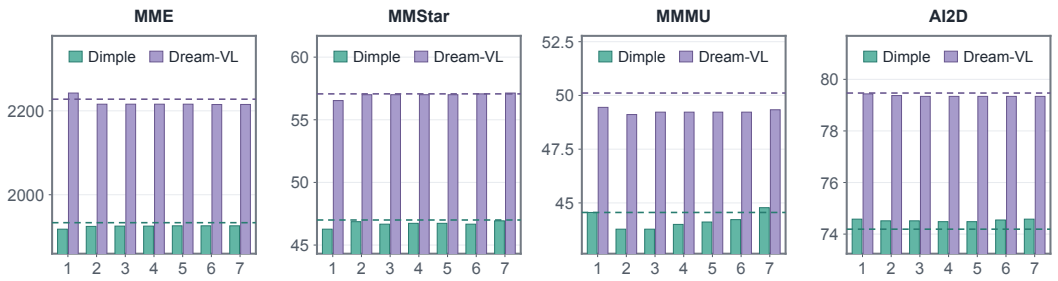

Figure 2: **Task performance under different visual exit steps.** Visual KV are permanently removed from the selected step onward. Longer visual computation does not consistently improve the final answer.

computation further does not consistently improve the final result. We call the direct participation of visual representations in the current denoising step *visual computation*, and the continued formation of an image-grounded answer *visual thinking*. Their lifetimes can clearly diverge.

First, seeing and thinking follow different temporal trajectories. To trace this separation inside generation, we compare unresolved predictions from the same state with and without visual input. The effect of vision changes substantially across denoising steps and can rise again after becoming small. Visual attention, in contrast, remains comparatively stable. The model can therefore keep allocating computation to the image while the role of that computation in shaping the answer has already changed.

This observation leads to a deeper question: where does visual information go as direct visual computation becomes less necessary? We find a clear signature in the evolving generation state. As denoising proceeds, generation representations progressively move toward the visual representational region, showing increasing geometric overlap and decreasing distance in the original hidden space.

Second, the generation state carries information about future visual need, as summarized in Figure 1. The geometric change is accompanied by a functional one. Even among states at the same denoising step with similar current visual influence, nearby generation representations tend to exhibit more similar future visual requirements. The current state therefore contains information about whether later steps will need to see the image again.

This reveals a gradual internalization process in diffusion multimodal reasoning. Visual information first enters through explicit computation on the image and becomes increasingly reflected in the state in which the answer is being formed. As this state develops, it begins to carry the information required for later visually grounded generation.

The strength of this signal is especially striking. A lightweight MLP with only 305 trainable parameters can read the current generation state and predict the maximum remaining visual influence over the future trajectory. Its training itself takes only seconds, yet the learned signal can determine when visual computation may end and transfer directly across DVLMs. Such a simple readout producing stable cross-model behavior indicates that future visual need is already clearly encoded in the generation state.

Our contributions are threefold:

- We reveal a temporal decoupling between **visual computation** and **visual thinking** in DVLMs: visually grounded answer formation can continue after explicit computation on the image has ended.
- We trace this process through intervention, attention, and hidden representations, and show that visual information becomes progressively reflected in the evolving generation state, whose structure further reveals whether vision will be needed again.
- We show that this internal signal is highly readable: a 305-parameter predictor trained at negligible cost can forecast future visual need and transfer across different DVLMs, providing functional evidence for the representation-level mechanism.

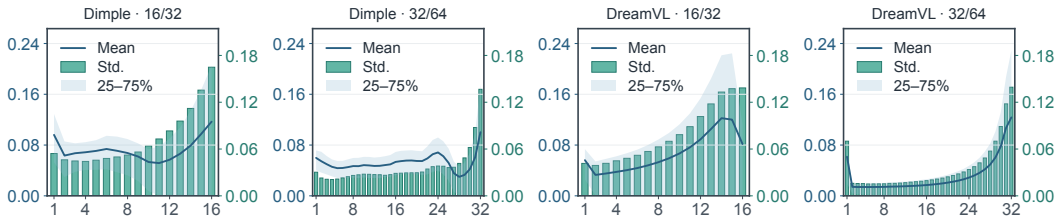

Figure 3: **Visual influence over denoising under different model and generation settings.** J_t measures the change in unresolved predictions caused by removing visual computation from the same generation state.

Overall, our results suggest a dynamic view of diffusion multimodal reasoning in which visual information gradually moves from sustained external computation into the evolving internal state. **A model does not need to keep seeing in order to keep thinking visually.**

2 Seeing and Thinking Have Different Lifetimes

Standard diffusion inference keeps visual representations active until denoising ends. We break this coupling by permanently removing visual keys and values at controlled steps and allowing the generation state already formed to continue evolving.

2.1 Thinking Continues After Seeing Stops

For an exit step τ, we retain visual KV for $t < \tau$ and remove them permanently for all $t \geq \tau$. The subsequent denoising trajectory therefore evolves entirely from the generation state established before the intervention.

Figure 2 reveals a clear temporal separation. Removing vision at the earliest stages causes substantial degradation, while after this initial period, much earlier exits already preserve performance close to full visual computation. The answer can therefore continue developing after direct visual computation has ended. Additional qualitative trajectories showing the same behavior are provided in Appendix F.

Visual thinking can outlive visual computation. The image may leave the computation while the visually grounded answer continues to form.

This separation raises a finer question: *how does the influence of vision evolve before seeing becomes dispensable?*

2.2 Visual Influence Evolves During Thinking

At each denoising step t, let U_t denote the unresolved positions. From the same state on the normal unpruned trajectory, we evaluate one forward pass with visual representations and one without them:

$$P_{t,i} = \mathrm{softmax}\left(\mathbf{z}_{t,i}^{V}\right), \qquad Q_{t,i} = \mathrm{softmax}\left(\mathbf{z}_{t,i}^{\neg V}\right), \quad i \in U_t. \tag{1}$$

We define the visual influence at step t as the mean Jensen–Shannon divergence between the two unresolved prediction distributions:

$$J_t = \frac{1}{|U_t|} \sum_{i \in U_t} \mathrm{JS}\left(P_{t,i}, Q_{t,i}\right). \tag{2}$$

Because the two passes share the same preceding generation state, J_t directly measures how much visual computation changes the unresolved predictions at the current step.

Figure 3 shows that visual influence evolves substantially across denoising, with no common monotonic trajectory. In many cases, J_t becomes small at an intermediate stage and rises again later.

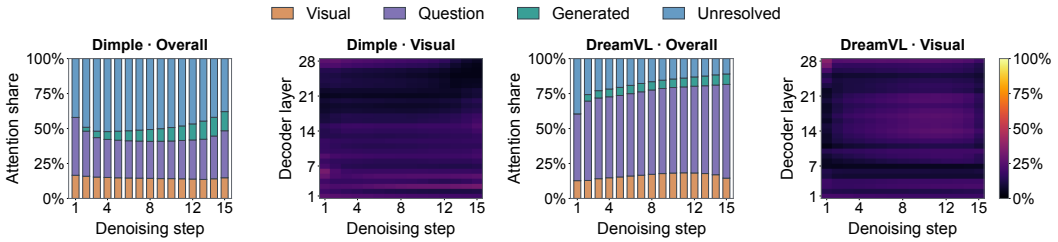

Figure 4: **Visual attention from unresolved generation positions in Dimple and DreamVL.** Despite the changing visual influence measured by J_t, attention to visual tokens remains comparatively stable throughout denoising.

We therefore characterize the strongest visual influence remaining after the current state by

$$y_t = \max_{t<s\leq T} J_s, \tag{3}$$

where T is the final denoising step. J_t describes how much vision matters now, while y_t captures how much it may matter again later.

> **Finding 1:** *Visual computation and visual thinking follow different lifetimes, while visual influence can fade and re-emerge during answer formation.*

This leaves the central question for the next section: whether the current generation state already carries information about the future quantity y_t.

3 Where Does Visual Thinking Go?

Section 2 shows that visually grounded answer formation can continue after direct visual computation has ended. We therefore ask where the visual information supporting the remaining generation appears inside the model.

We first examine visual attention. Figure 4 tracks attention from unresolved generation positions to visual tokens throughout denoising.

As shown in Figure 4, visual tokens continue to receive substantial attention throughout generation, while the intervention response J_t changes markedly across steps. The transition in visual influence is therefore largely absent from the attention trace. Complete layer-by-step patterns for both models are provided in Appendix C.

> **Observation:** *The handoff from direct visual computation to internal visual thinking is largely invisible in visual attention.*

We next trace the evolving generation state in which the answer itself is being formed.

3.1 Visual Information Moves into the Generation State

From normal unpruned trajectories, we extract final-layer hidden states. Let $\{\mathbf{v}_j\}_{j=1}^{N_v}$ denote visual representations fixed at the first denoising step, and $\{\mathbf{h}_{t,i}\}_{i=1}^{N_g}$ the generation-slot representations at step t.

For each model, we fit a shared two-dimensional PCA projection π and keep it fixed across denoising steps. Figure 5 and Figure 6 shows the resulting evolution in Dimple and DreamVL.

As denoising proceeds, generation representations progressively expand toward regions occupied by the fixed visual representations. Complete 16-step trajectories for Dimple, DreamVL, and LaViDa are provided in Appendix D.

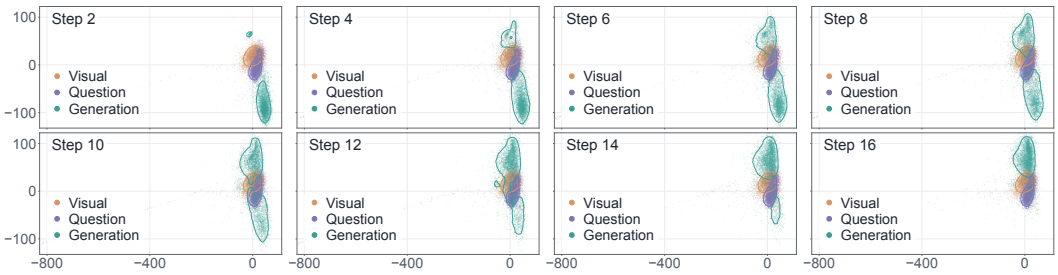

Figure 5: **Evolution of generation representations in Dimple under a fixed PCA projection.** Visual representations remain fixed while generation representations progressively occupy the visual representational region.

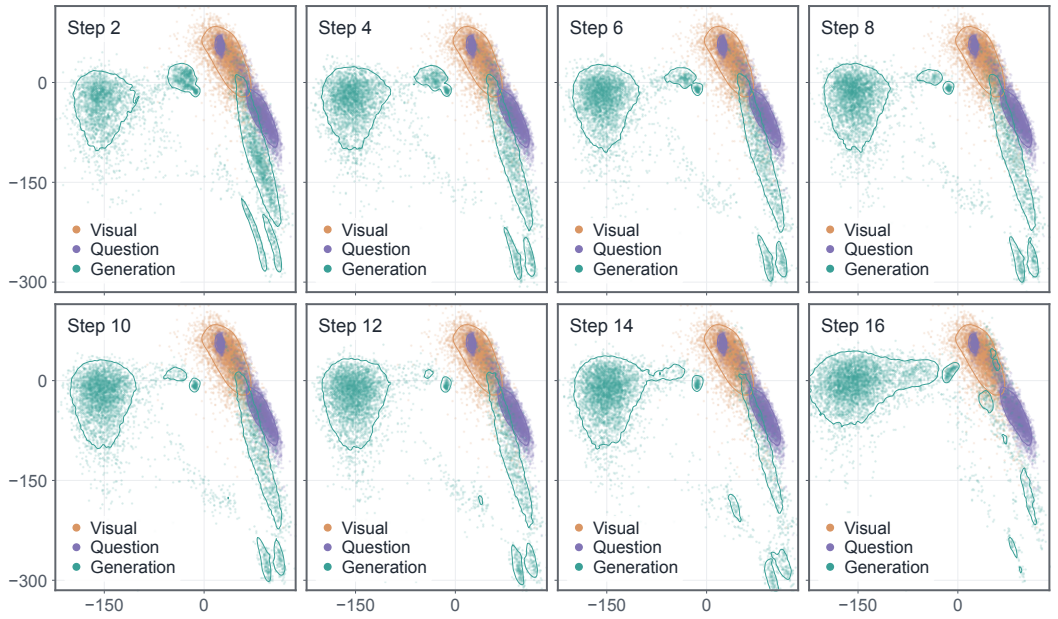

Figure 6: **Evolution of generation representations in DreamVL under a fixed PCA projection.** Visual representations remain fixed while generation representations progressively occupy the visual representational region.

We quantify this evolution using two complementary measures. In the shared PCA plane, we measure the fraction of visual points contained in the convex hull of the generation representations:

$$C_t = \frac{1}{N_v} \sum_{j=1}^{N_v} \mathbf{1} \left[\pi(\mathbf{v}_j) \in \operatorname{conv} \left(\{ \pi(\mathbf{h}_{t,i}) \}_{i=1}^{N_g} \right) \right]. \tag{4}$$

We also measure proximity directly in the original hidden space through the nearest generation representation:

$$D_t = \frac{1}{N_v} \sum_{j=1}^{N_v} \min_{1 \le i \le N_g} \left(1 - \frac{\mathbf{v}_j^\top \mathbf{h}_{t,i}}{\|\mathbf{v}_j\|_2 \|\mathbf{h}_{t,i}\|_2} \right). \tag{5}$$

Both quantities are computed per example and averaged with equal weight.

Table 1: **Visual–generation representation geometry from the first to the final denoising step.**

Model	2D Containment Ratio ↑	High-Dimensional Cosine Distance ↓
Dimple	2.22% → **61.47%**	0.701 → **0.588**
DreamVL	2.37% → **11.65%**	0.673 → **0.584**

Table 1 shows the same evolution in both models: containment increases while high-dimensional distance decreases. Together with the PCA trajectories, these measurements show that the evolving generation state increasingly approaches and occupies the visual representational region.

Visual information thus becomes progressively reflected in the state in which the answer is being formed. We next ask whether this state also carries information about what the remaining trajectory will require.

3.2 The State Knows Whether It Will Need to See Again

We match generation states by denoising step t and current visual influence J_t, and test whether their internal representations organize the future quantity y_t. This compares states with similar generation progress and immediate visual influence.

After denoising step t, let U_{t+1} denote the unresolved positions. We summarize their final-layer state and cross-step change as

$$\bar{\mathbf{h}}_t = \frac{1}{|U_{t+1}|} \sum_{i \in U_{t+1}} \mathbf{h}_{t,i}, \qquad \Delta\bar{\mathbf{h}}_t = \frac{1}{|U_{t+1}|} \sum_{i \in U_{t+1}} (\mathbf{h}_{t,i} - \mathbf{h}_{t-1,i}). \tag{6}$$

For the first denoising step, we set $\Delta\bar{\mathbf{h}}_1 = \mathbf{0}$. We use the logarithmic future visual influence

$$\ell_t = \log_{10}(y_t + \epsilon), \qquad \epsilon = 10^{-6}. \tag{7}$$

For each query state, we construct a reference candidate set from the same denoising step with similar current J_t. Within this matched set, neighbors are selected using visual attention, $\bar{\mathbf{h}}_t$, or $[\bar{\mathbf{h}}_t; \Delta\bar{\mathbf{h}}_t]$. Preprocessing is fitted only on the reference partition; full construction details are provided in the appendix.

Let $\mathcal{N}_{q,t}^{(\mathbf{z})}$ denote the k-nearest neighbors of query q under feature $\mathbf{z}$. We measure their difference in future visual influence as

$$E_{q,t}^{(\mathbf{z})} = \frac{1}{k} \sum_{j \in \mathcal{N}_{q,t}^{(\mathbf{z})}} \left| \ell_t^{(q)} - \ell_t^{(j)} \right|. \tag{8}$$

Smaller $E_{q,t}^{(\mathbf{z})}$ indicates that nearby states share more similar future visual influence. We additionally compare against random neighbors and examine five representation-distance quantiles.

Figure 7 shows that representation geometry organizes future visual influence. Nearby generation states exhibit more similar future visual requirements than distant states, and representation-based neighbors consistently yield smaller future-influence differences than random neighbors across all model and generation settings.

Because these states are already matched by denoising step and current J_t, the remaining variation is encoded in the generation state itself: states that look similar internally tend to require similar amounts of visual computation later.

> **Finding 2:** *The generation state knows whether the model will need to see again. Even at matched generation progress and current visual influence, similar generation states exhibit more similar future visual needs.*

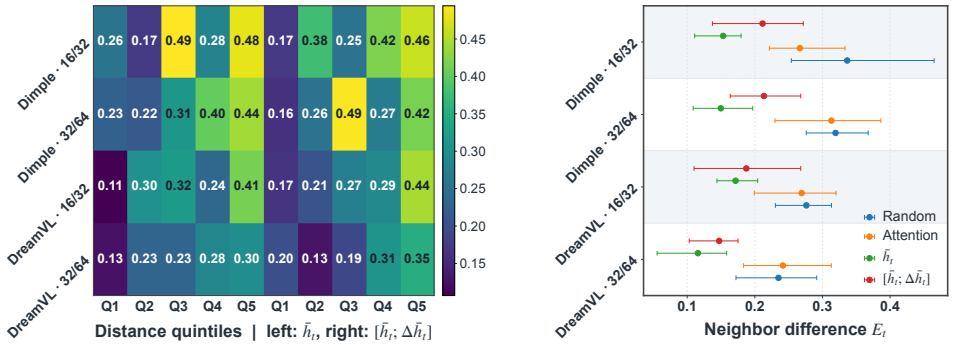

Figure 7: **Current generation-state geometry and future visual influence.** Candidate states are matched by denoising step and current J_t. Distance-quantile and nearest-neighbor analyses measure whether representation similarity corresponds to similarity in the future quantity y_t.

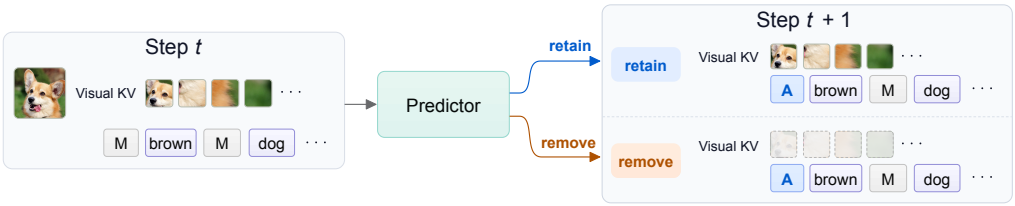

Figure 8: **Overview of the state-guided visual exit.**

Together, the geometric and functional evidence reveal a representation-level handoff: visual information becomes increasingly reflected in the evolving generation state, which also carries information about future visual influence.

This leads to a direct readout test: *how easily can y_t be recovered from the current generation state?*

4 Knowing When Seeing Can End

Section 3.2 shows that future visual influence is reflected in the current generation state. We now test how easily this signal can be read out.

4.1 A Lightweight Readout of Future Visual Influence

At each nonterminal step t, we concatenate the current unresolved generation state and its cross-step change, and use a lightweight predictor to estimate the logarithmic future visual influence defined in Section 2.2. Figure 8 summarizes this state-to-exit process.

After feature preprocessing, f_θ contains only 305 trainable parameters; architecture and training details are provided in the appendix. We recover the predicted future visual influence as $\hat{y}_t = 10^{\hat{\ell}_t} - \epsilon$ and terminate visual computation at

$$\mathbf{x}_t = [\bar{\mathbf{h}}_t \| \Delta\bar{\mathbf{h}}_t], \qquad \mathbf{u}_t = (\mathbf{x}_t, t/T), \qquad \hat{\ell}_t = f_\theta(\mathbf{u}_t), \qquad \mathcal{L}(\theta) = \mathbb{E}\left[(\hat{\ell}_t - \ell_t)^2\right]. \tag{9}$$

$$t_{\text{exit}} = 1 + \min\left(\{t : 1 \le t < T,\ \hat{y}_t \le \delta\} \cup \{T\}\right). \tag{10}$$

Visual KV are removed from t_{exit} onward, while denoising continues from the current generation state. The intervention-based J_t is used only to construct training supervision; inference reads only hidden states already produced by the normal forward pass.

Table 2: **Performance and computation of state-guided visual exit.** Retention is computed as $\mathrm{Ours}/\mathrm{Base} \times 100\%$. Decode attention saving excludes the first denoising step. Visual KV reduction measures the decrease in average visual KV participating per layer and decoding step.

Metric	SQA	AI2D	MMStar	POPE	MMB	MMB-CN	MMMU	Avg.
Dimple *8 Steps / 16 Tokens*								
Base	78.53	74.19	47.00	84.06	72.45	70.67	44.56	67.35
Ours	78.28	74.58	46.20	83.44	71.98	70.51	44.89	67.13
Retention	99.68%	**100.53%**	98.30%	99.26%	99.35%	99.77%	**100.74%**	99.66%
Avg.Exit step	3.07	3.06	2.67	3.27	2.68	3.16	3.27	3.03
Decode Attn. Saving	40.02%	51.47%	49.11%	58.72%	50.53%	46.03%	38.37%	47.75%
Visual KV Reduction	73.86%	74.17%	79.53%	71.69%	79.15%	72.96%	68.72%	74.30%
Dimple *16 Steps / 32 Tokens*								
Base	78.68	74.03	47.13	84.29	72.29	70.20	44.56	67.31
Ours	78.24	74.22	47.07	83.53	71.98	70.20	44.00	67.03
Retention	99.44%	**100.26%**	99.87%	99.10%	99.57%	**100.00%**	98.74%	99.57%
Avg.Exit step	3.81	3.67	4.40	8.86	4.57	4.74	5.26	5.04
Decode Attn. Saving	38.89%	50.07%	42.43%	35.76%	42.83%	42.06%	35.22%	41.04%
Visual KV Reduction	82.55%	83.28%	79.03%	50.91%	77.37%	76.82%	71.72%	74.53%
DreamVL *8 Steps / 16 Tokens*								
Base	92.12	79.47	57.07	85.77	81.97	48.61	50.11	70.73
Ours	91.82	79.40	56.53	83.92	81.11	51.78	49.67	70.60
Retention	99.67%	99.91%	99.05%	97.84%	98.95%	**106.52%**	99.12%	**100.15%**
Avg.Exit step	3.64	3.81	3.65	2.18	3.26	6.32	3.31	3.74
Decode Attn. Saving	48.54%	63.26%	55.91%	85.83%	59.43%	20.39%	65.34%	56.96%
Visual KV Reduction	64.93%	64.65%	64.93%	85.58%	72.14%	24.94%	69.84%	63.86%
DreamVL *16 Steps / 32 Tokens*								
Base	92.02	79.18	57.00	85.78	80.88	42.34	50.56	69.68
Ours	91.32	79.40	56.67	84.04	79.72	48.14	50.56	69.98
Retention	99.24%	**100.28%**	99.42%	97.97%	98.57%	**113.70%**	**100.00%**	**101.31%**
Avg.Exit step	3.65	3.87	3.93	2.31	3.36	12.25	3.69	4.72
Decode Attn. Saving	55.00%	72.75%	61.91%	82.71%	62.64%	20.41%	69.45%	60.70%
Visual KV Reduction	82.69%	81.97%	80.11%	91.88%	85.49%	28.06%	81.29%	75.93%

4.2 Functional Validation through Visual Exit

Experimental setup. We evaluate the state readout on Dimple (Yu et al., 2025) and DreamVL (Ye et al., 2025) under two generation settings, covering SQA (Lu et al., 2022), AI2D (Kembhavi et al., 2016), MMStar (Chen et al., 2024), POPE (Li et al., 2023), MMBench-V1.1 (Liu et al., 2024), MMBench-CN-V1.1 (Liu et al., 2024), and MMMU (Yue et al., 2024). The predictor is trained on Dimple trajectories and applied directly to DreamVL with all parameters and preprocessing frozen.

A simple readout is sufficient. Table 2 shows that a 305-parameter predictor can terminate visual computation well before generation ends while maintaining approximately 100% task-performance retention. Across the four model and generation settings, this corresponds to roughly 41%–61% decoding-stage attention saving and 64%–76% visual KV reduction. The predicted exit points vary across tasks and trajectories, showing that future visual influence can be read directly from the evolving generation state.

The readout generalizes across models. The Dimple-trained predictor transfers directly to DreamVL without retraining, while the reverse DreamVL-to-Dimple transfer shows the same behavior in Appendix B. We further apply the same Dimple-trained readout to LaViDa and MMaDA, where it continues to produce meaningful visual exits with largely preserved task performance. Representative results are shown in Table 3, with complete results in Appendix B.2. These results show that future visual influence remains readable across different DVLMs.

Table 3: **State-guided visual exit on additional DVLMs under the 8-step / 16-token setting.**

	LaViDa					MMaDA-Base				
Metric	**AI2D**	**MME**	**MMB**	**POPE**	**Avg.**	**AI2D**	**MME**	**MMB**	**POPE**	**Avg.**
Base	68.78	1833.13	72.83	86.11	–	58.26	1011.55	27.01	72.21	–
Ours	68.75	1838.59	72.52	84.81	–	58.39	1029.53	27.01	72.20	–
Retention	99.96%	**100.30%**	99.57%	98.49%	99.58%	**100.22%**	**101.78%**	**100.00%**	99.99%	**100.50%**
Decode Attn. Saving	14.95%	67.69%	33.87%	84.29%	50.20%	19.24%	52.77%	11.20%	10.82%	23.51%
Visual KV Reduction	16.65%	72.23%	39.40%	89.19%	54.37%	18.07%	48.35%	10.64%	9.88%	21.74%

Qualitative behavior. Figure 9 shows the same pattern, with visually grounded answers continuing to develop after visual KV leave the computation. Additional qualitative trajectories are provided in Appendix F.

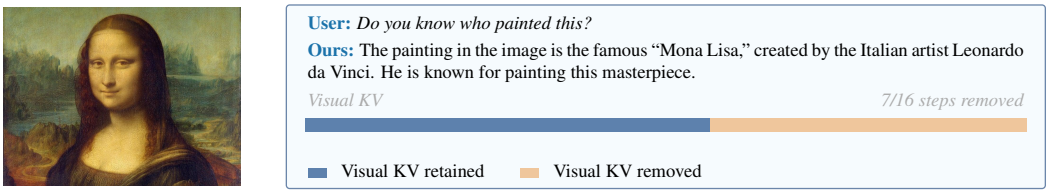

Figure 9: **Thinking continues after seeing ends.**

> **Finding 3:** *Future visual influence is readily readable from the generation state. A 305-parameter predictor can determine when seeing may end and transfer the same readout across DVLMs.*

5 Related Work

DVLMs. Recent DVLMs formulate vision-language generation as iterative denoising, supporting parallel token refinement, multimodal understanding, and unified multimodal generation (Yu et al., 2025; Li et al., 2025; Yang et al., 2025). Existing work has primarily focused on model design, training objectives, and generation capability. Much less is known about how visual information participates in the denoising process itself, especially how long direct visual computation remains necessary as the answer gradually forms.

Understanding and accelerating DVLMs. Recent work studies the internal dynamics and inference efficiency of DVLMs. Interpretability studies probe activation patterns, semantic anchors, and information flow (Zuo et al., 2026; Zhao et al., 2026b; Xu et al., 2026), while acceleration methods exploit redundancy in visual tokens, masked positions, and generation length (Chang et al., 2025; Li et al., 2026a; Zhao et al., 2026a; Cheng et al., 2026; Wan et al., 2026). Our work examines the temporal role of vision itself: when visual computation matters, how visual information enters the evolving generation state, and whether that state reveals future visual need.

6 Conclusion

We study whether visual computation must last as long as visual thinking in DVLMs. By terminating visual computation during denoising, we find that visually grounded answer formation can continue after the image has left the computation. This separation is reflected in the evolving generation state, which increasingly carries visual information and reveals whether vision will be needed again later. A lightweight predictor can read this signal to determine when visual computation may end. Together, these results suggest that seeing and thinking need not share the same lifetime in diffusion multimodal generation. **A model does not need to keep seeing in order to keep thinking visually.**

AI USE STATEMENT

In this work, we used generative AI tools for language editing, including grammar correction, wording refinement, and improving the clarity of written expressions; for assisting with code debugging; and for supporting literature search by helping identify potentially relevant prior work. All suggested references were manually verified by the authors for relevance and accuracy before inclusion. Generative AI was not used to generate research ideas, formulate the methodology, design experiments, analyze experimental results, or draw scientific conclusions. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted code debugging was independently verified and tested for correctness. We take full responsibility for the final content of this work, including all text, claims, results, references, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the internal behavior and inference efficiency of diffusion multimodal language models using existing models and publicly available benchmarks. It involves no human subjects, new user data collection, or newly acquired personal or sensitive information. The proposed analysis and visual-exit mechanism do not introduce new application-specific risks beyond those of the underlying multimodal models. We follow the licenses and usage conditions of all models and datasets used in this work.

REPRODUCIBILITY STATEMENT

All experiments in this work use publicly available DVLMs and benchmark datasets. The main text specifies the visual intervention, representation analysis, future visual influence, and state-guided visual-exit procedure. Appendix A provides complete implementation details, including feature preprocessing, predictor architecture, training data, hyperparameters, computational cost, and the full algorithm and models. Appendix E gives the mathematical characterization of the visual-influence measures and exit criterion. Additional cross-model transfer results are reported in Appendix B, while full attention patterns, representation trajectories, and qualitative examples are provided in Appendices C, D, and F, respectively. These details, together with the publicly available models and datasets used throughout the paper, are intended to support reproduction of all analyses and experiments.

REFERENCES

Shuochen Chang, Xiaofeng Zhang, Qingyang Liu, and Li Niu. D^{3}tom: Decider-guided dynamic token merging for accelerating diffusion mllms, 2025. URL https://arxiv.org/abs/2511.12280.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087. Curran Associates, Inc., 2024. doi: 10.52202/079017-0850. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2f8ee6a3d766b426d2618e555b5aeb39-Paper-Conference.pdf.

Shuang Cheng, Yuhua Jiang, Zineng Zhou, Dawei Liu, Tao Wang, Linfeng Zhang, Biqing Qi, and Bowen Zhou. SDAR-VL: Stable and efficient block-wise diffusion for vision-language understanding. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28882–28901, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1333. URL https://aclanthology.org/2026.acl-long.1333/.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran

He. MME: A comprehensive evaluation benchmark for multimodal large language models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/d79a27cf2772fe00be7f341efc0eb517-Abstract-Datasets_and_Benchmarks_Track.html.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Min Joon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pp. 235–251. Springer, 2016. doi: 10.1007/978-3-319-46493-0_15. URL https://doi.org/10.1007/978-3-319-46493-0_15.

Keuntae Kim, Mingyu Kang, and Yong Suk Choi. Thinking diffusion: Penalize and guide visual-grounded reasoning in diffusion multimodal language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5154–5164, June 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Yann Le Cun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks, Tricks of the Trade*, Lecture Notes in Computer Science LNCS 1524. Springer Verlag, 1998. URL http://leon.bottou.org/papers/lecun-98x.

Insu Lee, Wooje Park, Wonseok Shin, Jinwoo Son, and Byonghyo Shim. Visual information-guided parallel decoding for diffusion multimodal large language models, 2026. URL https://arxiv.org/abs/2608.26580.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. Lavida: A large diffusion language model for multimodal understanding. *CoRR*, abs/2505.16839, 2025. doi: 10.48550/ARXIV.2505.16839. URL https://doi.org/10.48550/arXiv.2505.16839.

Shufan Li, Jiuxiang Gu, Kangning Liu, Zhe Lin, Zijun Wei, Aditya Grover, and Jason Kuen. Sparselavida: Sparse multimodal discrete diffusion language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36103–36114, June 2026a.

Shufan Li, Yuchen Zhu, Jiuxiang Gu, Kangning Liu, Zhe Lin, Yongxin Chen, Molei Tao, Aditya Grover, and Jason Kuen. Lavida-r1: Advancing reasoning for unified multimodal diffusion language models, 2026b. URL https://arxiv.org/abs/2602.14147.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 292–305. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.20. URL https://doi.org/10.18653/v1/2023.emnlp-main.20.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VI*, pp. 216–233, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72657-6. doi: 10.1007/978-3-031-72658-3_13. URL https://doi.org/10.1007/978-3-031-72658-3_13.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=HjwK-Tc_Bc.

Kevin Rojas, Yuchen Zhu, Sichen Zhu, Felix X-F. Ye, and Molei Tao. Diffuse everything: Multimodal diffusion models on arbitrary state spaces. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51924–51956. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/rojas25a.html.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8317–8326. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00851. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

Zhewen Wan, Tianchen Song, Chen Lin, Zhiyong Zhao, and Xianpeng Lang. Efficient token pruning for llada-v, 2026. URL https://arxiv.org/abs/2601.20168.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, Jinbin Bai, Qian Yu, Dengyang Jiang, Yuandong Pu, Haoxing Chen, Le Zhuo, Junjun He, Gen Luo, Tianbin Li, Ming Hu, Jin Ye, Shenglong Ye, Bo Zhang, Chang Xu, Wenhai Wang, Hongsheng Li, Guangtao Zhai, Tianfan Xue, Bin Fu, Xiaohong Liu, Yu Qiao, and Yihao Liu. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding, 2025. URL https://arxiv.org/abs/2510.06308.

Ruixuan Xu, Jiexi Xu, Qiyan Zhao, and Xiaofeng Zhang. Fixing semantic blind spots in anchor tokens of dMLLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 19359–19374, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.966. URL https://aclanthology.org/2026.findings-acl.966/.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/caa934a507a952698d54efb24845fc4b-Abstract-Conference.html.

Jiacheng Ye, Shansan Gong, Jiahui Gao, Junming Fan, Shuang Wu, Wei Bi, Haoli Bai, Lifeng Shang, and Lingpeng Kong. Dream-vl & dream-vla: Open vision-language and vision-language-action models with diffusion language model backbone. *CoRR*, abs/2512.22615, 2025. doi: 10.48550/ARXIV.2512.22615. URL https://doi.org/10.48550/arXiv.2512.22615.

Zebin You, Shen Nie, Xiaolu Zhang, JUN ZHOU, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. Llada-v: Large language diffusion models with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10093–10105, June 2026.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *CoRR*, abs/2505.16990, 2025. doi: 10.48550/ARXIV.2505.16990. URL https://doi.org/10.48550/arXiv.2505.16990.

Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 9556–9567. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00913. URL https://doi.org/10.1109/CVPR52733.2024.00913.

Lunbin Zeng, Jingfeng Yao, Bencheng Liao, Hongyuan Tao, Wenyu Liu, and Xinggang Wang. Diffusionvl: Translating any autoregressive models into diffusion vision language models, 2026. URL https://arxiv.org/abs/2512.15713.

Qicheng Zhao, Qi Sun, and Zheyu Yan. Seeing the end at step zero: Accelerating diffusion mllms via mlp sparsity-aware truncation, 2026a. URL https://arxiv.org/abs/2607.14557.

Qiyan Zhao, Xiaofeng Zhang, Shuochen Chang, Qianyu Chen, Xiaosong Yuan, Xuhang Chen, Luoqi Liu, Jiajun Zhang, Xu-Yao Zhang, and Da-Han Wang. Context tokens are anchors: Understanding the repeat curse in dMLLMs from an information flow perspective. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=mOz9jVYxsD.

Haomin Zuo, Yidi Li, Luoxiao Yang, and Xiaofeng Zhang. Diffusion-CAM: Faithful visual explanations for dMLLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12085–12101, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.553. URL https://aclanthology.org/2026.acl-long.553/.

APPENDIX OVERVIEW

The appendix complements the main text with implementation details and additional evidence. Appendix A covers the predictor, algorithm and models, Appendix B reports additional cross-model transfer results, Appendices C and D provide full attention and representation analyses, Appendix E gives the mathematical characterization of visual lifetimes, and Appendix F presents additional qualitative examples.

A IMPLEMENTATION

A.1 IMPLEMENTATION DETAILS OF THE STATE-GUIDED VISUAL EXIT PREDICTOR

Feature preprocessing and predictor. For Dimple, the final-layer hidden dimension is 3,584. We concatenate the pooled generation state $\bar{\mathbf{h}}_t$ and its cross-step change $\Delta\bar{\mathbf{h}}_t$, yielding a 7,168-dimensional state feature. The feature is standardized and projected onto 16 principal components using statistics fitted on the training split. We then append the normalized generation progress t/T, resulting in a 17-dimensional predictor input. As illustrated in Figure 10, the predictor is a lightweight two-layer MLP with a 16-unit hidden layer, $\tanh$ activation, and a scalar output, totaling 305 trainable parameters (Kingma & Ba, 2015; Loshchilov & Hutter, 2019; Le Cun et al., 1998).

Training. Training trajectories are collected from MME (Fu et al., 2025) and TextVQA (Singh et al., 2019), comprising 7,374 questions in total, using 16 denoising steps and 32 generation slots. The supervision target is the maximum future visual influence defined in Eq 7. We use the transformed target $\ell_t = \log_{10}(y_t + \epsilon)$ with $\epsilon = 10^{-6}$ and standardize it using training-set statistics. The predictor is optimized with mean squared error and AdamW for 1,000 epochs, using a learning rate of 3×10^{-3} and weight decay of 10^{-2}. Figure 11 shows the training loss over optimization, and the checkpoint with the lowest validation loss is used for evaluation.

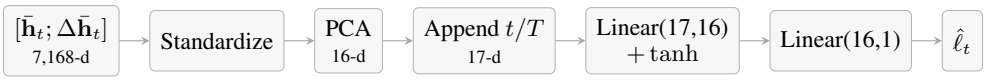

Figure 10: Pipeline of the state-guided visual exit predictor.

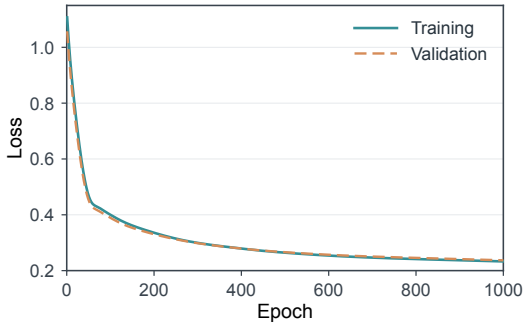

Figure 11: Training loss of the visual-exit predictor.

Visual exit. The predicted value is mapped back to the original JS scale before thresholding. We use $\delta = 0.05$ throughout evaluation. Once the predicted maximum future visual influence satisfies $\hat{y}_t \leq \delta$, visual KV are permanently removed from the next denoising step onward. Inference only reads hidden states from the normal forward pass and therefore does not require intervention-based JS computation online.

Table 4 summarizes the predictor and training configuration.

Offline cost. Table 5 reports the offline cost of collecting training signals and fitting the predictor. Collecting state representations and intervention-based JS labels takes 1 h 45 min 33 s over the 7,374 training questions, while training the predictor for 1,000 epochs takes only 9.85 s. Most of the offline cost therefore comes from constructing the supervision trajectories.

A.2 Algorithm Overview

Algorithm 1 summarizes the complete state-guided visual exit procedure. The intervention-based visual influence J_t is used only to construct offline supervision, while online inference requires only the hidden states produced by the normal denoising process and the lightweight state readout.

A.3 Model scale

We evaluate four DVLMs with comparable parameter scales. DreamVL (Ye et al., 2025) and Dimple (Yu et al., 2025) use 7B backbones, while MMaDA-Base (Yang et al., 2025) is an 8B model. LaViDa (Li et al., 2025) is released with both Dream-7B and LLaDA-8B backbones; throughout this work, we use the Dream-7B variant, denoted as LaViDa.

B Further Analysis of Cross-Model Transfer

The main text shows that a predictor trained on Dimple can be applied directly to another model without retraining. Here, we further examine this cross-model behavior from two directions.

B.1 Reverse Transfer from DreamVL to Dimple

We first reverse the training direction and train the same predictor on DreamVL trajectories. The resulting predictor is then directly applied to Dimple with all parameters and preprocessing kept fixed. As shown in Table 6, the predictor continues to identify useful visual-exit points under both generation settings while largely preserving task performance.

Table 4: Implementation details of the state-guided visual exit predictor.

Item	Setting
Training data	MME + TextVQA, 7,374 questions
Generation setting	16 steps / 32 tokens
Hidden dimension	3,584
State feature dimension	7,168
PCA components	16
Predictor input dimension	17
MLP hidden dimension	16
Activation	$\tanh$
Trainable parameters	305
Training epochs	1,000
Optimizer	AdamW
Learning rate	3×10^{-3}
Weight decay	10^{-2}
Loss	Mean squared error
Target transform	$\log_{10}(y_t + 10^{-6})$
Visual-exit threshold	$\delta = 0.05$

Table 5: Offline cost of collecting state representations and intervention-based JS labels and training the predictor on Dimple. Measurements use 7,374 questions under the 16-step / 32-token setting.

Stage	Time
State + JS collection	1 h 45 min 33 s
Predictor training (1,000 epochs)	9.85 s

Together with the Dimple-to-DreamVL transfer in the main text, this result shows that the transfer is not tied to a particular source–target direction. Generation states from both models contain a similar signal about whether later steps will still need visual computation, and this signal can be read by a predictor learned from another model.

B.2 Transfer to Additional Models

We next apply the original Dimple-trained predictor to additional DVLMs without retraining. Tables 7 and 8 report the complete results on LaViDa and MMaDA, respectively. Across the evaluated tasks, the transferred predictor continues to produce meaningful visual exits while maintaining the overall task performance.

These results extend the observation beyond the two models studied in the main analysis. Although the amount of removable visual computation varies substantially across models and tasks, the generation state consistently provides a usable signal for deciding when visual computation can end. This supports the view that the state signature captured by the predictor reflects a shared property of diffusion multimodal generation rather than a model-specific exit pattern.

C Full Attention Analysis

We provide the full attention patterns for Dimple and DreamVL in Figures 12 and 13, complementing the attention allocation analysis in the main text.

[1]MMaDA uses a 4,096-dimensional hidden state, while the predictor is built for Dimple's 3,584-dimensional representation. We therefore use a fixed seed of 42 to sample 3,584 channels without replacement and apply the same sorted channel indices to both h and Δh for all samples. The generation seed is set separately to 0.

Algorithm 1 State-Guided Visual Exit

Require: Training trajectories $\mathcal{D}_{\text{train}}$, denoising steps T, exit threshold δ
Ensure: Predictor f_θ and adaptive visual-exit rule

Offline: learn the state readout
for each unpruned trajectory in $\mathcal{D}_{\text{train}}$ **do**
  for $t = 1, \dots, T-1$ **do**
    Obtain unresolved positions U_{t+1}
    $\bar{\mathbf{h}}_t \leftarrow \frac{1}{|U_{t+1}|} \sum_{i \in U_{t+1}} \mathbf{h}_{t,i}$
    if $t = 1$ **then**
      $\Delta\bar{\mathbf{h}}_t \leftarrow \mathbf{0}$
    else
      $\Delta\bar{\mathbf{h}}_t \leftarrow \frac{1}{|U_{t+1}|} \sum_{i \in U_{t+1}} (\mathbf{h}_{t,i} - \mathbf{h}_{t-1,i})$
    end if
    $\mathbf{x}_t \leftarrow [\bar{\mathbf{h}}_t \| \Delta\bar{\mathbf{h}}_t]$
    $\mathbf{u}_t \leftarrow (\mathbf{x}_t, t/T)$
    $y_t \leftarrow \max_{t < s \le T} J_s$
    $\ell_t \leftarrow \log_{10}(y_t + \epsilon)$
    Store $(\mathbf{u}_t, \ell_t)$
  end for
end for
Fit preprocessing on the training features and optimize
$\theta \leftarrow \arg\min_\theta \mathbb{E}\left[\left(f_\theta(\mathbf{u}_t) - \ell_t\right)^2\right]$

Online: adaptive visual exit
for $t = 1, \dots, T-1$ **do**
  Run the normal denoising step with visual KV
  Obtain unresolved positions U_{t+1}
  Compute $\bar{\mathbf{h}}_t$
  if $t = 1$ **then**
    $\Delta\bar{\mathbf{h}}_t \leftarrow \mathbf{0}$
  else
    Compute $\Delta\bar{\mathbf{h}}_t$
  end if
  $\mathbf{x}_t \leftarrow [\bar{\mathbf{h}}_t \| \Delta\bar{\mathbf{h}}_t]$
  $\mathbf{u}_t \leftarrow (\mathbf{x}_t, t/T)$
  $\hat{\ell}_t \leftarrow f_\theta(\mathbf{u}_t)$
  $\hat{y}_t \leftarrow 10^{\hat{\ell}_t} - \epsilon$
  if $\hat{y}_t \le \delta$ **then**
    Remove visual KV permanently from step $t+1$
    Continue denoising from steps $t+1, \dots, T$ without visual KV
    return
  end if
end for
Retain visual KV through the final denoising step

D Full Evolution of Generation Representations

We provide the complete 16-step representation trajectories as a supplement to the analysis in Section 3.1. While the main text visualizes selected denoising steps for clarity, Figures 14, 15, and 16 show the evolution at every step for Dimple, DreamVL, and LaViDa, respectively. All steps use the same fixed PCA projection within each model, allowing the movement of generation representations to be compared directly over denoising. The denser trajectories make the gradual change in

Table 6: Results with the visual-exit predictor trained on DreamVL under two generation settings. Retention is computed as $\mathrm{Ours}/\mathrm{Base} \times 100\%$. Average exit step reports the mean denoising step at which visual computation terminates. Visual exit rate denotes the fraction of samples for which visual KV are removed before generation ends. Decode attention saving excludes the first denoising step. Visual KV reduction measures the average decrease in visual KV participation. Avg. denotes the macro average over the reported benchmarks.

Metric	SQA	AI2D	MMStar	POPE	MMB	MMB-CN	MMMU	Avg.
Dimple *8 Steps / 16 Tokens*								
Base	78.53	74.19	47.00	84.06	72.45	70.67	44.56	67.35
Ours	78.33	74.58	46.20	83.44	71.98	70.51	44.67	67.10
Retention	99.75%	**100.53%**	98.30%	99.26%	99.35%	99.77%	**100.25%**	99.60%
Avg. Exit Step	3.192	3.215	2.894	2.000	3.088	3.192	3.441	3.003
Visual Exit Rate	99.95%	100.00%	100.00%	100.00%	100.00%	100.00%	98.22%	99.74%
Decode Attn. Saving	39.05%	50.13%	47.01%	71.67%	47.07%	45.69%	38.44%	48.44%
Visual KV Reduction	72.08%	72.23%	76.13%	87.50%	73.72%	72.42%	68.83%	74.70%
Dimple *16 Steps / 32 Tokens*								
Base	78.68	74.03	47.13	84.29	72.29	70.20	44.56	67.31
Ours	78.33	74.22	47.13	81.03	71.98	70.28	44.44	66.77
Retention	99.56%	**100.26%**	**100.00%**	96.13%	99.57%	**100.11%**	99.73%	99.34%
Avg. Exit Step	4.336	5.318	3.776	2.012	3.639	4.002	4.982	4.009
Visual Exit Rate	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.78%	99.83%
Decode Attn. Saving	36.63%	43.83%	44.43%	65.80%	46.25%	44.37%	36.78%	45.44%
Visual KV Reduction	77.77%	72.90%	82.74%	93.67%	83.55%	81.05%	74.89%	80.94%
DreamVL *8 Steps / 16 Tokens*								
Base	92.12	79.47	57.07	85.77	81.97	48.61	50.11	70.73
Ours	91.87	79.44	56.53	83.92	81.11	53.64	49.56	70.87
Retention	99.73%	99.96%	99.05%	97.84%	98.95%	**110.35%**	98.90%	**100.68%**
Avg. Exit Step	2.026	2.002	2.010	2.000	2.014	2.302	2.151	2.072
Visual Exit Rate	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	95.78%	99.40%
Decode Attn. Saving	65.20%	85.61%	75.25%	87.75%	71.93%	68.54%	78.53%	76.12%
Visual KV Reduction	87.22%	87.49%	87.39%	87.50%	87.32%	83.85%	83.93%	86.39%
DreamVL *16 Steps / 32 Tokens*								
Base	92.02	79.18	57.00	85.78	80.88	42.34	50.56	69.68
Ours	91.32	79.24	57.13	81.42	79.88	48.68	49.56	69.60
Retention	99.24%	**100.08%**	**100.23%**	94.92%	98.76%	**114.97%**	98.02%	**100.89%**
Avg. Exit Step	2.001	2.001	2.010	2.000	2.017	3.491	2.366	2.269
Visual Exit Rate	100.00%	100.00%	100.00%	100.00%	100.00%	99.90%	97.22%	99.59%
Decode Attn. Saving	62.35%	83.20%	72.43%	84.39%	68.65%	61.47%	78.02%	72.93%
Visual KV Reduction	93.75%	93.74%	93.73%	93.75%	93.69%	84.52%	91.32%	92.07%

Table 7: Complete results of state-guided visual exit on LaViDa.

Metric	SQA	MMMU	MMStar	MME	AI2D	MMB	MMB-CN	POPE	Avg.
LaViDa									
Base	75.16	43.00	43.73	1833.13	68.78	72.83	70.05	86.11	–
Ours	75.11	43.11	43.73	1838.59	68.75	72.52	70.28	84.81	–
Retention	99.93%	**100.26%**	**100.00%**	**100.30%**	99.96%	99.57%	**100.33%**	98.49%	99.85%
Decode Attn. Saving	15.78%	3.28%	7.07%	67.69%	14.95%	33.87%	15.13%	84.29%	30.26%
Visual KV Reduction	19.30%	3.89%	8.25%	72.23%	16.65%	39.40%	17.66%	89.19%	33.32%

visual–generation geometry visible throughout the full generation process and provide an additional view of the representation evolution discussed in the main text.

Table 8: Complete results of state-guided visual exit on MMaDA.

Metric	SQA	AI2D	MMStar	MME	MMB	MMB-CN	POPE	Avg.
				MMaDA[1]				
Base	52.75	58.26	31.80	1011.55	27.01	20.20	72.21	–
Ours	52.85	58.39	31.80	1029.53	27.01	20.12	72.20	–
Retention	**100.19%**	**100.22%**	**100.00%**	**101.78%**	**100.00%**	99.60%	99.99%	**100.25%**
Decode Attn. Saving	18.25%	19.24%	8.86%	52.77%	11.20%	9.97%	10.82%	18.73%
Visual KV Reduction	17.69%	18.07%	8.48%	48.35%	10.64%	9.44%	9.88%	17.51%

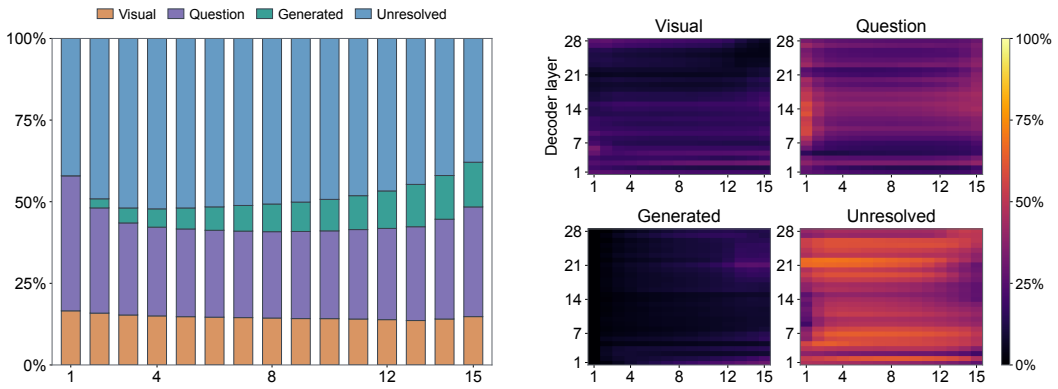

Figure 12: Full attention patterns from unresolved generation positions in Dimple across denoising steps and decoder layers. Visual attention remains broadly persistent throughout generation despite the changing visual influence measured by J_t.

E Mathematical Characterization of Visual Lifetimes

E.1 Interventional Visual Influence

For an unresolved position $i \in U_t$, let $P_{t,i}$ and $Q_{t,i}$ denote the predictive distributions with and without direct visual computation, respectively. Define

$$j_{t,i} = \mathrm{JS}(P_{t,i}, Q_{t,i}), \qquad J_t = \frac{1}{|U_t|} \sum_{i \in U_t} j_{t,i}. \tag{11}$$

Throughout this section, KL and JS divergences use natural logarithms. Let the corresponding total variation distance be

$$d_{t,i} = \mathrm{TV}(P_{t,i}, Q_{t,i}) = \frac{1}{2} \sum_{v \in \mathcal{V}} \left| P_{t,i}(v) - Q_{t,i}(v) \right|. \tag{12}$$

Proposition 1. The mean intervention-induced change in the unresolved predictive distributions satisfies

$$\frac{1}{|U_t|} \sum_{i \in U_t} d_{t,i} \le \sqrt{2 J_t}. \tag{13}$$

Proof. Let $M_{t,i} = (P_{t,i} + Q_{t,i})/2$. By Pinsker's inequality,

$$\mathrm{KL}(P_{t,i} \| M_{t,i}) \ge 2\,\mathrm{TV}(P_{t,i}, M_{t,i})^2, \qquad \mathrm{KL}(Q_{t,i} \| M_{t,i}) \ge 2\,\mathrm{TV}(Q_{t,i}, M_{t,i})^2. \tag{14}$$

Since

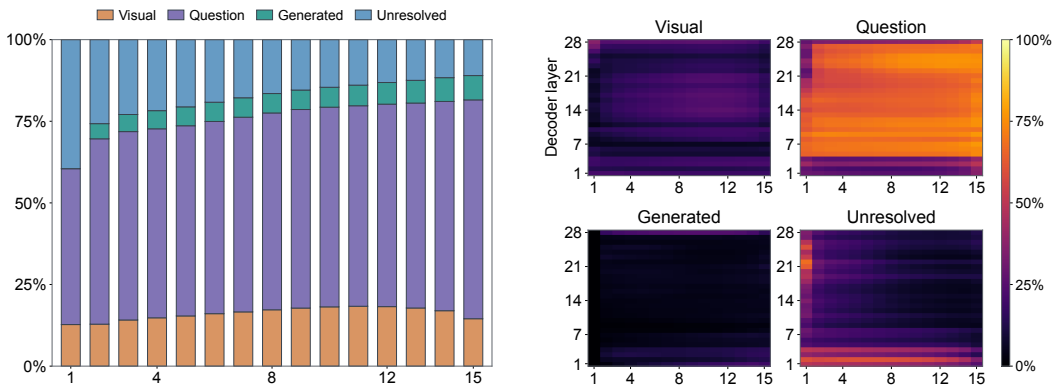

Figure 13: Full attention patterns from unresolved generation positions in DreamVL across denoising steps and decoder layers. The attention trace shows no clear transition corresponding to the changing visual influence measured by J_t.

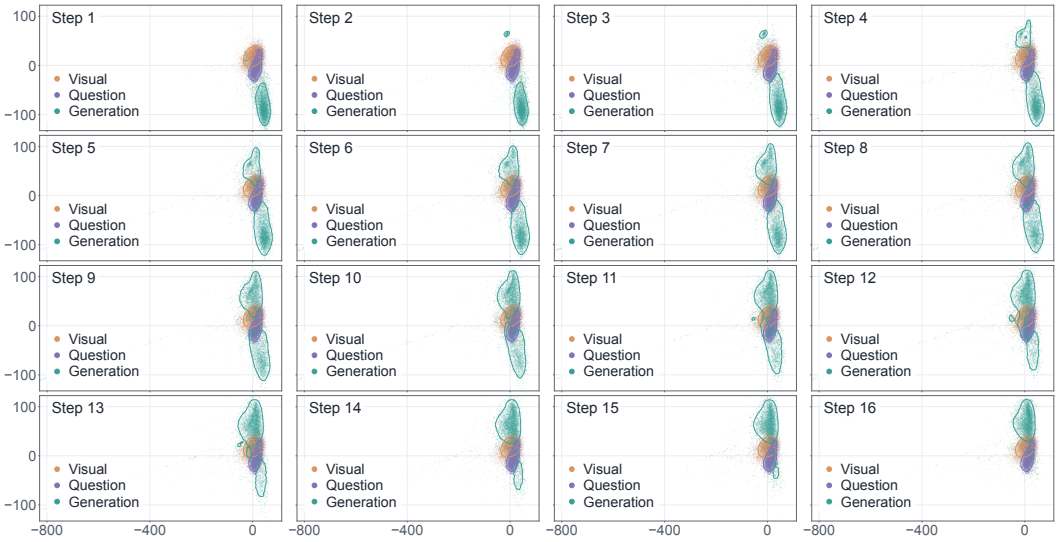

Figure 14: Full 16-step evolution of generation representations in Dimple. A single PCA projection is shared across all denoising steps. Visual representations are fixed at the first step, while generation representations evolve throughout the trajectory.

$$\mathrm{TV}(P_{t,i}, M_{t,i}) = \mathrm{TV}(Q_{t,i}, M_{t,i}) = \frac{1}{2} d_{t,i}, \tag{15}$$

we obtain

$$j_{t,i} \geq \frac{1}{2} d_{t,i}^2, \qquad d_{t,i} \leq \sqrt{2 j_{t,i}}. \tag{16}$$

Averaging over U_t and applying Jensen's inequality gives

$$\frac{1}{|U_t|} \sum_{i \in U_t} d_{t,i} \leq \frac{1}{|U_t|} \sum_{i \in U_t} \sqrt{2 j_{t,i}} \leq \sqrt{\frac{2}{|U_t|} \sum_{i \in U_t} j_{t,i}} = \sqrt{2 J_t}. \tag{17}$$

□

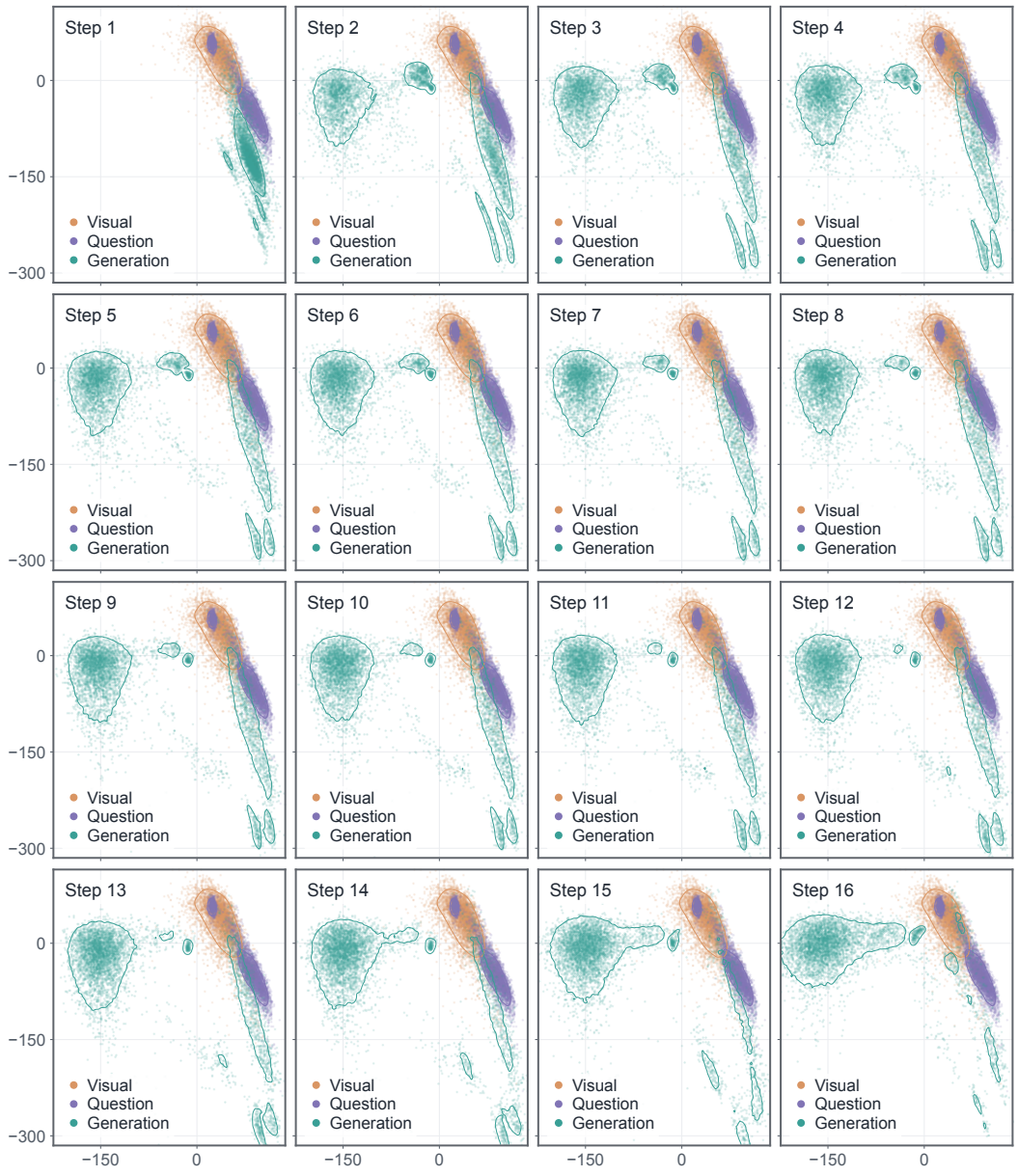

Figure 15: Full 16-step evolution of generation representations in DreamVL. The same fixed PCA coordinates are used across steps, exposing the continuous change in the geometry between visual and generation representations.

E.2 Future Influence and Visual Lifetime

For $t < T$, define the maximum future visual influence

$$y_t = \max_{t<s\leq T} J_s. \tag{18}$$

Proposition 2. For any threshold $\delta > 0$,

$$y_t \leq \delta \quad \Longleftrightarrow \quad J_s \leq \delta,\ \forall s \in \{t+1, \ldots, T\}. \tag{19}$$

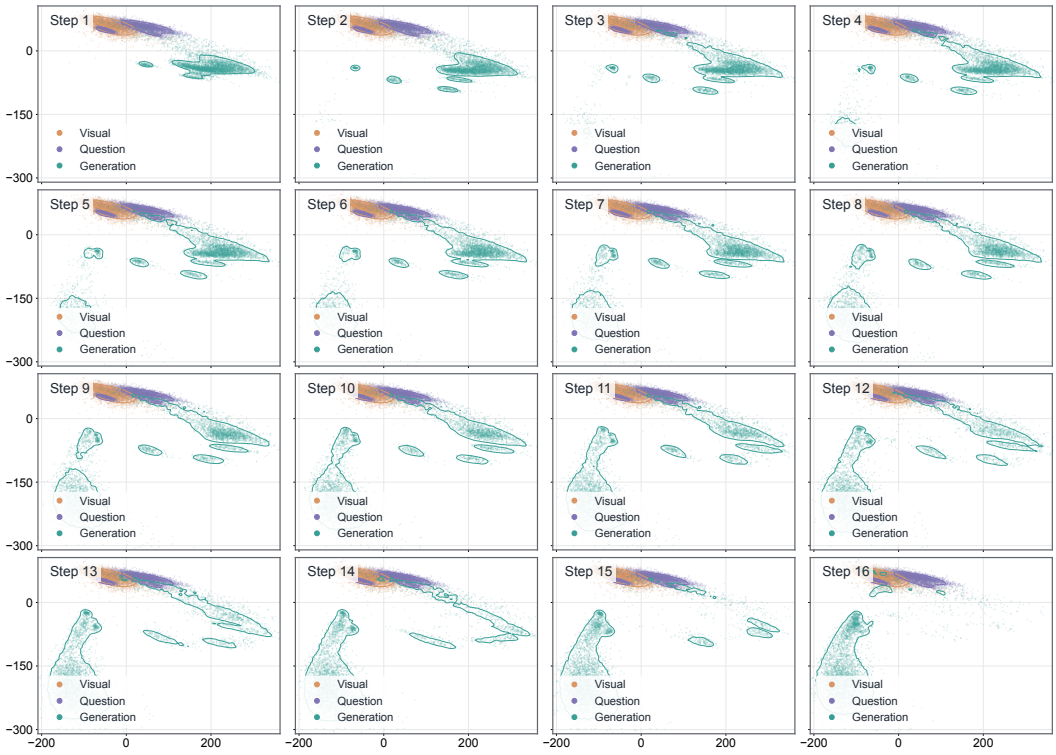

Figure 16: Full 16-step evolution of generation representations in LaViDa. The fixed projection provides a step-by-step view of how the generation state changes relative to the visual representations during denoising.

Moreover,

$$y_{t+1} \le y_t, \qquad t = 1, \dots, T-2. \tag{20}$$

Proof. Equation 19 follows directly from the definition of the maximum. Since $\{s : s > t+1\} \subseteq \{s : s > t\}$,

$$y_{t+1} = \max_{s>t+1} J_s \le \max_{s>t} J_s = y_t. \tag{21}$$

□

Thus,

$$y_1 \ge y_2 \ge \cdots \ge y_{T-1}, \tag{22}$$

even when $J_1, \dots, J_T$ is non-monotonic.

Combining Propositions 1 and 2 yields

$$y_t \le \delta \quad \Longrightarrow \quad \frac{1}{|U_s|} \sum_{i \in U_s} \mathrm{TV}(P_{s,i}, Q_{s,i}) \le \sqrt{2\delta}, \qquad \forall s > t. \tag{23}$$

Define the oracle visual-exit step as

$$\tau_\delta^* = 1 + \min\left(\{t \in \{1, \dots, T-1\} : y_t \le \delta\} \cup \{T\}\right). \tag{24}$$

Here $\tau_\delta^* \in \{2, \dots, T+1\}$, where $\tau_\delta^* = T+1$ denotes retaining visual computation throughout all T denoising steps. The corresponding visual lifetime is

$$L_{\mathrm{see}}(\delta) = \tau_\delta^* - 1. \tag{25}$$

Whenever $\tau_\delta^* \le T$,

$$J_s \le \delta, \qquad \forall s \in \{\tau_\delta^*, \dots, T\}. \tag{26}$$

E.3 Generation State and Future Visual Influence

Let $\mathbf{S}_t$ denote the generation state after denoising step t, and define the full conditional state

$$\mathbf{Z}_t = (\mathbf{S}_t, \mathbf{V}), \tag{27}$$

where $\mathbf{V}$ denotes the fixed visual condition on the unpruned trajectory. The denoising dynamics can be written as

$$\mathbf{Z}_{t+1} = F_t(\mathbf{Z}_t, \boldsymbol{\xi}_t), \tag{28}$$

with $\boldsymbol{\xi}_t$ denoting the stochastic variables used at step t. The future trajectory is

$$\Gamma_t = (\mathbf{Z}_{t+1}, \mathbf{Z}_{t+2}, \dots, \mathbf{Z}_T). \tag{29}$$

The future visual influence is a trajectory functional

$$y_t = G_t(\Gamma_t). \tag{30}$$

For deterministic denoising, the future trajectory is determined by the current conditional state, and therefore

$$y_t = g_t(\mathbf{Z}_t). \tag{31}$$

The readout observes the generation-state feature

$$\mathbf{x}_t = \phi(\mathbf{S}_t) = \left[\bar{\mathbf{h}}_t \| \Delta\bar{\mathbf{h}}_t\right]. \tag{32}$$

The regression target is the logarithmic future visual influence

$$\ell_t = \log_{10}(y_t + \epsilon). \tag{33}$$

Appending normalized denoising progress gives the predictor input

$$\mathbf{u}_t = (\mathbf{x}_t, t/T). \tag{34}$$

For squared-error regression,

$$\mathcal{L}(f) = \mathbb{E}\left[\left(f(\mathbf{u}_t) - \ell_t\right)^2\right]. \tag{35}$$

Proposition 3. The minimum-risk readout is

$$f^*(\mathbf{u}_t) = \mathbb{E}\left[\ell_t \mid \mathbf{u}_t\right]. \tag{36}$$

Its minimum achievable squared error is

$$\mathcal{L}^* = \mathbb{E}\left[\mathrm{Var}\left(\ell_t \mid \mathbf{u}_t\right)\right]. \tag{37}$$

Proof. Let $\mu_t = \mathbb{E}[\ell_t \mid \mathbf{u}_t]$. Then

$$\mathbb{E}\left[(\ell_t - f(\mathbf{u}_t))^2 \mid \mathbf{u}_t\right] = \mathrm{Var}(\ell_t \mid \mathbf{u}_t) + \left(f(\mathbf{u}_t) - \mu_t\right)^2. \tag{38}$$

The second term is minimized at $f(\mathbf{u}_t) = \mu_t$. Taking expectation over $\mathbf{u}_t$ gives Eq 37. □

For two states a and b at the same denoising step and matched current visual influence,

$$t^{(a)} = t^{(b)}, \qquad J_t^{(a)} \approx J_t^{(b)}, \tag{39}$$

the analysis in Section 3.2 examines the empirical relation

$$\left\|\mathbf{x}_t^{(a)} - \mathbf{x}_t^{(b)}\right\|_2 \downarrow \qquad \text{corresponds empirically to} \qquad \left|\ell_t^{(a)} - \ell_t^{(b)}\right| \downarrow. \tag{40}$$

E.4 First-Order Dynamics of the Generation State

Let the normalized denoising time be

$$r_t = \frac{t}{T}, \qquad \Delta r = \frac{1}{T}. \tag{41}$$

For a local first-order interpretation, assume that the pooled generation state follows a locally smooth trajectory $\bar{\mathbf{h}}(r)$. The backward finite difference satisfies

$$\Delta\bar{\mathbf{h}}_t = \bar{\mathbf{h}}(r_t) - \bar{\mathbf{h}}(r_t - \Delta r) = \Delta r\, \dot{\bar{\mathbf{h}}}(r_t) + \mathcal{O}(\Delta r^2). \tag{42}$$

Equivalently,

$$\dot{\bar{\mathbf{h}}}(r_t) = \frac{\Delta\bar{\mathbf{h}}_t}{\Delta r} + \mathcal{O}(\Delta r). \tag{43}$$

Suppose that future visual influence can locally be represented by a smooth function $\psi(\mathbf{h}, r)$. For a small forward increment $\eta > 0$,

$$\psi(\bar{\mathbf{h}}(r_t + \eta), r_t + \eta) = \psi(\bar{\mathbf{h}}_t, r_t) + \eta\, \nabla_{\mathbf{h}}\psi(\bar{\mathbf{h}}_t, r_t)^\top \dot{\bar{\mathbf{h}}}(r_t) + \eta\, \frac{\partial\psi}{\partial r}(\bar{\mathbf{h}}_t, r_t) + \mathcal{O}(\eta^2). \tag{44}$$

Substituting Eq. 43 gives

$$\psi(\bar{\mathbf{h}}(r_t + \eta), r_t + \eta) = \psi(\bar{\mathbf{h}}_t, r_t) + \frac{\eta}{\Delta r}\nabla_{\mathbf{h}}\psi(\bar{\mathbf{h}}_t, r_t)^\top \Delta\bar{\mathbf{h}}_t + \eta\, \frac{\partial\psi}{\partial r}(\bar{\mathbf{h}}_t, r_t) + \mathcal{O}(\eta\Delta r + \eta^2). \tag{45}$$

Thus, to first order, the current pooled state $\bar{\mathbf{h}}_t$, its cross-step change $\Delta\bar{\mathbf{h}}_t$, and normalized progress $r_t = t/T$ jointly characterize the local evolution relevant to future visual influence. This motivates the predictor input $\mathbf{u}_t = (\mathbf{x}_t, t/T)$ defined in Eq. 34.

E.5 Readout Error and Exit Decision

Let

$$\hat{\ell}_t = f_\theta(\mathbf{u}_t), \qquad \hat{y}_t = 10^{\hat{\ell}_t} - \epsilon. \tag{46}$$

Assume a bounded log-space readout error

$$\left|\hat{\ell}_t - \ell_t\right| \le \rho. \tag{47}$$

Since $y_t + \epsilon = 10^{\ell_t}$ and $\hat{y}_t + \epsilon = 10^{\hat{\ell}_t}$,

$$10^{-\rho}\left(\hat{y}_t + \epsilon\right) \le y_t + \epsilon \le 10^{\rho}\left(\hat{y}_t + \epsilon\right). \tag{48}$$

Hence,

$$10^{\rho}\left(\hat{y}_t + \epsilon\right) - \epsilon \le \delta \quad \Longrightarrow \quad y_t \le \delta. \tag{49}$$

Combining with Proposition 2 gives

$$10^{\rho}\left(\hat{y}_t + \epsilon\right) - \epsilon \le \delta \quad \Longrightarrow \quad J_s \le \delta, \qquad \forall s > t. \tag{50}$$

Similarly,

$$10^{-\rho}\left(\hat{y}_t + \epsilon\right) - \epsilon > \delta \quad \Longrightarrow \quad y_t > \delta, \tag{51}$$

which implies

$$\exists\, s > t \quad \text{s.t.} \quad J_s > \delta. \tag{52}$$

If $\hat{y}_t = y_t$ for all t, define the predicted visual-exit step as

$$\hat{\tau}_\delta = 1 + \min\left(\{t \in \{1, \ldots, T-1\} : \hat{y}_t \le \delta\} \cup \{T\}\right). \tag{53}$$

Then

$$\boxed{\hat{\tau}_\delta = \tau^*_\delta}. \tag{54}$$

F Qualitative analysis

Figure 17 provides additional examples of state-guided visual exit across different types of visual questions, including recognition, counting, description, and reasoning. The examples show that visually grounded answers can often remain stable even when visual KV are removed for a substantial portion of the denoising trajectory. At the same time, the model does not always terminate visual computation early; some examples retain visual KV throughout generation. These cases qualitatively illustrate that the predicted exit point adapts to the evolving generation state rather than following a fixed schedule.



User: *How many coffee mugs are in the set?*

Ours: 3.

Visual KV *11/16 steps removed*



User: *What brand is featured in this advertisement?*

Ours: Subway.

Visual KV *7/16 steps removed*



User: *Describe this photo in detail.*

Ours: The image showcases the iconic Space Needle, a prominent landmark tower located in Seattle, Washington. The Space Needle is a tall observation tower with a distinctive design.

Visual KV *8/16 steps removed*



User: *Explain all the cues that indicate the current traffic conditions.*

Ours: The image shows conditions that heavy traffic is present. The presence of multiple vehicles in close proximity to each other further supports the notion of heavy traffic congestion.

Visual KV *11/16 steps removed*



User: *What are the animals in the painting and what are they doing?*

Ours: The painting features a bear, a cat, and a rabbit. They are gathered around a small table, enjoying a tea party with food and cookies.

Visual KV *retained throughout*

Visual KV retained Visual KV removed

Figure 17: **Additional qualitative examples of state-guided visual exit.** The visual-KV strip shows when direct visual computation is retained and when it is removed over the 16-step denoising trajectory.