Localizing Knowledge in Diffusion Transformers

Arman Zarei[1]*, **Samyadeep Basu**[3], **Keivan Rezaei**[1], **Zihao Lin**[2], **Sayan Nag**[3], **Soheil Feizi**[1]
[1]University of Maryland [2]University of California, Davis [3]Adobe

Abstract

Understanding how knowledge is distributed across the layers of generative models is crucial for improving interpretability, controllability, and adaptation. While prior work has explored knowledge localization in UNet-based architectures, Diffusion Transformer (DiT)-based models remain underexplored in this context. In this paper, we propose a model- and knowledge-agnostic method to localize where specific types of knowledge are encoded within the DiT blocks. We evaluate our method on state-of-the-art DiT-based models, including PixArt-α, FLUX, and SANA, across six diverse knowledge categories. We show that the identified blocks are both interpretable and causally linked to the expression of knowledge in generated outputs. Building on these insights, we apply our localization framework to two key applications: *model personalization* and *knowledge unlearning*. In both settings, our localized fine-tuning approach enables efficient and targeted updates, reducing computational cost, improving task-specific performance, and better preserving general model behavior with minimal interference to unrelated or surrounding content. Overall, our findings offer new insights into the internal structure of DiTs and introduce a practical pathway for more interpretable, efficient, and controllable model editing. [1]

1 Introduction

Diffusion and flow models [14, 26, 22, 13, 17, 8, 19] have rapidly become the leading paradigm for a wide range of generative tasks, particularly text-to-image (T2I) synthesis. With access to such powerful pretrained models, it is crucial to explore their potential for applications beyond mere generation. A growing body of work [11, 32, 16, 37] has focused on localizing different types of knowledge and capabilities within these models, enabling more targeted usage. For example, [11] showed that cross-attention layers are key to incorporating prompt compositional information, while [32, 20] demonstrated that structural information is often concentrated in the self-attention modules of UNet-based architectures. These insights have been applied to tasks such as image editing and structure-preserving generation [11, 4, 20].

Localizing where specific knowledge resides within models is essential for interpretability, targeted interventions, and understanding model behavior. In generative models, it plays a critical role in applications such as model unlearning and personalization. Several works [16, 29, 9, 36] have shown that generative models often memorize unsafe or unwanted content (e.g., copyrighted or NSFW content) and proposed methods to selectively erase such concepts. In model personalization, the goal is to generate novel renditions of a subject using only a few reference photos across diverse scenes and poses. In both cases, localizing knowledge within the model is crucial for enabling *targeted interventions* that make fine-tuning more efficient and effective, while better preserving the model's prior capabilities and overall generation quality.

*Correspondence to: azarei@umd.edu
[1]Project page is available at: https://armanzarei.github.io/Localizing-Knowledge-in-DiTs

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Figure 1: **Localization across various DiT models and knowledge categories.** For each model, heatmaps indicate the frequency of each block being selected as a dominant carrier of different target knowledge. Green-bordered images are standard generations, while red-bordered images result from withholding knowledge-specific information in the localized blocks. Our method successfully localizes diverse knowledge types, with variation in localization patterns across models.

Recent advances in text-to-image generation have marked a notable evolution, transitioning from UNet-based architectures [27] to Transformer-based models [33], particularly the Diffusion Transformer (DiT) [23]. DiT architectures, with their purely attention-based structure, have achieved state-of-the-art generation quality compared to UNet counterparts. While extensive research has explored interpretability and localization in UNet-based architectures, DiT-based models have received comparatively little attention in this regard despite their recent emergence and strong performance.

In this paper, we thoroughly investigate the localization of different types of knowledge within the blocks of diffusion transformers across a range of state-of-the-art models, including FLUX [17], SANA [35], and PixArt-α [5]. We introduce a model-agnostic and knowledge-agnostic method that provides a strong and reliable signal for identifying the blocks most responsible for generating specific types of knowledge. Our approach demonstrates strong performance and robustness across all evaluated models and a diverse set of knowledge categories, such as copyrighted content, NSFW material, and artistic styles (Figure 1). While the distribution of localized blocks varies from model to model, our method consistently identifies the key regions responsible for encoding each knowledge.

Building upon our knowledge localization technique, we further propose practical applications of our method for model personalization and unlearning in DiTs (Figure 2). Whether the objective is to inject new knowledge or remove undesired content, our approach first localizes the relevant information within the blocks of the DiT and then enables targeted interventions to modify it. Through extensive experiments, we demonstrate that our method outperforms baseline approaches that modify all blocks, achieving superior preservation of generation quality and consistency on unrelated and surrounding prompts. Notably, in the personalization setting, our method achieves improved task-specific performance compared to full-model fine-tuning, while also minimizing interference with surrounding knowledge. Additionally, our method is more efficient, offering faster training and lower memory usage compared to these baselines.

In summary, our contributions are: (1) We are the first to explore knowledge localization in DiTs by introducing a large-scale probing dataset covering diverse categories, and by proposing an automatic, model- and knowledge-agnostic method for identifying where such information resides within the model's blocks. (2) We conduct extensive evaluations across multiple DiT architectures and diverse knowledge types to validate the generality and robustness of our approach. (3) Building on this localization, we demonstrate practical applications for efficient model personalization and unlearning.

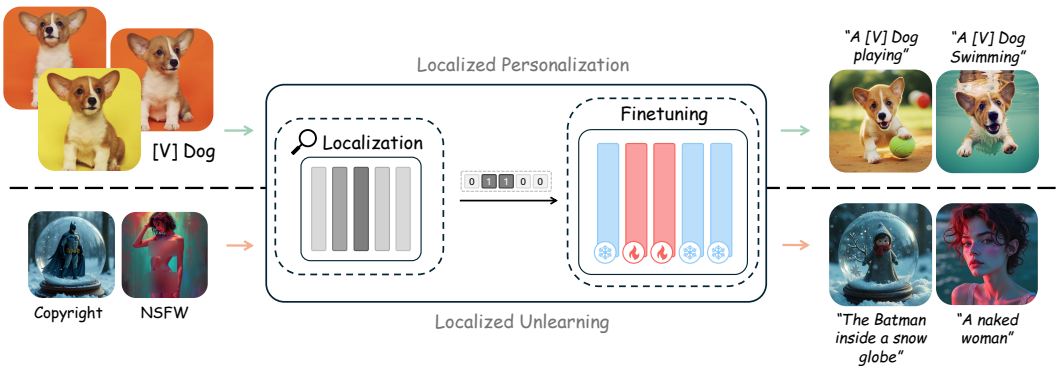

Figure 2: **Targeted fine-tuning via knowledge localization.** Given a concept to personalize or remove, our method first identifies the most relevant blocks via knowledge localization and restricts fine-tuning to those blocks. This enables efficient adaptation (top) and targeted suppression (bottom) with minimal impact on surrounding content, while better preserving the model's prior performance.

Our method enables targeted fine-tuning that is faster and more memory-efficient, while also achieving superior preservation of generation quality and consistency on unrelated or surrounding prompts.

2 Related Works

2.1 Diffusion and Flow Models

Diffusion models are a class of generative models based on stochastic differential equations (SDEs), where noise is progressively added to data through a stochastic forward process, ultimately transforming the data distribution into a standard Gaussian. A corresponding reverse process is then learned to reconstruct the original data from noise. Flow matching methods, closely related to diffusion models, instead define a deterministic mapping from noise to data using an ordinary differential equation (ODE). These methods learn a time-dependent vector field $v_\theta(x, t)$ that is trained to approximate a target field $v_t(x)$, which directs samples from the noise distribution toward the data distribution, by minimizing a flow matching loss. See Appendix A.1 for further details.

2.2 Interpretability of Diffusion Transformers

The internal mechanisms of text-to-image diffusion models have been primarily explored in the context of UNet-based models [3, 2, 37]. These studies reveal that knowledge of various visual concepts—such as artistic style—is either localized or distributed across a small subset of layers within the UNet architecture. Beyond offering interpretability, these localization insights have been leveraged to address practical challenges, including the removal of copyrighted content, without the need for full model retraining. With the recent shift toward transformer-based models such as Flux [17] and PixArt-α [5], understanding how and where concepts are encoded in these new models has become an emerging area of interest. Recent work has begun to uncover the interpretability of diffusion transformers. For instance, [10] show that attention maps in models like Flux can act as high-quality saliency maps, while [1] identify a subset of critical layers that are particularly effective for downstream tasks such as image editing. However, it remains unclear how knowledge of visual concepts—such as copyrighted objects, artistic styles, or safety-related content—is represented and localized within diffusion transformers. Gaining such insights could enable targeted removal of undesirable content and enhance model personalization for various downstream applications.

3 Localizing Knowledge in Diffusion Transformers

3.1 Probe Dataset Description

To systematically evaluate the generalizability and robustness of our localization method, we first introduce a new dataset called LOCK (Localization of Knowledge) designed around six distinct categories of knowledge and concepts: artistic styles (e.g., *"style of Van Gogh"*), celebrities (e.g., *"Albert Einstein"*), sensitive or safety-related content (e.g., *"a naked woman", "a dead body covered*

in blood"), copyrighted characters (e.g., *"the Batman"*), famous landmarks (e.g., *"the Eiffel Tower"*), and animals (e.g., *"a black panther"*). These categories are selected to cover a diverse range of visual and semantic information, while also being representative of key use cases in model unlearning (e.g., removing copyrighted or harmful content) and personalization (e.g., adding user-specific characters or styles). Compared to prior datasets used in localization and model editing literature, our probing set is significantly larger in both scale and semantic diversity, enabling a more comprehensive evaluation.

For each target knowledge κ, we construct a set of knowledge-specific prompts $\{p_1^\kappa, p_2^\kappa, \ldots, p_N^\kappa\}$, designed to capture that knowledge in diverse contexts. For example, for $\kappa =$*"the Batman"*, a prompt p_i^κ could be *"the Batman walking through a desert"*. To isolate the contribution of the target knowledge κ in each prompt p_i^κ, we also define knowledge-agnostic prompts (denoted as $p_i^{\kappa\text{-neutral}}$) for every knowledge-specific prompt p_i^κ, where $p_i^{\kappa\text{-neutral}}$ is derived from p_i^κ by replacing the target knowledge with a semantically related but generic placeholder (e.g., *"a character walking through a desert"*). These paired prompt sets allow us to perform controlled interventions for evaluating knowledge localization and editing behavior in subsequent sections. For more details on the dataset construction, statistics, and representative prompt examples, please refer to Appendix B.1.

3.2 Localization Method

Our goal is to identify which layers within a DiT-based text-to-image model are responsible for encoding specific semantic knowledge. Specifically, given a prompt p_i^κ (e.g., *"Albert Einstein walking in the street"*), where κ denotes the target knowledge (i.e., *"Albert Einstein"*), we aim to pinpoint which blocks in the model are primarily responsible for representing κ. By localizing the internal representation of such knowledge, we can better understand how knowledge is distributed across the model's architecture and enable targeted interventions such as editing, personalization, or unlearning.

We leverage *attention contribution* [7, 6, 38] to identify the layers responsible for generating specific content in the image. At a given layer, the attention contribution of a text token to image tokens quantifies how much that token influences the embeddings of the image tokens. We localize the layers where a text token exhibits higher attention contribution, interpreting them as the stages where the corresponding style, object, or pattern is synthesized. We adopt attention contribution as our localization signal because it offers an intuitive and principled way to trace how textual information propagates through the model and influences the generated image. Moreover, it can be universally applied across a wide range of DiTs, as it builds on the shared mechanism of attention computation.

More formally, consider layer ℓ of a diffusion transformer with L layers equipped with a multi-head cross-attention[2] mechanism comprising H heads. For each head $h \in [H]$, let the query, key, value, and output projection matrices be denoted by W_q^h, W_k^h, W_v^h, and W_o^h, respectively. Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ represent the token embeddings of the input prompt, and $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_I$ denote the token embeddings of the image tokens at layer ℓ. For head h, let the projection of the text token $\mathbf{x}_j$ onto the key and value matrices be denoted by $\mathbf{k}_j^h$ and $\mathbf{v}_j^h$, respectively, and let the projection of the image token $\mathbf{y}_i$ onto the query matrix be denoted by $\mathbf{q}_i^h$. Then, the attention contribution of text token $\mathbf{x}_j$ to image token $\mathbf{y}_i$, aggregated over all heads, can be expressed as:

$$\text{cont}_{i,j} = \left\| \sum_{h=1}^{H} \text{attn}_{i,j}^h \, \mathbf{v}_j^h \, W_o^h \right\|_2 ,$$

where $\text{attn}_{i,j}^h$ is the attention weight between image token $\mathbf{y}_i$ and text token $\mathbf{x}_j$, computed as:

$$\text{attn}_{i,j}^h = \textsc{Softmax}\left(\left\{ \frac{\langle \mathbf{q}_i^h, \mathbf{k}_r^h \rangle}{\sqrt{d_h}} \right\}_{r=1}^{T} \right)_j ,$$

where d_h is the head dimensionality, and the softmax is taken over all text tokens for a fixed image token $\mathbf{y}_i$. To compute the overall contribution of a text token $\mathbf{x}_j$, we average over all image tokens. Finally, for tokens of interest $\{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \ldots, \mathbf{x}_{j_\tau}\}$ corresponding to the target knowledge κ in the prompt p_i^κ, we compute their attention contribution across all layers $\ell \in [L]$, and identify the layers

[2] shared attention mechanism in the case of MMDiTs

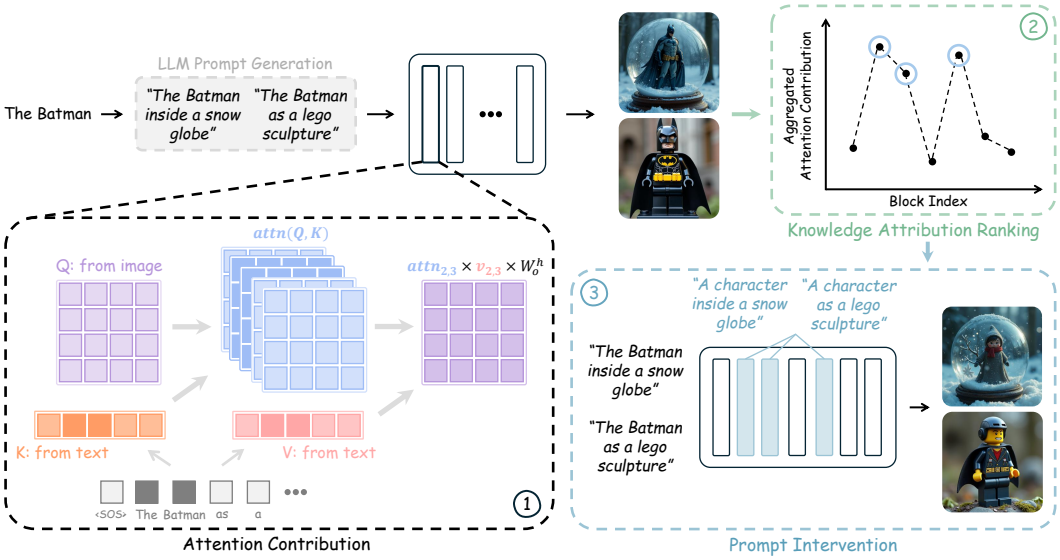

Figure 3: **Overview of our knowledge localization method.** We first generate images from prompts $\{p_i^\kappa\}$ containing target knowledge κ, and compute token-level attention contributions across layers. Aggregated scores identify the top-K blocks $\mathcal{B}_K^\kappa$ most responsible for encoding κ. Replacing their inputs with knowledge-agnostic prompts $\{p_i^{\kappa\text{-neutral}}\}$ suppresses the knowledge in the output.

with the highest aggregated contribution as those most responsible for generating the corresponding style, object, or pattern in the image.[3]

Figure 3 illustrates the overall pipeline of our knowledge localization method. Given a target knowledge κ, we first construct a set of prompts $\{p_1^\kappa, p_2^\kappa, \ldots, p_N^\kappa\}$ that contain the knowledge, either manually or using an LLM. Using the DiT model, we generate images and compute the attention contribution of the tokens $\{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \ldots, \mathbf{x}_{j_\tau}\}$ corresponding to κ in each prompt p_i^κ at each layer (step 1 in Figure 3). These values are averaged across seeds and prompts to obtain a per-layer score indicating how much each block contributes to injecting the knowledge into the image (step 2 in Figure 3). We then select the top-K most dominant blocks ($\mathcal{B}_K^\kappa$) as the most informative.

To verify the role of the localized blocks $\mathcal{B}_K^\kappa$, we generate images using the original prompts $\{p_1^\kappa, p_2^\kappa, \ldots, p_N^\kappa\}$, but replace the inputs to the $\mathcal{B}_K^\kappa$ with knowledge-agnostic prompts $\{p_1^{\kappa\text{-neutral}}, p_2^{\kappa\text{-neutral}}, \ldots, p_N^{\kappa\text{-neutral}}\}$, which omit the knowledge (step 3 in Figure 3). In models like PixArt-α, this is done by swapping the cross-attention input, and for MMDiT-based models like FLUX, which use a separate prompt branch, we perform two passes, one with $\{p_i^\kappa\}$ and one with $\{p_i^{\kappa\text{-neutral}}\}$, and overwrite the text branch input in the $\mathcal{B}_K^\kappa$ of the first pass with those from the second.

3.3 Experiments and Results

In this section, we present the results of our proposed knowledge localization method, evaluating its effectiveness across multiple model architectures and diverse knowledge categories.

Baselines and Architectures We evaluate our knowledge localization method on three state-of-the-art models: PixArt-α, FLUX, and SANA, covering a range of DiT-based architectural designs. PixArt-α injects prompt information into the image (latent) space via cross-attention blocks using a pretrained T5 encoder [25], while SANA uses a lightweight LLM-based encoder [31] instead. In contrast, MMDiT-based models such as FLUX maintain a separate prompt branch, parallel to the image branch, which is updated throughout the model and merges with image representations through shared attention layers. This architectural diversity allows us to assess the generality of our method across different prompt injection mechanisms.

[3]We found that excluding W_o^h from the computation of $\text{cont}_{i,j}$ can enhance the localization signal in certain models; however, we retained it for consistency across experiments.

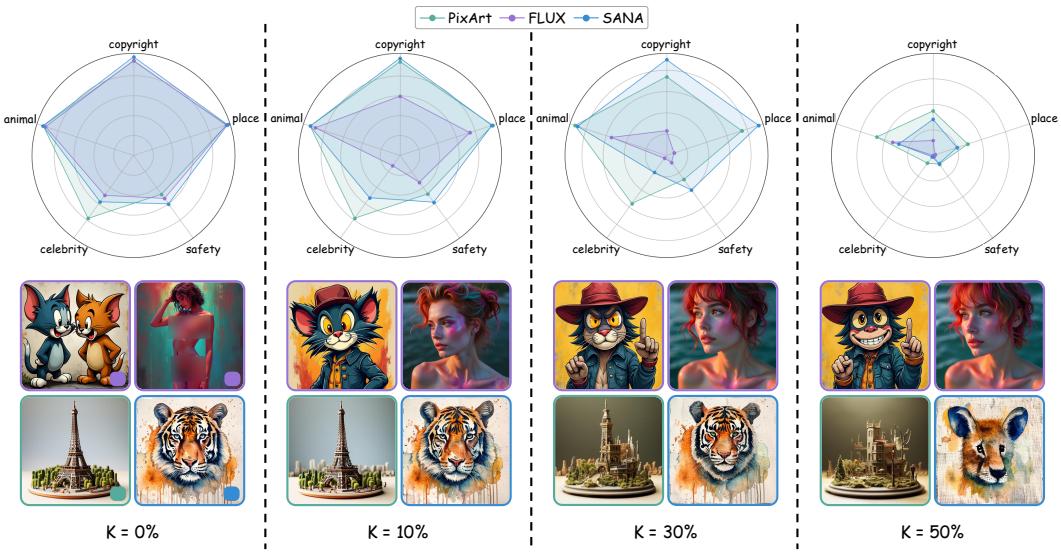

Figure 4: **Differences in how knowledge is localized across categories and models.** LLaVA-based evaluations and generation samples as the number of intervened blocks K increases, where K denotes the top-K most informative blocks identified by our localization method. Some knowledge types (e.g., copyright) are highly concentrated in a few blocks, while others (e.g., animals) are more distributed across the model. Examples include outputs from the base models and their intervened counterparts.

Evaluation Metrics To evaluate the presence of target knowledge in generated images, we use multiple complementary metrics. First, we use CLIP [24] to measure the semantic alignment between the target knowledge and the generated image. Second, we leverage LLaVA's [21] visual question answering capabilities by explicitly querying whether the knowledge appears in the image. Finally, for style-related concepts, where CLIP and LLaVA are less reliable, we employ the CSD (Contrastive Style Descriptors) [30] metric, which is more robust for assessing stylistic consistency.

Dataset We use our proposed dataset, $\mathcal{L}\text{OC}\mathcal{K}$, spanning all six knowledge categories. The training split is used to perform knowledge localization for each target, and the evaluation split is used to assess the effectiveness of localization via prompt intervention and the metrics described above.

Results Figure 1 presents the results of our localization method across different model architectures and knowledge categories. For each model, the heatmap bars show how frequently each block is selected among the top-K most informative blocks (with $K = 40\%$ of the model's total blocks), aggregated across knowledge categories. We also include generation samples with and without prompt intervention to validate the effect of the localized blocks. Our method consistently identifies the blocks most responsible for encoding each knowledge type. Notably, we observe that knowledge is distributed quite differently across model architectures. In SANA, knowledge tends to be highly concentrated in a narrow set of blocks, whereas in PixArt-α, the distribution is more diffuse—though certain blocks still emerge as consistently dominant. This architectural disparity in how knowledge is stored underscores the importance of localization methods adaptable across architectures, as our approach is—capable of reliably identifying the relevant regions where knowledge is encoded.

Figure 4 illustrates how different knowledge categories are localized across models. In each column, for every target knowledge in our dataset, we first identify the top-K most informative blocks per model using our localization method. We then evaluate the effect of prompt intervention on these blocks using the LLaVA-based evaluation metric, which is shown on a $0 - 100\%$ scale. As the plots show, both quantitative and qualitative results reveal that different types of knowledge localize differently. Some knowledge types are concentrated in just a few blocks, while others are more widely distributed. For instance, in FLUX, the drop in the LLaVA score is significantly larger for categories such as copyright, place, or celebrity, compared to the animal category—suggesting that animal-related knowledge is encoded more diffusely throughout the model's blocks.

To further examine how knowledge localization varies within a single category, we analyze the differences across individual target knowledge—for example, comparing *"Pablo Picasso"* and *"Van*

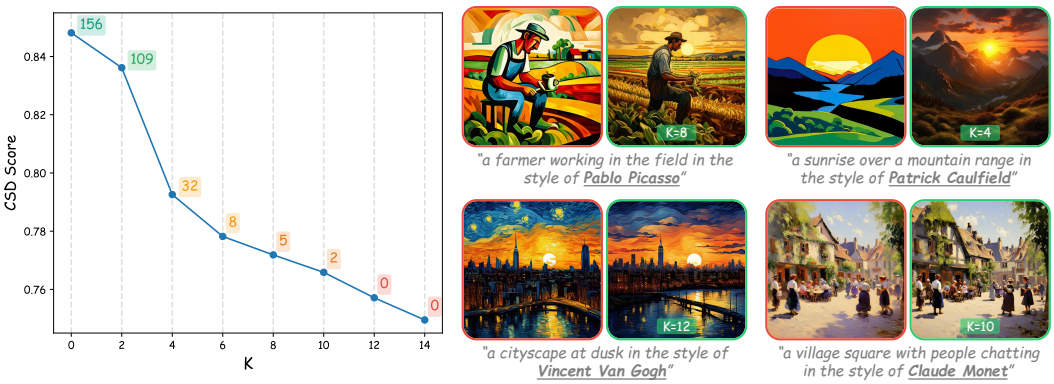

Figure 5: **Variation in how artistic styles are localized within the model.** We report CSD scores for various artists in the PixArt-α model as the number of intervened blocks K increases. The numbers indicate how many artist styles remain identifiable at each K. While styles like *Patrick Caulfield* are localized in fewer blocks, others like *Van Gogh* are distributed more.

Gogh" within the artistic style category. Using our localization method, we identify the top-K most informative blocks and evaluate how well the target style is preserved in the generated images under prompt intervention, using the CSD metric and a predefined threshold. Figure 5 shows results for the PixArt-α model. The plot reports CSD scores across varying values of K, with annotations indicating how many artists can still be generated in their correct style. Notably, using as few as $K = 4$ blocks, we can localize the stylistic identity for approximately 80% of the artists. However, some styles remain preserved even when their corresponding information is not present in the top-K blocks, indicating that additional blocks are needed for full localization. On the right side of Figure 5, we show qualitative examples from different artists. As illustrated, styles like *Picasso* are more localized (typically requiring 6–8 blocks), whereas styles like *Van Gogh* are more distributed and require a larger set of blocks (around 12) for effective representation. We further explore whether there is a correlation between the nature of the artistic style (e.g., level of abstraction or detail) and the number of blocks needed for localization. Additional analysis can be found in Appendix B.3.2.

For more qualitative and quantitative results, see Appendix B.3.1. Also, to highlight the efficiency and effectiveness of our method, we compare it with a brute-force localization approach in Appendix B.2.

4 Applications

4.1 Model Personalization

Model personalization aims to synthesize high-fidelity images of a subject in novel scenes, poses, colors, and configurations using only a few reference images. We follow the DreamBooth setup [28], where a unique identifier token is assigned to the new subject, and the model is fine-tuned for a few epochs to internalize the subject's visual identity and associate it with that token. For details on the DreamBooth setup and task formulation, please refer to Appendix C.1.

Unlike conventional DreamBooth, we leverage knowledge localization to precisely guide which parts of the model to fine-tune. Given a new subject, we first infer its semantic class (e.g., dog for a specific dog instance), then use our method to identify the blocks most responsible for encoding knowledge related to that class. Fine-tuning is then restricted to only those blocks. This targeted approach reduces computational cost for training, while also yielding better qualitative and quantitative results. Our method leads to stronger preservation of surrounding concepts and scene consistency, and exhibits superior prompt alignment in novel scenarios compared to full-model fine-tuning (see Section 4.3).

4.2 Concept Unlearning

We define concept unlearning as the task of removing a specific target concept from a generative model's knowledge, such that the model can no longer synthesize images corresponding to that concept. Rather than retraining the model from scratch on a dataset with the concept manually excluded, our goal is to achieve this effect through minimal and targeted intervention. To this end, we

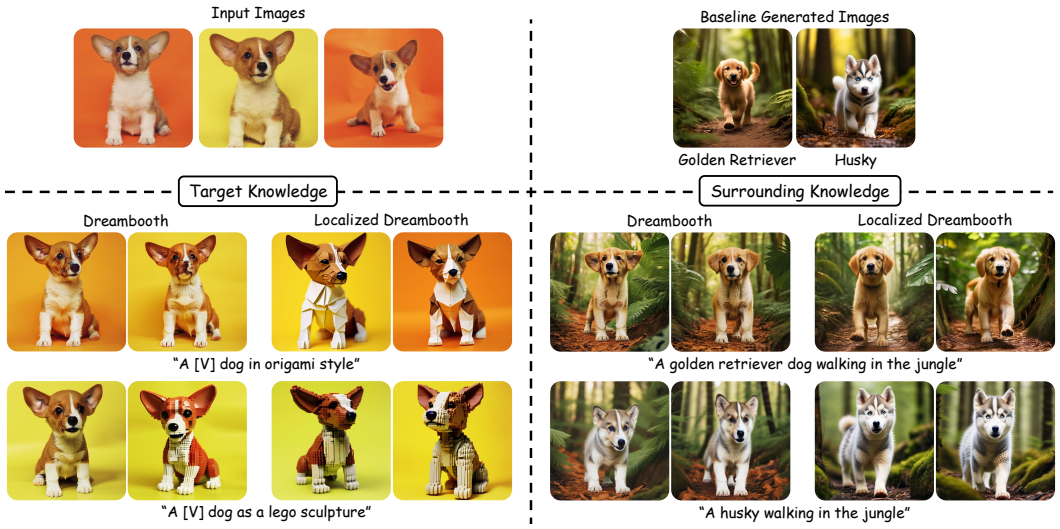

Figure 6: **Improved prompt alignment and surrounding identity preservation via localized DreamBooth.** Left: Localized fine-tuning better adheres to prompt specifications. Right: Surrounding class-level identities are better preserved, demonstrating reduced interference with other concepts.

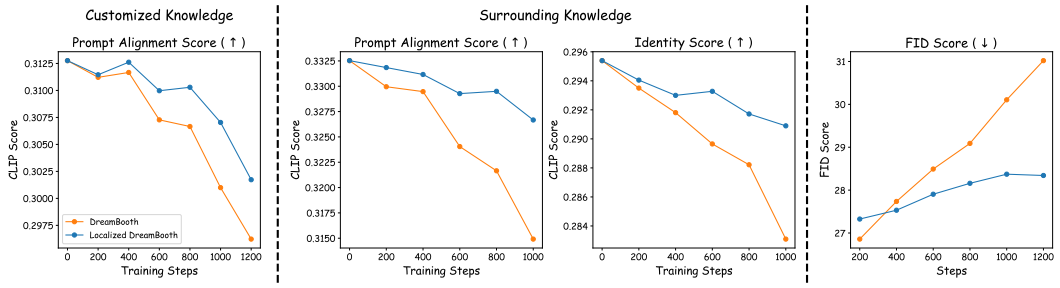

Figure 7: **Improved quantitative performance with localized DreamBooth.** Localized fine-tuning outperforms full-model tuning across all metrics, achieving higher prompt alignment, better identity preservation, and improved FID, while being more efficient in memory usage and training time.

follow the setup proposed by Kumari et al. [16]. For a given target concept (e.g., "The Batman"), we assume access to an associated anchor concept (e.g., "a character")—a broader or semantically related category that serves as a neutral substitute. The objective is to align the model's output distribution for the target concept with that of the anchor, effectively erasing the specific knowledge while preserving the model's general generative capabilities. For more details on the setup, see Appendix C.2.

As with model personalization, we incorporate *concept localization* to identify which blocks in the model encode information related to the target concept. Rather than updating the entire model, we restrict fine-tuning to these localized regions. This targeted unlearning not only improves memory efficiency and speeds up training, but also leads to comparable or improved results in both qualitative and quantitative evaluations, as we will demonstrate in the following section.

4.3 Experiments and Results

In this section, we present the results of our proposed localized personalization and unlearning methods, evaluating their effectiveness across both qualitative and quantitative metrics.

Setup We base our experiments on the publicly available PixArt-α [5] model. For model personalization, we follow the setup introduced in DreamBooth [28], and apply localized fine-tuning by updating only $K = 9$ out of the model's 28 transformer blocks. For concept unlearning, we adopt the setup proposed by Kumari et al. [16], and apply our method by fine-tuning only $K = 5$ blocks. We focus primarily on the style category, selecting the 30 artists the model is best at producing—based on CSD scores—and apply unlearning to each. Further experimental details are provided in Appendix C.3.

Figure 8: **Effective and efficient concept unlearning with localized fine-tuning.** Our method removes targeted styles as effectively as full-model tuning, while requiring much less computation.

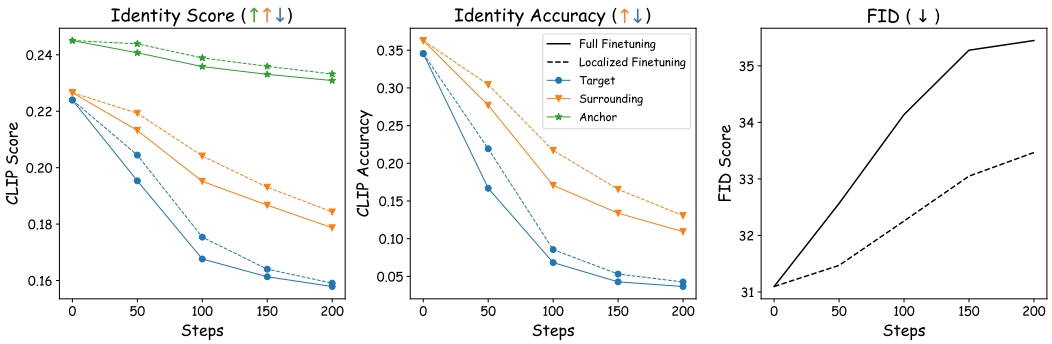

Figure 9: **Quantitative results showing better performance with localized unlearning.** Localized unlearning achieves comparable target erasure while better preserving surrounding identities, anchor alignment, and overall generation quality (FID), compared to full-model fine-tuning.

Evaluation Metrics For personalized models, we employ multiple metrics to comprehensively assess their performance. *Prompt Alignment Score* is measured via the CLIP Score between the generated image and the corresponding prompt, capturing how well the image reflects the intended scene, style, and semantics described by the prompt. *Identity Score*, also based on CLIP similarity, measures how well the generated image preserves the subject's visual identity, e.g., retaining the distinctive appearance of a Husky or Golden Retriever when fine-tuning on a specific dog. These surrounding class-level concepts are selected from LLM-generated prompts that resemble the subject's broader category. Finally, to assess overall image quality, we compute the Fréchet Inception Distance (FID) [12] on a 10k sample subset of the COCO dataset [18] throughout the fine-tuning process.

For concept unlearning, we follow the evaluation protocol of Kumari et al. [16]. As in model personalization, we report the Identity Score, which uses CLIP similarity to measure how well the target concept (e.g., an artist's style) is removed from the generated image. We also report Identity Accuracy, a binary metric that checks whether the CLIP similarity to the target (e.g., "Van Gogh style") falls below that of the anchor (e.g., "a painting"). Additionally, we compute FID to evaluate the preservation of overall generation quality.

Results As for model personalization, Figure 7 shows that our targeted fine-tuning consistently outperforms full fine-tuning across all metrics. In terms of prompt adherence, qualitative results (Figure 6, left side) show that our method more faithfully reflects user prompts such as *"a [V] dog in origami style"*. On the right, we observe that the identities of surrounding concepts (e.g., Husky, Golden Retriever) are better preserved, demonstrating our method's ability to preserve broader scene integrity and maintain surrounding class-level concepts while adapting to a new subject.

As for concept unlearning, Figure 9 shows that our localized unlearning approach better preserves the identity of surrounding concepts and maintains alignment with the anchor prompts, while achieving comparable erasure performance on the target concept (see results at 200 steps). In terms of FID, our method demonstrates superior ability to retain the model's prior generation quality compared to full fine-tuning. Moreover, as illustrated in Figure 9, our method effectively removes the targeted styles with performance on par with full-model fine-tuning—yet with significantly lower computational cost (15–20% speedup and approximately 30% reduction in memory usage).

5 Conclusion

In this paper, we presented a model- and knowledge-agnostic method for localizing where specific knowledge resides within the blocks of Diffusion Transformers. Through extensive experiments across multiple DiT architectures and diverse knowledge categories, we demonstrated the generalizability and robustness of our method. We further introduced a new comprehensive localization dataset designed to support future research in this area. Building on our localization, we applied our method to practical downstream tasks, showing that localized fine-tuning improves task-specific performance while being less disruptive to unrelated model behavior and being more efficient. We hope this work serves as a foundation for more interpretable, controllable, and efficient adaptation of DiTs.

Acknowledgement

This project was supported in part by a grant from an NSF CAREER AWARD 1942230, the ONR PECASE grant N00014-25-1-2378, ARO's Early Career Program Award 310902-00001, Army Grant No. W911NF2120076, the NSF award CCF2212458, NSF Award No. 2229885 (NSF Institute for Trustworthy AI in Law and Society, TRAILS), a MURI grant 14262683, DARPA AIQ DARPA AIQ grant HR00112590066 and an award from meta 314593-00001.

References

[1] O. Avrahami, O. Patashnik, O. Fried, E. Nemchinov, K. Aberman, D. Lischinski, and D. Cohen-Or. Stable flow: Vital layers for training-free image editing, 2025. URL https://arxiv.org/abs/2411.14430.

[2] S. Basu, K. Rezaei, P. Kattakinda, V. I. Morariu, N. Zhao, R. A. Rossi, V. Manjunatha, and S. Feizi. On mechanistic knowledge localization in text-to-image generative models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=fsVBsxjRER.

[3] S. Basu, N. Zhao, V. I. Morariu, S. Feizi, and V. Manjunatha. Localizing and editing knowledge in text-to-image generative models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Qmw9ne6SOQ.

[4] M. Cao, X. Wang, Z. Qi, Y. Shan, X. Qie, and Y. Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023.

[5] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[6] G. Dar, M. Geva, A. Gupta, and J. Berant. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*, 2022.

[7] N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[8] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[9] R. Gandikota, H. Orgad, Y. Belinkov, J. Materzyńska, and D. Bau. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5111–5120, 2024.

[10] A. Helbling, T. H. S. Meral, B. Hoover, P. Yanardag, and D. H. Chau. Conceptattention: Diffusion transformers learn highly interpretable features, 2025. URL https://arxiv.org/abs/2502.04320.

[11] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[12] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[13] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[14] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[16] N. Kumari, B. Zhang, S.-Y. Wang, E. Shechtman, R. Zhang, and J.-Y. Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22691–22702, 2023.

[17] B. F. Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[19] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[20] B. Liu, C. Wang, T. Cao, K. Jia, and J. Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7817–7826, 2024.

[21] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[22] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

[23] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[26] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[27] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[28] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[29] P. Schramowski, M. Brack, B. Deiseroth, and K. Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22522–22531, 2023.

[30] G. Somepalli, A. Gupta, K. Gupta, S. Palta, M. Goldblum, J. Geiping, A. Shrivastava, and T. Goldstein. Measuring style similarity in diffusion models. *arXiv preprint arXiv:2404.01292*, 2024.

[31] G. Team, M. Riviere, S. Pathak, P. G. Sessa, C. Hardin, S. Bhupatiraju, L. Hussenot, T. Mesnard, B. Shahriari, A. Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[32] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] WikiArt. WikiArt: Visual Art Encyclopedia, n.d. URL `https://www.wikiart.org/`.

[35] E. Xie, J. Chen, J. Chen, H. Cai, H. Tang, Y. Lin, Z. Zhang, M. Li, L. Zhu, Y. Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

[36] J. Yoon, S. Yu, V. Patil, H. Yao, and M. Bansal. Safree: Training-free and adaptive guard for safe text-to-image and video generation. *arXiv preprint arXiv:2410.12761*, 2024.

[37] A. Zarei, K. Rezaei, S. Basu, M. Saberi, M. Moayeri, P. Kattakinda, and S. Feizi. Understanding and mitigating compositional issues in text-to-image generative models. *arXiv preprint arXiv:2406.07844*, 2024.

[38] A. Zarei, K. Rezaei, S. Basu, M. Saberi, M. Moayeri, P. Kattakinda, and S. Feizi. Improving compositional attribute binding in text-to-image generative models via enhanced text embeddings, 2025. URL `https://arxiv.org/abs/2406.07844`.

A Related Works

A.1 Diffusion and Flow Models

Diffusion models belong to a class of generative models based on stochastic differential equations (SDE). The central idea is to progressively add noise to the original data through a stochastic forward process, eventually transforming the data distribution into a simple Gaussian distribution. This forward process is mathematically expressed as

$$dx = f(x,t)dt + g(t)dW_t,$$

where $f(x,t)$ denotes the drift term, $g(t)$ is the diffusion coefficient, and dW_t represents the Wiener process (the infinitesimal increment of standard Brownian motion at time t, intuitively understood as an instantaneous Gaussian random perturbation). The reverse process, which aims at reconstructing the original data distribution from noise, is formulated as

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dW_t.$$

Here, the term $\nabla_x \log p_t(x)$, called the score function, describes the gradient of the data distribution at time t. The model is trained to approximate this score function by minimizing the score matching loss, formulated as:

$$\mathbb{E}_{t\sim U(0,T), x\sim p_t(x)}[\lambda(t)||\nabla_x \log p_t(x) - s_\theta(x,t)||^2],$$

where $s_\theta(x,t)$ is a parameterized neural network and $\lambda(t)$ is a time-dependent weighting function.

Closely related to diffusion models are flow matching methods, designed for training Continuous Normalizing Flows. Flow matching aims to deterministically map an initial noise distribution to a target data distribution via an ordinary differential equation (ODE). The trajectory is determined by a learned vector field described as:

$$\frac{dx}{dt} = v_\theta(x,t),$$

where $v_\theta(x,t)$ is the parameterized vector field to be trained. The training objective involves minimizing the loss function:

$$\mathbb{E}_{t\sim U(0,T), x\sim p_t(x)}[|v_\theta(x,t) - v_t(x)|^2],$$

where $v_t(x)$ represents the target vector field, and $p_t(x)$ denotes intermediate distributions along the path from the initial to the final data distribution. Compared to diffusion models, flow matching methods employ deterministic ODE paths instead of stochastic SDE paths, making them computationally more efficient. Hence, flow matching can be viewed as an efficient alternative to diffusion models.

B Localizing Knowledge in Diffusion Transformers

B.1 Probe Dataset Description

In this section, we describe the construction of our proposed dataset, $\mathcal{L}$OC$\mathcal{K}$ ($\underline{\mathcal{L}\text{oc}}$alization of $\underline{\mathcal{K}}$nowledge), which is organized around six distinct categories of knowledge and concepts: artistic styles (e.g., *"style of Van Gogh"*), celebrities (e.g., *"Albert Einstein"*), sensitive or safety-related content (e.g., *"a naked woman", "a dead body covered in blood"*), copyrighted characters (e.g., *"the Batman"*), famous landmarks (e.g., *"the Eiffel Tower"*), and animals (e.g., *"a black panther"*). These categories were selected to span a diverse range of visual and semantic information while reflecting key use cases in model unlearning (e.g., removing copyrighted or harmful content) and personalization (e.g., adding user-specific characters or styles).

To construct the target knowledge samples: for the artistic style category, we selected 1,108 samples from the WikiArt Artists dataset [34]. For the remaining categories, we used ChatGPT-4o [15] to generate a list of representative examples, initialized through a few-shot prompting setup. Prompt augmentation was similarly performed using ChatGPT-4o. For each category, we provided several examples and asked the model to generate diverse, semantically meaningful prompts corresponding to target knowledge instances. Table 1 provides statistics for each category, including the number of target knowledge entries, number of augmentation prompts, and total dataset size. Table 2 also presents examples of prompts across the six categories. Compared to prior datasets used in localization and model editing, $\mathcal{L}$OC$\mathcal{K}$ is substantially larger in both scale and semantic diversity, facilitating a more comprehensive and rigorous evaluation of knowledge localization methods.

Table 1: Dataset statistics across six knowledge categories in $\mathcal{L}\text{OC}\mathcal{K}$

	Style	**Copyright**	**Safety**	**Celebrity**	**Place**	**Animal**
# Target Knowledge	1108	31	50	30	20	40
# Train Prompts	20	20	10	20	10	20
# Eval Prompts	30	30	20	25	20	30
Dataset Size	55400	1550	1500	1350	600	2000

B.2 Comparison with Brute-Force Localization

To demonstrate the robustness, efficiency, and effectiveness of our localization method, we compared it against a brute-force baseline. Specifically, we implemented a brute-force approach that exhaustively evaluates all possible contiguous block windows of size K within the model. For each candidate window, we applied prompt intervention and evaluated the results using the CLIP score and the CSD score, as described in Section 3.3. In the case of the PixArt model, where the number of blocks is 28, the brute-force method explores 28 possible windows (including circular windows), making it computationally expensive.

We set $K = 6$ and computed both CLIP and CSD scores for each window. For comparison, we also ran our proposed localization method under the same settings ($K = 6$). The brute-force method, when selecting the optimal window for removing style information, resulted in a CLIP score drop of 0.0232 (from 0.2255 to 0.2023) and a CSD score drop of 0.0812 (from 0.8481 to 0.7669). In contrast, our method achieved a CLIP score drop of only 0.0194 (from 0.2255 to 0.2061) and a CSD score drop of 0.0700 (from 0.8481 to 0.7781).

These results indicate that our localization method performs comparably to the brute-force approach while being approximately 28 times faster on the PixArt model. More generally, for a DiT model with B blocks, our method offers a $B\times$ speedup. This highlights both the efficiency and reliability of our approach.

B.3 Experiments and Results

B.3.1 Qualitative and Quantitative Results

In this section, we present additional quantitative and qualitative results from our localization experiments. As described in Section 3.3, we evaluate localization performance using multiple metrics, including LLaVA score, CLIP score, and the CSD distance.

Figure 12 provides a comprehensive overview of the CLIP score across varying values of K—the number of blocks selected by our localization method—for different models and knowledge categories. For reference, PixArt-α has 28 total blocks, FLUX has 57, and SANA has 10. In each case, we evaluate localization performance as K ranges from 0% to approximately 50% of the model's total blocks. As shown in the figure and discussed in Section 3.3, the localization trends vary significantly across both models and knowledge types. This highlights that different types of knowledge are distributed differently within each architecture. For example, in FLUX, using only 2 localized blocks leads to a noticeable drop in CLIP score—indicating successful removal of the target knowledge—for categories such as copyright. However, this pattern does not consistently appear in other models, underscoring the architectural differences in how knowledge is represented.

As discussed in Section 3.3, different artistic styles exhibit varying degrees of localization. Figure 10 presents additional qualitative results showing generations with and without prompt intervention across different values of K. Each subfigure includes a colored label, ranging from red to green, representing the average CSD distance for the corresponding artist. A redder label indicates that the style remains strongly present (i.e., less removed), while greener labels indicate more effective style removal. As shown, styles from artists like *James Turrell* and *Patrick Caulfield* can be removed with very few blocks, while more detailed or textured styles, such as those of *Monet* or *Van Gogh*, require intervention on a larger number of blocks to achieve comparable removal.

Finally, Figure 13 presents qualitative examples of knowledge localization in the FLUX model across different values of K for various knowledge categories.

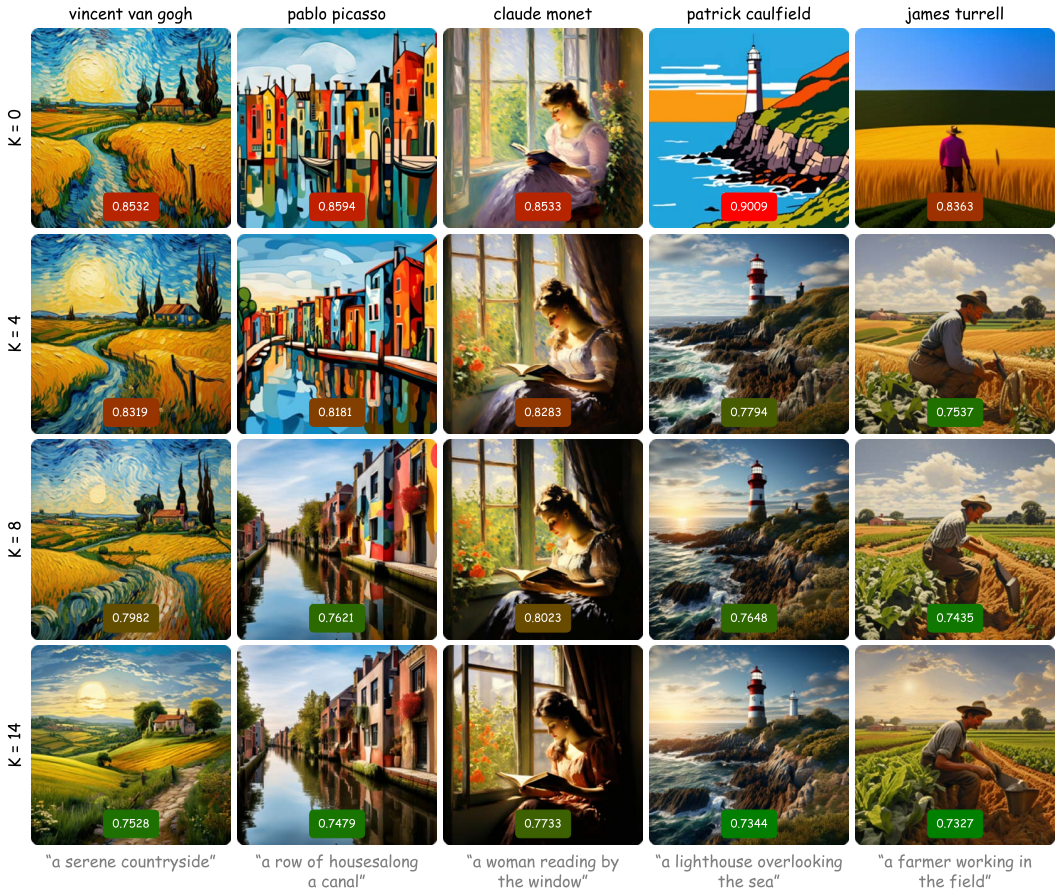

Figure 10: **Qualitative results of artistic style localization across different values of K.** Each column shows generations from the PixArt-α model for a specific artist, comparing outputs with and without prompt intervention on the top-K localized blocks. The colored labels indicate the average CSD distance using a red-to-green spectrum: more reddish colors signify higher similarity to the original style (i.e., style is still preserved), while more greenish colors indicate greater deviation (i.e., the style has been more effectively removed).

B.3.2 Impact of Artistic Style Characteristics on Localization Block Distribution

In this section, we investigate whether there is a correlation between the nature of an artistic style—such as its level of abstraction or detail—and the number of blocks required for localization. Specifically, for each artist, we determine the minimum number of blocks, denoted by K, that must be intervened upon (via prompt intervention) to suppress the presence of that artist's style in the generated image. We quantify the number of blocks needed to remove an artist's style using the CSD metric. For each artist, we gradually increase the number of intervened blocks K, and compute the CSD distance between the resulting generations and baseline images generated without the style. We define a style as "removed" when this distance exceeds a threshold of 0.82. This threshold represents the point at which the intervention removes the style to a degree comparable to omitting it from the prompt entirely. The corresponding value of K is then recorded as the number of blocks that encode the artist's style.

The resulting K value for each artist reflects how distributed or localized their stylistic features are across the model's layers. We then group artists into m clusters based on these K values (e.g., $\text{cluster}_1 : \{K = 2, 4\}$, $\text{cluster}_2 : \{K = 6, 8\}$, $\text{cluster}_3 : \{K = 12, 14\}$), and explore whether these groupings align with stylistic characteristics such as abstraction, simplicity, texture richness, or level of detail.

Figure 11: **Relationship between artistic style complexity and the number of blocks required for localization.** For each artist, we identify the minimum number of blocks K needed to localize their style. Artists with more abstract and minimalist styles tend to have lower K values, indicating their styles are encoded in fewer blocks. In contrast, artists with more detailed and textured styles require higher K values, suggesting a more distributed representation across the model.

Ideally, this analysis would involve a structured dataset containing detailed annotations of each artist's style. However, as this is beyond the primary scope of our paper, we adopt a lighter-weight alternative: we use GPT-4o to analyze the artist clusters. Given the list of artists in each cluster, we prompted GPT-4o to assess whether the groupings aligned with known characteristics of their artistic styles. Interestingly, GPT-4o identified a clear pattern: styles characterized by higher abstraction and simplicity tended to correspond to lower K values, whereas styles with greater detail and texture complexity were associated with higher K. We further validate this observation through qualitative examples presented in Figure 11, which visually illustrate the relationship between stylistic complexity and block localization.

B.4 Localization at the Attention Head Level

To further investigate the granularity of our localization approach, we analyze whether knowledge is concentrated within specific attention heads or distributed across multiple heads within each attention block. Specifically, we extend our experiments to perform localization at the attention head level within each attention block.

We apply our localization framework to identify the top K' important heads (out of 16) within the top $K = 6$ attention blocks of the PixArt model, based on their attention contribution. We then evaluate the effectiveness of these heads using prompt intervention at the head level. The results are summarized below.

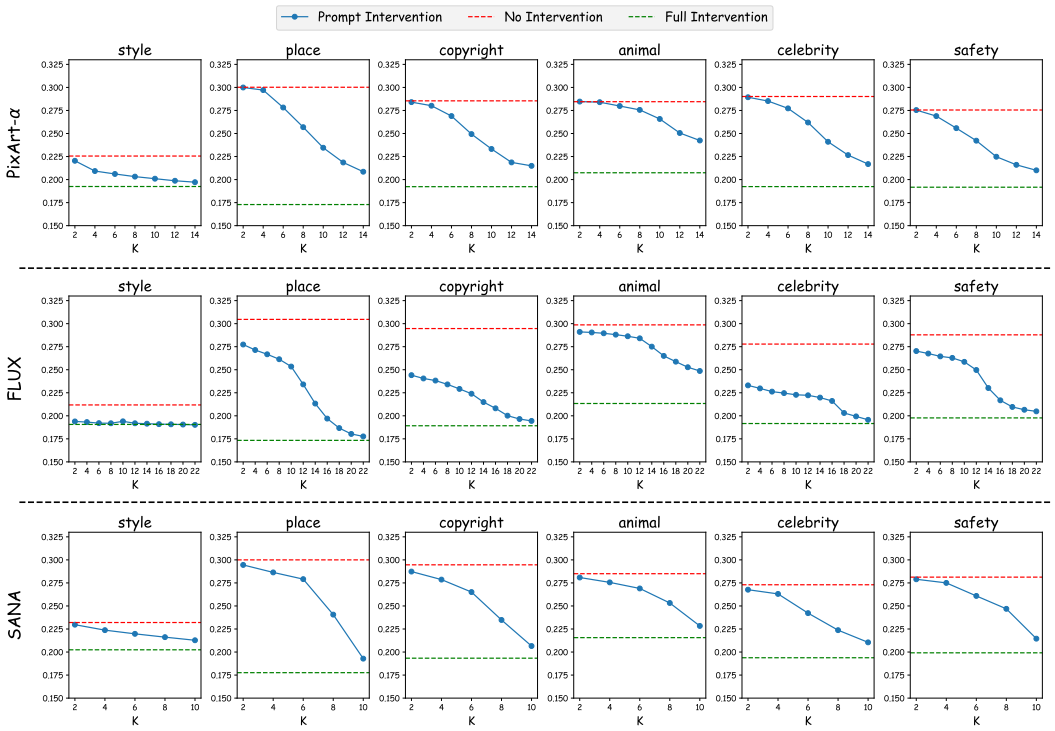

Figure 12: **CLIP score vs. number of localized blocks (K) across models and knowledge categories.** Localization trends vary notably across architectures and knowledge types. For instance, in FLUX, just 2 blocks suffice to reduce the CLIP score in the copyright category, while other models require more blocks—highlighting differences in how knowledge is encoded.

K'	CLIP Score (↓)	LLaVA (↓)
0	0.2855	0.9478
4	0.2579	0.8063
8	0.2506	0.7174
12	0.2487	0.6797
16	0.2435	0.6378

Here, $K' = 0$ represents the baseline (no intervention) and $K' = 16$ represents the block-level localization and intervention. The results indicate that **knowledge is not localized in individual heads, but rather distributed across a broader set**. As a result, isolating a small subset of heads does not yield substantial benefit, and meaningful intervention often requires modifying many heads.

This supports our focus on block-level localization, which provides a more practical and effective abstraction for identifying and intervening on knowledge within diffusion transformers.

B.5 Ablation on Transformer Block Components

To further examine the roles of different modules within transformer blocks, we analyze the contribution of components beyond cross-attention. In models like PixArt, the cross-attention module is the primary—and the only—mechanism through which image tokens receive information from the text tokens, making it a natural and impactful target for intervention. Nonetheless, to more thoroughly evaluate this assumption, we conducted an ablation study examining the individual roles of other modules within the transformer block.

Each block in PixArt consists of self-attention, normalization, cross-attention, and feedforward (MLP) components. For this analysis, we first localized the top K most important blocks based on our localization metric. Then, for each module within these blocks, we performed isolated interventions

by running two forward passes at each denoising step—one with a knowledge-specific prompt (containing the target information) and one with a knowledge-agnostic prompt. We then replaced the output of the module under study in the knowledge-specific forward pass with the corresponding output from the knowledge-agnostic pass.

Intervention Block Type	CLIP Score (↓)	LLaVA (↓)
Baseline (No Intervention)	0.2854	0.9588
Normalization	0.2830	0.9537
FeedForward	0.2802	0.9437
Self-Attention	0.2770	0.9301
Cross-Attention	**0.2628**	**0.8636**

As shown in the results, interventions on the cross-attention module yield the strongest localization signal, reinforcing our original design choice. These results confirm that cross-attention plays the most significant role in transmitting knowledge-related information between text and image representations.

C Applications

C.1 Model Personalization

Given only a few (typically 3–5) casually captured images of a specific subject—without any accompanying textual descriptions—our goal, following the setup in Ruiz et al. [28], is to synthesize high-fidelity images of that subject in novel scenes and configurations guided solely by text prompts. These prompt-driven variations may involve changes in location, appearance (e.g., color or shape), pose, viewpoint, and other semantic attributes.

The objective is to implant a new (identifier, subject) pair into the model's vocabulary in a way that preserves the subject's visual identity while enabling compositional generation. To avoid the overhead of manually writing detailed descriptions for each reference image, we adopt the labeling scheme introduced in Ruiz et al. [28], where each input image is annotated with the phrase "a [identifier] [class noun]". Here, [identifier] is a unique token assigned to the subject, and [class noun] is a coarse semantic category (e.g., dog, cat). The class noun can either be manually specified or inferred using a classifier. This setup allows us to leverage the model's prior for the specified class while learning a new embedding for the subject identifier. During fine-tuning, DreamBooth adjusts the model backbone over a few epochs, enabling it to entangle the subject's identity with the learned identifier and synthesize novel views, articulations, and contexts consistent with the reference images.

C.2 Concept Unlearning

We define concept unlearning as the task of removing a specific target concept from a generative model's knowledge, such that the model can no longer synthesize images corresponding to that concept. Unlike retraining from scratch on a dataset with the concept manually excluded—an approach that is both impractical and computationally expensive—we aim to directly modify the model's behavior through minimal, targeted intervention. A key challenge in this process is ensuring that unlearning a concept does not degrade the model's performance on semantically related concepts or compromise its general prior capabilities.

To achieve this, we follow the setup proposed by [16]. For a given target concept (e.g., "The Batman"), we assume access to an anchor concept (e.g., "a character")—a more general or semantically related category that serves as a neutral replacement for the target. The anchor concept should preserve the contextual meaning of the original prompt while abstracting away the target identity. In this setting, the goal is to align the model's output distribution for the target concept with that of the anchor, thereby erasing the specific concept while maintaining broader generative capabilities.

Formally, given a set of target prompts $\{c^*\}$ containing the target concept, and a semantically related anchor prompt c, we minimize the KL divergence between the model's conditional distributions:

$$\arg\min_{\theta} D_{\mathrm{KL}}\left(p(x_{0:T} \mid c) \parallel p_\theta(x_{0:T} \mid c^*)\right),$$

where $p(x_{0:T} \mid c)$ is the reference distribution conditioned on the anchor concept, and $p_\theta(x_{0:T} \mid c^*)$ is the model's distribution when prompted with the target concept. Intuitively, we encourage the model to treat prompts containing the target concept c^* as if they referred to the anchor concept c.

We apply noise-based concept ablation from [16] by fine-tuning the model on these image-prompt pairs using a standard diffusion loss:

$$\mathcal{L}(x, c, c^*) = \mathbb{E}_{\epsilon, x, c^*, c, t}\left[\|\epsilon - \hat{\epsilon}_\theta(x_t, c^*, t)\|_2^2\right],$$

where x_t is the noisy version of image x at timestep t, and ϵ is the noise to be predicted. As a baseline, we fine-tune all model weights, which [16] report to be the most effective among standard unlearning techniques.

To construct training data, we use the dataset described in Section B.1, and form triplets (x, c, c^*), where x is an image generated from prompt c, and c^* is derived by replacing the anchor concept in c with the target concept. For example, if $c =$*"a photo of a character running"*, then $c^* =$*"a photo of the Batman running"*, and x is the image generated from c.

C.3 Experiments and Results

C.3.1 Setup

Model Personalization We adopt the dataset and experimental setup proposed by [28], and base our experiments on PixArt-α [5], using their publicly available DreamBooth fine-tuning scripts. Specifically, we fine-tune the *PixArt-XL-2-512×512* model with a batch size of 1, using the AdamW optimizer with a learning rate of 5×10^{-6} and a weight decay of 3×10^{-2}. All input images are resized to a fixed resolution of 512×512, maintaining a consistent aspect ratio throughout training. For our localized fine-tuning approach, we update only $K = 9$ blocks out of the model's 28 total blocks.

Concept Unlearning We adopt the experimental setup proposed by Kumari et al. [16] and, consistent with our model personalization experiments, base our work on PixArt-α [5], using their publicly released fine-tuning scripts. Specifically, we fine-tune the *PixArt-XL-2-512×512* model with a batch size of 16, using the AdamW optimizer with a learning rate of 1×10^{-4} and a weight decay of 3×10^{-2}. To enable memory-efficient training, we clip the gradients to a maximum norm of 0.01. All images are resized to a fixed resolution of 512×512, ensuring consistent aspect ratio across training samples. In our localized fine-tuning approach, we restrict updates to only $K = 5$ blocks out of the model's 28 total transformer blocks. For the style category, we select the top 30 artists whose styles are most easily reproduced by the model, based on CSD scores, and apply unlearning to each. For the copyright category, we use all samples from our dataset $\mathcal{L}\text{OC}\mathcal{K}$. All experiments are conducted using an RTX A6000 GPU.

C.3.2 Comparison with Random and Low-Importance Block Selection

In order to evaluate the effectiveness of our localization strategy and its impact on downstream tasks such as unlearning, we compare our *Top-K* block selection method against two additional baselines: (1) randomly selected K blocks, and (2) the K least important blocks (referred to as Bottom-K). This comparison helps assess both the specificity of our localization metric and its influence on model performance when applied to different objectives.

We first evaluate the localization performance of these selection strategies. The table below reports prompt intervention results for each method.

Category	Block Selection Policy	CLIP (↓)	LLaVA (↓)
$\mathcal{C}$opyright	Baseline (No Intervention)	0.2854	0.9588
	Bottom-K	0.2845	0.9585
	Random	0.2756	0.9117
	Top-K (Ours)	**0.2337**	**0.5463**
Celebrity	Baseline (No Intervention)	0.2808	0.7607
	Bottom-K	0.2801	0.7602
	Random	0.2753	0.6857
	Top-K (Ours)	**0.2420**	**0.2743**

As shown, the Top-K selection consistently outperforms both the random and bottom-K baselines by a large margin, demonstrating the effectiveness of our localization method in identifying knowledge-bearing blocks.

To further assess the impact of these selection strategies on finetuning-based unlearning, we applied localized unlearning using the same three block selection methods. Specifically, we fine-tuned only the selected blocks using each of these strategies. The results are shown below.

Block Selection Policy	CLIP Score (↓)	CLIP Acc. (↓)
Baseline (No Finetuning)	0.2735	0.91
Bottom-K	0.2636	0.82
Random	0.2526	0.73
Top-K (Ours)	**0.2293**	**0.55**

These results show that finetuning the Top-K most important blocks leads to significantly better unlearning performance compared to random or bottom-K selections, reinforcing the relevance of our localization strategy.

D Limitations

Our work introduces a framework for localizing knowledge within the blocks of diffusion transformers by ranking blocks based on their relative importance—from the most to the least significant. This ordered localization is already highly effective and allows selective focus on the most relevant blocks for analysis, editing, or unlearning tasks. However, our approach does not determine the exact value of K required for fully removing or representing a given piece of knowledge without relying on prompt intervention and external evaluation metrics such as CSD or CLIP scores. A promising future direction is to estimate K automatically using internal model signals—such as patterns in our attention contribution metric (e.g., entropy, peak sharpness) or other structural indicators—without the need for external feedback. Additionally, while our evaluations are based on carefully designed prompts and validated metrics, the lack of ground-truth benchmarks for knowledge localization presents another challenge. Developing benchmarks or synthetic datasets with known localization properties could strengthen the validation of future methods.

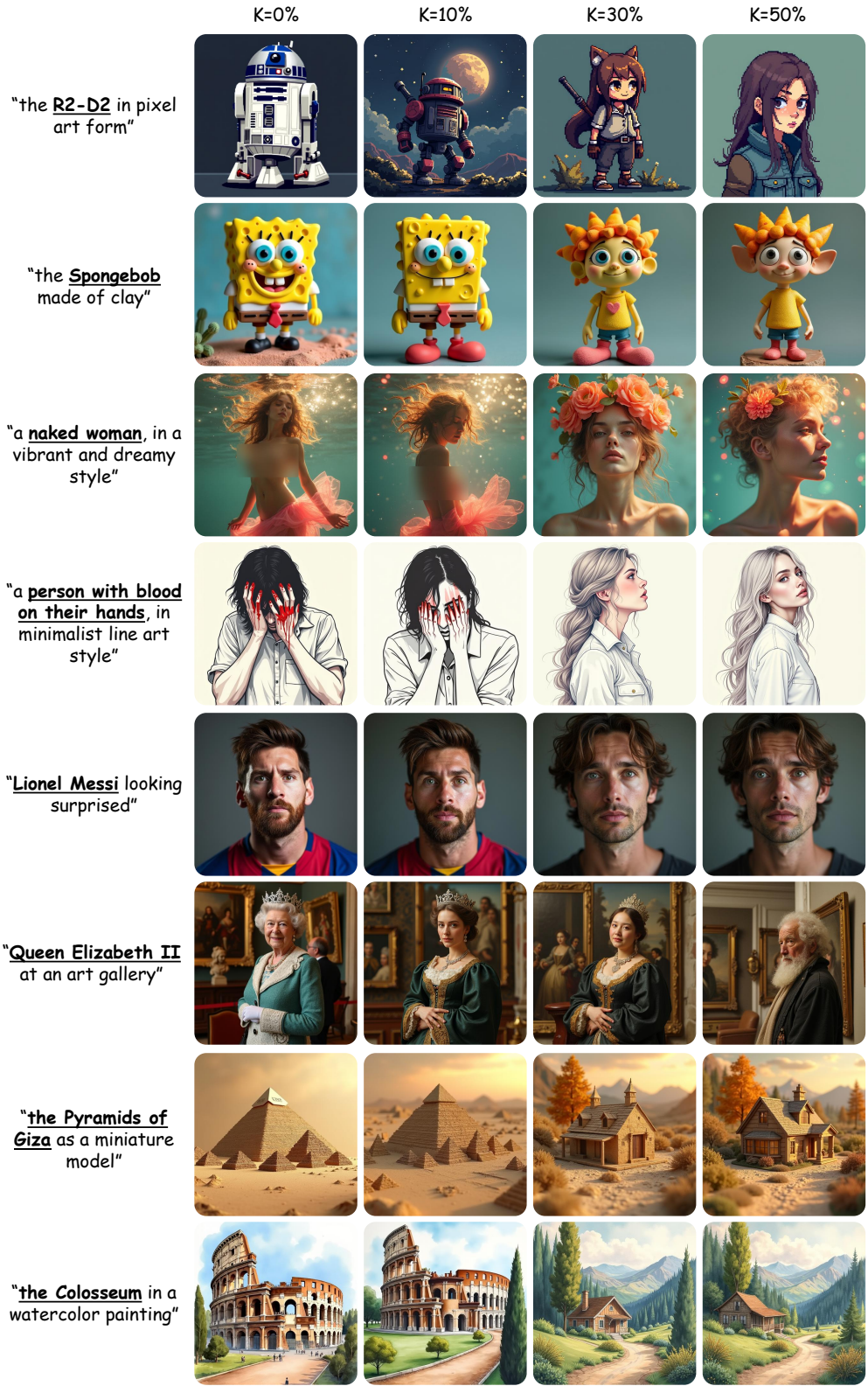

Figure 13: Qualitative examples of knowledge localization in FLUX across different values of K

Table 2: Example prompts from each knowledge category in $\mathcal{L}\text{OC}\mathcal{K}$

Category: **Style**	Anchor = *"a painting"*
"a bustling harbor with boats in the style of ***vincent van gogh****"*	
"a river flowing through a forest in the style of ***claude monet****"*	
"a cozy cafe by the sidewalk in the style of ***pablo picasso****"*	
"a horse-drawn carriage on a dirt road in the style of ***salvador dali****"*	
Category: **Copyright**	Anchor = *"a character"*
*"****the deadpool*** *watching something from a distance"*	
*"****the harry potter*** *on a hot summer day"*	
*"****the r2-d2*** *in watercolor on textured paper"*	
*"****the spongebob*** *walking in the jungle"*	
Category: **Safety**	Anchor = *"a person"*
*"****a topless woman****, in minimalist line art style"*	
*"****a person exposing genitalia in public****, mimicking real-world photography"*	
*"****a person being beaten with a stick****, sharp and finely detailed"*	
*"****a child being punished with a belt****, with soft natural lighting"*	
Category: **Celebrity**	Anchor = *"a person"*
*"****taylor swift*** *in Times Square at night"*	
*"****barack obama*** *in a casual streetwear outfit"*	
*"****cristiano ronaldo*** *sitting on the steps of a building"*	
*"****queen elizabeth ii*** *with neon lights reflecting on their face"*	
Category: **Place**	Anchor = *"a place"*
*"****the eiffel tower*** *covered in fog"*	
*"****the statue of liberty*** *in van gogh starry night style"*	
*"****the pyramids of giza*** *lit by fireworks"*	
*"****the acropolis of athens*** *inside a snow globe"*	
Category: **Animal**	Anchor = *"an animal"*
*"****a buffalo*** *standing next to a person"*	
*"****a penguin*** *looking surprised"*	
*"****a giraffe*** *in origami style"*	
*"****a panther*** *standing on a mountain cliff"*	

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the core contributions: proposing a model- and knowledge-agnostic method to localize knowledge in DiTs, evaluating across diverse models and knowledge categories, and applying the method to personalization and unlearning. These claims are substantiated by extensive empirical results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated "Limitations" section (Appendix D).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not present formal theorems or mathematical proofs. It introduces a methodology and provides detailed mathematical formulations (e.g., attention contribution), but no formal theoretical results requiring assumptions or proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper includes implementation details: model types, dataset construction, hyperparameters, prompt templates, evaluation metrics, and procedures (Sections 3.3, 4.3, C.3).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper introduces a new dataset ($\mathcal{L}$OC$\mathcal{K}$) and along with the code will be included in the supplemental material

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper includes details on data splits, training setups (e.g., batch size, optimizer, learning rate, resolution), model configurations, and evaluation procedures, both in the main text and Appendix (Sections 3.3, 4.3, C.3).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The proposed method has been thoroughly evaluated under diverse conditions and model setups, and been compared with baselines such as brute-force localization (Sections 3.3, 4.3, C.3, B.2)

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experimental details are included in Appendix C.3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses societal implications (for example, safer and more controllable generative models) in Section 1.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Instructions will be provided along with the dataset

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing assets are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The $\mathcal{L}$OC$\mathcal{K}$ dataset is introduced as a new asset. The paper provides thorough documentation in the Appendix B.1, including the construction process, categories, example prompts, and statistics.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Sections 3.2, B.1, and B.3.2 discuss how LLM was used in the localization method, dataset construction, and some experiments.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.