Diffusion Lens: Interpreting Text Encoders in Text-to-Image Pipelines

Anonymous ACL submission

Abstract

Text-to-image diffusion models (T2I) use a latent representation of a text prompt to guide the image generation process. However, the encoder that produces the text representation is largely unexplored. We propose the DIFFUSION LENS, a method for analyzing the text encoder of T2I models by generating images from its intermediate representations. Using the DIFFUSION LENS, we perform an extensive analysis of two recent T2I models. We find that the text encoder gradually builds prompt representations across multiple scenarios. Complex scenes describing multiple objects are composed progressively and more slowly than simple scenes; earlier layers encode the concepts in the prompts without a clear interaction, which emerges only in later layers. Moreover, the retrieval of uncommon concepts requires further computation until a faithful representation of the prompt is achieved. Concepts are built from coarse to fine, with details being added until the very late layers. Overall, our findings provide valuable insights into the text encoder component in T2I pipelines.[1]

1 Introduction

The text-to-image (T2I) diffusion pipeline is made of two components: the text encoder and the diffusion model. The text encoder encodes a text prompt into a latent representation that guides the diffusion process. A few recent papers studied the diffusion model and the cross attention mechanism that connects the two components (Tang et al., 2023; Hertz et al., 2023; Orgad et al., 2023; Chefer et al., 2023a). However, to the best of our knowledge, while the text encoder is a key component of the pipeline with a large effect on image quality and text–image alignment (Saharia et al., 2022), the inner mechanisms of the text encoder have not yet been investigated.

[1]Code and data are available at anonymized.

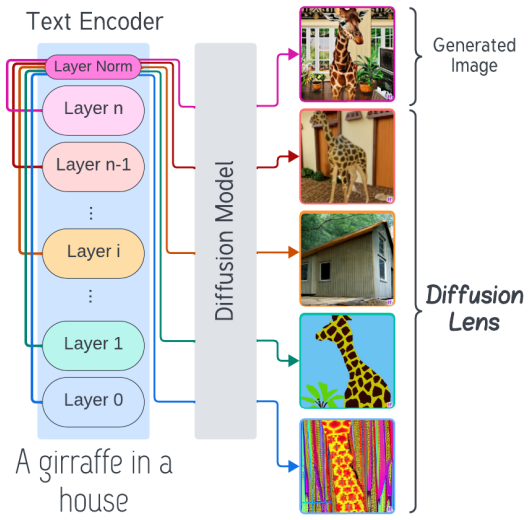

Figure 1: Visualization of the text encoder's intermediate representations using the DIFFUSION LENS. At each layer of the text encoder (in blue), the DIFFUSION LENS takes the full hidden state, passes it through the final layer norm, and feeds it into the diffusion model.

Our main question is, "What can we learn about the computation process by which the text encoder builds the prompt representation?". To this end, we propose the DIFFUSION LENS, a method for analyzing the representations at intermediate layers of the text encoder.

Current T2I architectures use a pre-trained transformer (Vaswani et al., 2017) as their text encoder. Usually, to generate images, the input prompt is passed through the text encoder and the representation after the final layer is used to condition the diffusion process. The DIFFUSION LENS conditions the diffusion process on intermediate representations of the prompt, leading to visually-coherent, human-understandable images for most layers (Figure 1). Notably, the DIFFUSION LENS relies solely on the pre-trained weights of the model and does not depend on any specific task or external modules. Comparing images generated from different layers, we reveal patterns that emerge during the computa-

tion process performed by the text encoder.

We use the DIFFUSION LENS to perform qualitative and quantitative experiments with two popular T2I models – Stable Diffusion (Rombach et al., 2022) and Deep Floyd (StabilityAI, 2023). For each analysis, we either construct a specific dataset to isolate a particular phenomenon, or use a naturally occurring human-written image captions. We uncover several insights regarding the computation mechanism of the text encoder in the T2I pipeline.

First, we examine the T2I model's ability to perform conceptual combination. We find that complex representations (e.g., "A yellow pickup truck and a pink horse") are built incrementally: As depicted in 2 (Left), images generated from representations at early layers encode the concepts either separately or together, but without reflecting the correct relationship between the concepts, acting more as a "bag of concepts". Images from later layers also encode the relation. We find that the sequence in which objects emerge during the computation process is determined by either their linear or their syntactic precedence in the sentence, a factor influenced by the particular text encoder under consideration: Deep FLoyd's text encoder is more sensitive to syntactic structure, while Stable Diffusion's text encoder tends to reflect the linear order.

In the second part, we investigate memory retrieval. We find that faithful representations for familiar concepts, such as the animal "Kangeroo", exist already in early layers, while unfamiliar concepts like the animal "Dik-dik" require a longer computation to generate representations from which the diffusion process can extract a faithful representation, as demonstrated in Figure 2 (Right, top). We also find a difference in memory retrieval patterns between the two text encoders of the models: Deep FLoyd's memory retrieval shows a more incremental behavior than Stable Diffusion's. The differences we found suggest that factors such as architecture, pretraining objective or data may influence knowledge encoding or language representation of the models. Moreover, complex concepts, like specific people, are developed gradually with tiny details being added at each layer, such as hair style, eye color, and at the last layers, their facial features, as shown in Figure 2 (Right, bottom).

Our contributions are summarized as follows:

- We develop the DIFFUSION LENS, a new intrinsic method for analyzing the intermediate states of the text encoder within T2I pipelines.
- Through rigorous experiments, we uncover how complexity, commonality, and syntactic structure influence the computation process of text encoders.

Ultimately, we shed light on text encoder dynamics, and hope this method aids the community in building and evaluating T2I models.

2 Diffusion Lens

Preliminiaries. Current text-to-images diffusion models comprise two main components (Saharia et al., 2022; Ramesh et al., 2022): a language model used as a text encoder that takes the textual prompt as input and produces latent representations; and a diffusion model that is conditioned on the representations from the text encoder and generates an image from an initial input noise.

The language model in the T2I pipeline is typically a transformer model. Transformer models consist of a chain of transformer blocks, each composed of three sub-blocks: attention, multi-layer perceptron, and layer norm (Vaswani et al., 2017).

We denote the transformer block at layer l as F_l. The input to the model is a sequence of T word embeddings, denoted as $\mathbf{h}^0 = [h_1^0, \ldots, h_T^0]$. Then, the output of the transformer block at layer l is a sequence of hidden states $\mathbf{h}^{l+1}$:

$$\mathbf{h}^{l+1} = F_l(\mathbf{h}^l) \quad (1)$$

The output representations of the last block, L, go through a final layer norm, denoted as ln_f. Then, they condition the image generation process through cross-attention layers, resulting in an image I. We abstract this process as:

$$I = \text{Diff}(ln_f(\mathbf{h}^L)) \quad (2)$$

Diffusion Lens. In a T2I pipeline with a text encoder of L layers, for layer $l < L$, we process the output of block l through the final layer norm (ln_f), including padding tokens. We condition the diffusion process on this output, as illustrated in Figure 1. Namely, we generate an image I from an intermediate layer l as follows:

$$I = \text{Diff}(ln_f(\mathbf{h}^{l+1})) \quad (3)$$

The final layer norm is a crucial step in generating coherent images (see further details in Appendix A.3). It projects the representations into the cross-attention embedding space without the caveat of adding new information to the representation, as may happen with learned projections. This process

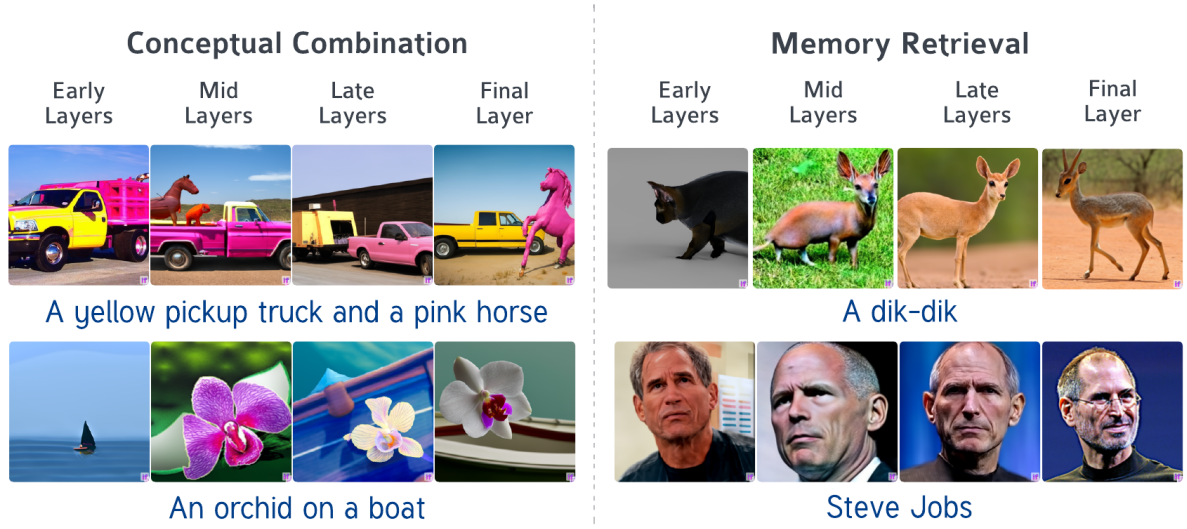

Figure 2: Insights gained from using DIFFUSION LENS. **Conceptual combination** (left): early layers often act as a "bag of concepts", lacking relational information which emerges in later layers. **Memory Retrieval** (right): uncommon concepts gradually evolve over layers, taking longer to generate compared to common concepts.

has a strong potential to generate an image representing the intermediate state of the text-encoder as interpreted by the diffusion model.

3 Experimental Setup

Models. The experiments are performed on Stable Diffusion 2.1 (denoted *SD*, Rombach et al., 2022) and Deep Floyd (denoted *DF*, StabilityAI, 2023). SD is an open-source implementation of latent diffusion (Rombach et al., 2022), with OpenCLIP-ViT/H (Ilharco et al., 2021) as the text-encoder. DF is another open-source implementation of latent diffusion inspired by Saharia et al. (2022), with a frozen T5-XXL (Raffel et al., 2020) as the text encoder. We usually only report the results on DF, unless there is a difference between the models, which we then discuss. The full results on SD are given in Appendix E.

Data. Depending on the specific experiment, we either curate prompt templates and automatically generate a list of prompts from a collected list of concepts we are interested in investigating, or use a list of natural, handwritten prompts from COCO (Lin et al., 2015). The data for each experiment is detailed in the next sections. With each prompt, we generate images that are conditioned on representations from every fourth layer in the model, which serves as a representative subset. This results in 7 images for DF (which has 25 layers in total) and 6 images for SD (which has 24). We generate each prompt using four random seeds.

Evaluation. For every experiment we ask questions regarding the images at every layer, e.g., "Does the prompt correspond to the generated image"; or, when there are two objects in the prompt, "Does object A appear in the generated image?". We describe the questions in detail for every experiment below. We collected answers to the questions by ten human annotators, with 10% overlap to measure inter-annotator agreement.

In one case, when we found that more samples are needed due to high variance of the results. In this case, we added additional samples annotated with GPT-4V (OpenAI, 2023) to the human annotated samples, after validating the agreement between the model and human annotators. Overall, we collected answers to roughly $66,560$ questions, 37% of them by GPT-4V. For full details on the annotation process, inter-annotator agreement and integration with GPT-4V, refer to Appendix B.

We always ask the annotator if the generated image matches the prompt. As we aim to analyze a full representation building process, we report our main findings only on successful generations where the answer at the last layer is "yes". Later, we separately analyze failure cases in Section 6.

4 Conceptual Combination

T2I diffusion models are popular for their ability to generalize beyond their training data, creating composite concepts (Ramesh et al., 2022). Conceptual combination is the cognitive process by which at least two existing basic concepts are combined

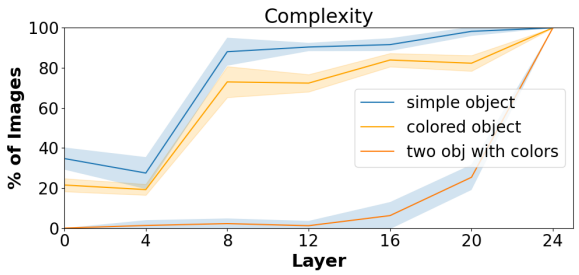

Figure 3: Percentages of prompt-matching images across various layers. As prompts become more complex, DIFFUSION LENS has to utilize more layers to extract a correct image.

to generate a new higher-order, composite concept (ling Wu and Barsalou, 2009). Conceptual combination is at the core of knowledge representation, since it asks how the meaning of a complex phrase connects to its component parts (Hampton, 2013), e.g., "A cat in a box". This section uses the DIFFUSION LENS to trace the process by which the text encoder creates composite concepts.

4.1 Building complex scenarios

This study investigates the text encoder's ability to combine concepts at varying levels of complexity. We utilize COCO classes (Lin et al., 2015) as a diverse set of prompts with readily identifiable visual meanings. Each experiment commences with a simple list of objects as prompts, progressively increasing in complexity as outlined subsequently.

Colors and conjunction. We compile three lists of prompts: (1) objects (e.g., "a dog"); (2) objects with color description ("a red dog"); and (3) two objects with color description ("a red dog and a white cat"). To investigate how conceptual combination emerges throughout the layers, we annotated a random sample of 80 [2], asking the following questions for each layer: (a) Does object X appear in the image? (b) Does color X appear in the image? (c) Does object X appear in the correct color? X is either the 1st or the 2nd object, for a total of six questions.

Physical relations. We compile two lists of prompts: (1) objects; and (2) a list where each prompt describes two objects and a preposition—either "in" or "on"—-for example, "A cat in a box". As before, we sample 40 prompts. We ask three questions: (a-b) Does object X appear in the image? X is either object A or B, and (c) Is object A in / on object B?

[2]In this experiment, human annotators annotated 40 prompts and GPT4-V annotated additional 40.

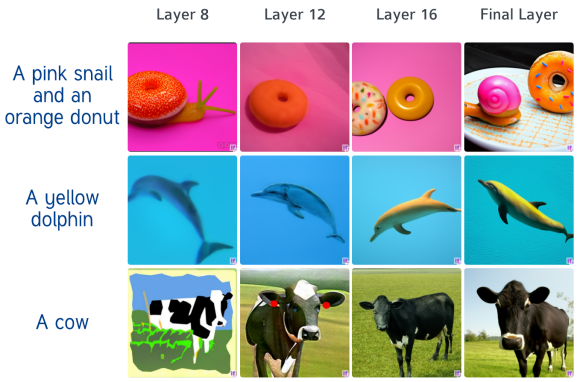

Figure 4: Complex prompts take more computation blocks to emerge.

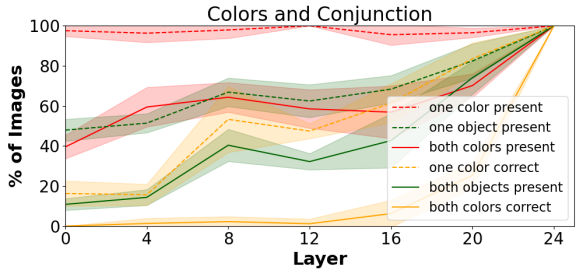

Figure 5: The proportion of images where either the object, the colors, or both were present, and where either the objects or the colors were accurately represented.

Results

The simpler the concept, the earlier it emerges. Figure 3 shows the percentage of images that correctly generated the concepts for each category: objects alone, an object and a color, and two objects and colors. Prompts describing a single object emerge the earliest, between layers 4 and 16, while prompts containing a color descriptors emerge in layers 16–20. Conjunction prompts emerge last, around layers 20–24. We observe a similar pattern for the preposition prompts, which we describe in Appendix A.1. As demonstrated in Figure 4, "A cow" is fully represented by layer 8, while "A yellow dolphin" does not correctly form until layer 16. Lastly, "A pink snail and an orange donut" only fully forms at much later layers, correctly matching the objects and colors at the final layer, 24.

The complex representation is constructed gradually. We continue to investigate the complex prompts of two colored objects. Figure 5 aggregates the answers to illustrate the behavior of either or both objects in intermediate layers. Colors often emerge first, with both colors often emerging in early layers. A single object is also gradually

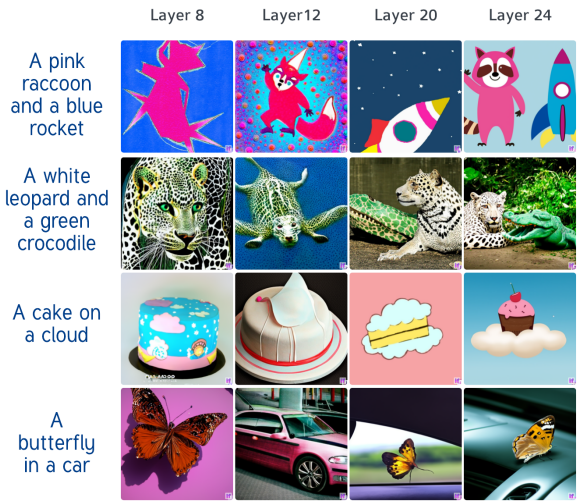

Figure 6: Complex representations are constructed gradually. In some cases, objects are mixed in early representations. In other cases, only one of the objects appear in early representations.

represented in layers 4-12. Notably, while the colors and one of the objects appear, the object is not necessarily generated in the correct color. This can be seen in the first example in Figure 6: While a raccoon and a rocket do appear, and the image contains both blue and pink elements, the rocket is not blue until the final layer. In some cases, we observe a mixture of concepts in early layers, as seen in the second example of Figure 6. Similarly, the bottom two examples in Figure 6 show prompts composing two objects and a proposition. As with colors, we observe that individual objects appear in early layers but the correct relation emerges much later. For example, "A cake on a cloud" generates images of both a cake and a cloud, with different relations; at layer 8 the cake is decorated with a cloud and in layer 20 the clouds are depicted as frosting. The correct relation is only generated at the final layer. These patterns suggest that the early layers of the text encoder behave like a "bag of concepts", with a representation for each concept but no clear relations between them.

4.2 Syntactic dependencies

To investigate the order in which different objects emerge, we focus on the association between syntactic depth and the appearance order of nouns. Specifically, we explore whether, in a dependency path where noun A precedes noun B, noun A appears at earlier layers through DIFFUSION LENS. Using 63K prompts from COCO that we parsed with Stanza (Qi et al., 2020), we filtered for instances with two nouns per prompt and analyzed

	Antecedent first		Antecedent second	
Model	1st noun	2nd noun	1st noun	2nd noun
DF (T5)	50.8%	33.87%	35.50%	51.60%
SD (Clip)	58.4%	23.80%	54.90%	27.90%

Table 1: The percentage of prompts in each group where the antecedent noun (either the first or the second noun mentioned) appeared earlier.

the dependency relations between the nouns. We categorized the data based on the linear position of the antecedent and generated images with 40 random samples from each group. For each generation and intermediate layer, and each object X, we queried whether object X appears in the image.

Results. First, we sometimes observe a "race" between the nouns: in 11.9% of the cases in DF, the object that appears in an earlier layer disappears at a later layer, while the other object takes dominance. See Appendix A.2 for examples.

Second, Table 1 presents information on the order of generation for both models, revealing that the sequence in which objects emerge during the computation process is determined by either their linear or their syntactic precedence, *depending on the particular text encoder*. In DF's T5 text encoder, slightly over half of the instances feature the antecedent appearing at an earlier layer than the descendant, with a smaller fraction showing the opposite, and the rest indicating simultaneous appearances. This holds true regardless of linear order. Conversely, in SD's Clip, the first noun tends to appear before the second more frequently, irrespective of the syntactic role.

While the two models differ in multiple respects (architecture, pretraining data, training objective, and more), it is intriguing to observe that T5, trained on a language modeling objective, demonstrates a greater awareness of syntactic structure compared to Clip – a model trained to align pairs of prompts and images without a specific language modeling objective. This discrepancy points to a possible impact of training objectives on the models' representation building process.

5 Memory Retrieval

Text-to-image diffusion models are able to retrieve information of many concepts (Ramesh et al., 2022), encompassing entities like notable individuals, animals, and more. Memory retrieval—the recall of stored information—involves a constructive process rooted in the interactive dynamics between

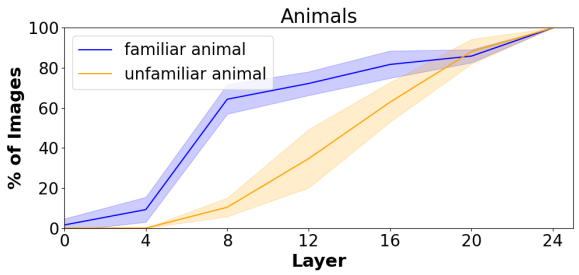

Figure 7: Familiar vs. unfamiliar animals across layers. Familiar animals emerge at much earlier layers.

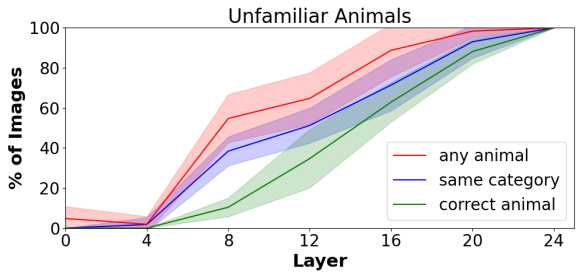

Figure 8: Subset of layers encoding different features in the process of unfamiliar animal generation.

memory trace features and retrieval cue characteristics (Smelser et al., 2001). In this section, we leverage the DIFFUSION LENS to scrutinize the memory retrieval mechanism in the text encoder.

5.1 Common and Uncommon Concepts

We investigate whether there is a difference in the generation process for prompts describing common and uncommon concepts. To this end, we collect a list of familiar and unfamiliar *animals*.[3] The classification criterion was derived from the average daily view statistics of Wikipedia pages spanning the period from October 2022 to October 2023. In particular, an animal was classified as "familiar" if it had an average of 1500 visitors per day on its Wikipedia page (e.g., a kangaroo), while one having fewer than 800 visitors per day was classified as "unfamiliar". See Appendix C for details.

We hypothesize that as the model might have seen the unfamiliar animals less frequently during training, it might take longer to generate these animals. Hence, we ask the annotators if the specific animal appears in the generated image for each prompt describing the animal.

Results. As summarized in Figure 7, *common concepts emerge early*, as early as layer 8 out of 24. In contrast, *uncommon concepts gradually become apparent across the layers*, with the diffusion model generating accurate images primarily at the top layers.

5.2 Gradual Retrieval of Knowledge

To delve deeper into the knowledge retrieval process, we pose additional questions for each prompt and generated image of unfamiliar animals: (a) Is there an animal in the image? (b) Does the image feature an X, where X represents the informal "category" of the animal, like "mammal", "bird", etc.?

[3]We encountered objects or individuals that were unfamiliar, but the models struggled to generate them effectively.

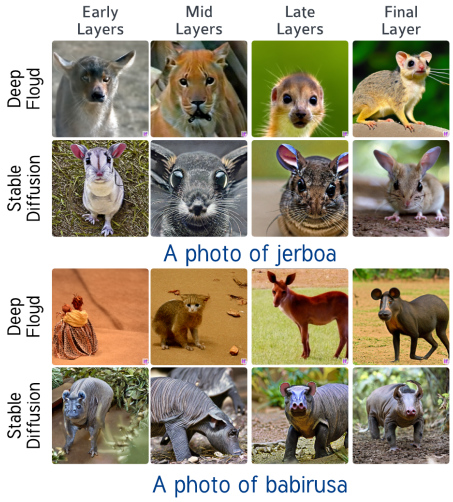

Figure 9: The difference between DF and SD in knowledge retrieval of animals.

[4] (c) Does the image depict the exact animal in the prompt?

Results. Figure 8 illustrates *incremental knowledge extraction*, beginning with a general animal, progressing to a more specific animal within the same category, and reaching a representation of the particular animal mentioned in the prompt.

Though the plot for SD reveals a similar pattern (Appendix E), qualitative analysis reveals *distinct knowledge retrieval patterns between the two models*: In the case of DF's T5, knowledge retrieval is gradual, unfolding as computation progresses (Figure 9). Layers generate animal, mammal, and ultimately construct a representation of the specific animal. In contrast, SD's text encoder, Clip, does not exhibit a similar progression, as retrieval appears less incremental. The model seems to establish the representation in a less gradual manner: The first layer with a meaningful image already closely resembles the final animal, with subsequent layers

[4]We chose to use an informal taxonomy because the animal kingdom taxonomy is a complex subject that is under research and debate, and its terms are not familiar to the general population – which suggests that it also less present in the T2I training data.

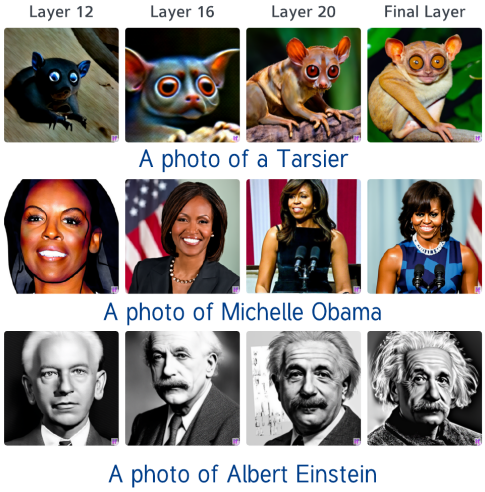

Figure 10: Intricate details are refined gradually.

primarily refining features. These differences echo the syntactic findings in Section 4.2. They suggest that pretraining objectives, data, or model architecture might influence information organization, leading to distinct memory retrieval patterns.

5.3 Gradual refinement of features

As the computation progresses, both accuracy and realistic representation significantly improve with refining details at each step. This progression is evident in Figure 10 (top row), as seen in the gradual refinement of the "Tarsier" image. A similar trend occurs in the representation construction of human subjects, with facial features undergoing refinement for a more faithful portrayal (Figure 10, rows 2+3). To systematically assess this phenomenon, we compiled a list of 30 celebrities, using DIFFUSION LENS to generate images from intermediate representations in the text encoder. For each prompt and generated image, we ask: (a) Is there a person in the image? (b) Does the person align with the celeberity's (self-identified) gender? (c) Does the person exhibit the celebrity's style (hair, clothing, etc.)? (d) Is the individual in the image distinctly recognizable as the specified celebrity based on facial features?

Results. Figure 11 quantifies the *step-by-step construction of the representation*, culminating in its maximum resemblance to the celebrity. The integration of distinct features follows a hierarchical pattern, progressing from broad characteristics (such as the overall human form) to finer details (specifically, facial features), which become evident only in the final layers.

Discussion The results in this section regrading the gradual retrieval and refinement of knowledge suggest an alternative perspective on how knowledge is encoded in language models. This viewpoint is different from recent work suggesting that models utilize a key–value memory structure, where facts are local to specific layers (Geva et al., 2022; Meng et al., 2022). Our results indicate that some information is distributed across layers, allowing for a gradual retrieval of knowledge rather than a retrieval at a particular point in the model. This aligns with earlier research proposing hierarchical representations in vision models (Zeiler and Fergus, 2014; Zhou et al., 2014; Bau et al., 2017).

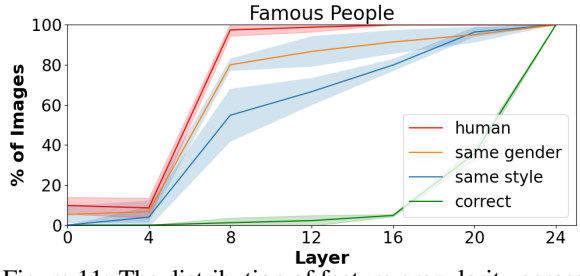

Figure 11: The distribution of feature granularity across layers in generated images.

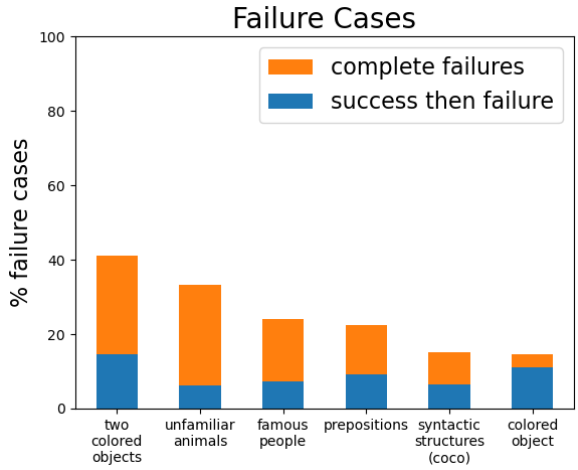

Figure 12: Many cases display successful generations from earlier layers before turning into failures.

6 Error Analysis

In the previous sections, we analyzed the computation process of the text encoder in success cases. In this section we briefly discuss insights about the computation patterns in failure cases, that is, cases where the image generated from the final layer does not faithfully capture the prompt. Figure 12 shows the percentage of failures for each experiment that had over 10 failures. We split failures to two types: *complete failures* when no layer generated a correct image through DIFFUSION LENS, and cases when at least one layer generated a correct image, but the top layer led to a failure (*success then failure*).

Figure 13: DIFFUSION LENS reveals a correct image generation at a middle layer, while the final image fails to fully represent the prompt.

Generally, the percentage of failure cases (total height of each bar) is low, from 10% to 25% for most categories. Prompts about two colored objects have a higher failure rate. Importantly, in many failure cases, the representations in earlier layers lead to a correct generation via our method. Notably, in simple prompts (relations and colored objects), about 80% of the failures had successful generations at earlier layers. See Figure 13 for an example. Once more constraints are imposed (two colored objects), we have a lower rate of early success. Finally, for knowledge-related tasks (famous people, uncommon animals), there are very few cases of early success turned to failure. Presumably, when the model fails, it is mostly because it does not encode the information at all.

7 Related Work

Interpreting language models. A wide range of work has analyzed language model internals. We briefly mention a few directions and refer to existing surveys (Belinkov and Glass, 2019; Rogers et al., 2020; Madsen et al., 2022). Many studies employ an auxiliary model, like a probing classifier, to analyze whether internal representations correlate with external properties (e.g., Ettinger et al., 2016; Hupkes et al., 2018). However, this approach suffers from various flaws (Belinkov, 2022). Others employ interventions in representations, measuring how they impact a model's prediction (e.g., Vig et al., 2020; Elazar et al., 2021; Meng et al., 2022). Interventions allow making powerful claims but are tricky to design (Zhang and Nanda, 2023) and often restricted to narrow use cases.

Another influential approach is the Logit Lens (nostalgebraist, 2020), which projects intermediate representations of language models onto a probability distribution over the vocabulary space. This projection captures the internal computation, reflecting the model's gradual estimation of likely next words and the transfer of information across modules (Geva et al., 2022; Katz and Belinkov, 2023; Pal et al., 2023). Recent extensions to the Logit Lens learn a projection to aid with the representational drift between the intermediate layers and the final output, or to shortcut calculations (Belrose et al., 2023; Din et al., 2023). This line of work has focused on auto-regressive decoder language models. Inspired by this idea, we propose to use the diffusion module in T2I pipelines to visualize intermediate representations of the prompt and thus reveal the computation process in the text encoder. This approach renders an intermediate layer directly observable.

Interpreting vision–language models. Compared to unimodal models, research on interpretability in multimodal vision–language models is rather limited. Goh et al. (2021) found multi-modal neurons responding to specific concepts in CLIP (Radford et al., 2021), Gandelsman et al. (2023) decomposed CLIP's image representations into text-based characteristics, and Tang et al. (2023) analyzed the influence of input words on generated images via cross-attention layers in T2I pipelines. Chefer et al. (2023b) decomposed textual concepts, focusing on the diffusion component. In contrast, our work investigates the under-explored text encoder in T2T pipelines.

8 Discussion and Conclusion

This paper introduces DIFFUSION LENS, a novel method to analyze language models within T2I pipelines. Our approach deconstructs the T2I pipeline by examining specific sub-block outputs, offering a deeper insight into language-to-visual concept translation. We showcased the method's potential by analyzing two open-source text encoders with a pre-trained image diffusion model across diverse topics.

While we focused on the overall output of each block, our approach paves the way for visualizing individual sub-block outputs. Our application centered on T2I pipeline text encoders. Extending it to other language models, while non-trivial, offers a promising direction for future research. Our experiments also showed differing knowledge extraction patterns among text encoders, prompting further exploration on the impact of architecture, pretraining data, and objectives.

We hope that our method will be a valuable tool for the community. Integrating DIFFUSION LENS into the development and research pipelines offers new opportunities for exploring broader practical questions, such as biases and failure cases.

Limitations

In addition to the limitations we stated in the conclusion, our analysis was restricted due to the limited availability of open-source models. Future work will benefit from increased diversity models. Moreover, the set of prompts used in most of our experiments were automatically generated. However, this limitation provided an opportunity to meticulously isolate and investigate specific effects.

Ethics Statement

In this work, our primary objective is to enhance the transparency of text-to-image models. While not the focus our analyses, the DIFFUSION LENShas the potential to unveil biases within these models. We anticipate that our work will contribute positively to the ongoing discourse on ethical practices in text-to-image models. At present, we do not foresee major ethical concerns arising from our methodology.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics (TACL)*, 7:49–72.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. 2023a. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10.

Hila Chefer, Oran Lang, Mor Geva, Volodymyr Polosukhin, Assaf Shocher, Michal Irani, Inbar Mosseri, and Lior Wolf. 2023b. The hidden language of diffusion models. *arXiv preprint arXiv:2306.00966*.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. Jump to conclusions: Short-cutting transformers with linear transformations. *arXiv preprint arXiv:2303.09435*.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. 2023. Interpreting clip's image representation via text-based decomposition. *arXiv preprint arXiv:2310.05916*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. 2021. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

James Hampton. 2013. Conceptual combination 1. In *Knowledge Concepts and Categories*, pages 133–159. Psychology Press.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program.

In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2023. Prompt-to-prompt image editing with cross-attention control. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip. If you use this software, please cite it as below.

Shahar Katz and Yonatan Belinkov. 2023. Visit: Visualizing and interpreting the semantic information flow of transformers. *Findings of The 2023 Conference on Empirical Methods in Natural Language Processing*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Ling ling Wu and Lawrence W. Barsalou. 2009. Perceptual simulation in conceptual combination: Evidence from property generation. *Acta Psychologica*, 132(2):173–189. Spatial working memory and imagery: From eye movements to grounded cognition.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NeurIPS*.

nostalgebraist. 2020. interpreting gpt: the logit lens. lesswrong, 2020.

OpenAI. 2023. Gpt-4 technical report.

Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. 2023. Editing implicit assumptions in text-to-image diffusion models. *arXiv preprint arXiv:2303.08084*.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560, Singapore. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu

Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494.

Neil J Smelser, Paul B Baltes, et al. 2001. *International encyclopedia of the social & behavioral sciences*, volume 11. Elsevier Amsterdam.

StabilityAI. 2023. Deepfloyd if.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. 2023. What the DAAM: Interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5644–5659, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS, Spotlight presentation)*.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. 2022. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yutaro Yamada, Yingtian Tang, and Ilker Yildirim. 2022. When are lemons purple? the concept association bias of clip. *arXiv preprint arXiv:2212.12043*.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham. Springer International Publishing.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning deep features for scene recognition using places database. *Advances in neural information processing systems*, 27.

A Additional Results

A.1 Prepositions

We explore prepositions in prompts. We investigate how prompts, including certain relations, affect the generation process. These prompts are complex, challenging the compositional understanding of the T2I model. In particular, we examine the prepositions "on" and "in". Figure 14 illustrates the percentage of images that correctly generated the concepts for two categories: objects alone and objects with specified relation. Our findings reveal that prompts involving only one of the objects tend to perform well in the early layers of the model. However, more intricate prompts, including both objects and a relational context, emerge only in later layers of the model.

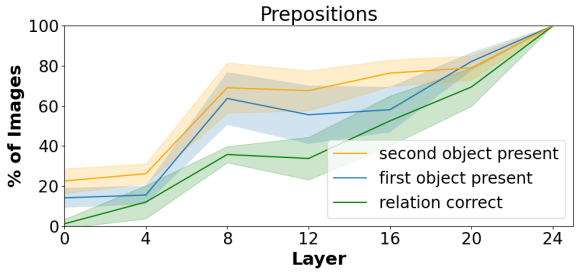

Figure 14: The proportion of images where either the objects, or objects with prepositions, were accurately represented.

A.2 Race between objects

Figure 16 presents examples of "race" between the objects in the prompts: one object appears first, and then disappears at a later layer to make room for the other object, before finally emerging again in the top layers.

A.3 Final layer norm necessity

In the DIFFUSION LENS process, we pass the output of block l through the last layer norm ln_f. However, we examine the option to bypass the ln_f layer and directly connect to the components of the diffusion model. As Figure 17 demonstrates, images generated without the final layer normalization are meaningless. The final layer norm thus plays a crucial role in generating meaningful images. It highlights the necessity of the ln_f layer within DIFFUSION LENSpipeline. A similar finding has been observed in the LogitLens (nostalgebraist, 2020) and TunedLens (Belrose et al., 2023).

B Annotation Process

The results in this paper rely on human annotators to determine the presence of different concepts in the generated images. We employed a team of ten professional full-time annotators using the Dataloop platform , in accordance with institutional regulations. The annotator teams was based in India, and were paid a rate of 8 USD per hour, in accordance with laws in India.

Each annotator received the instructions in Figure 18. The annotators were given the instruction to be liberal towards a positive answer. We manually validated each question, making sure the concepts in the question are not abstract (e.g., "beautiful"), and that the answer should be clear for each case. For each experiment, we duplicate 10% of the images, and ask an additional annotator the same questions, used to calculate inter annotator agreement. For experiments containing rare animals and celebrities, annotators were given reference images from google.

We provide our main results based on the human annotations. We chose to use human annotations since the existing automatic methods are limited. CLIP as an image classifier was shown to fail when required to explicitly bind attributes to objects (Ramesh et al., 2022; Yamada et al., 2022), and exploratory experiments we performed with BLIP (Li et al., 2022) showed similar issues.

However, we found a high agreement between GPT-4V and the human annotators on most tasks and questions, as shown in Table 2. For one experiment – two colored objects – we found a high variance using the human annotations and thus extended it to further annotations using GPT4-V.

C Animals Experiment: Implementation Details

C.1 Animal classes used

To measure the gradual knowledge retrieval, one of the questions we ask in the experiment on unfamiliar animals is whether the image contains an animal of class X, where we vary X according to an informal, popular taxonomy that the specific animal belongs to. Note that although it does not faithfully represent the scientific view on the animals we generate, it is more suitable to observe a model that was trained on data that was taken from the wide internet.

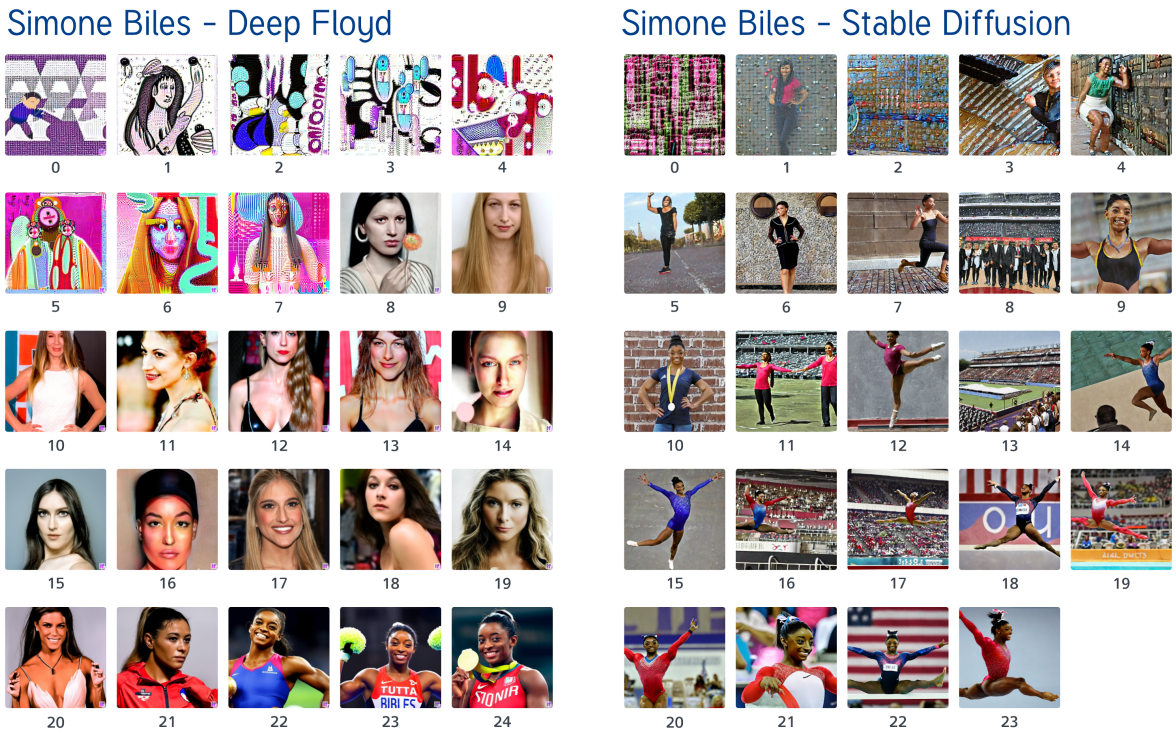

Figure 15: Example generations from all layers

Figure 16: A sequential "race" between two objects in the sentence, where one initially appears before the other, only to subsequently vanish and make room for the latter object.

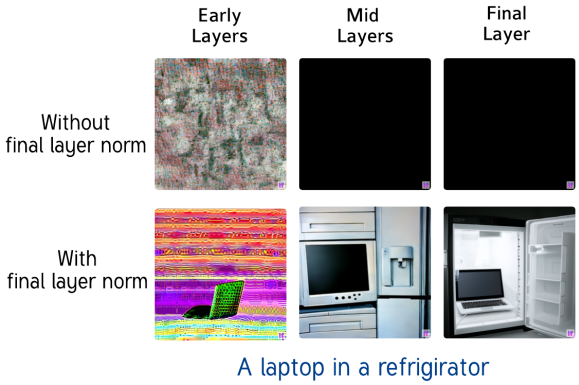

Figure 17: Example generations from DIFFUSION LENS with and without the final layer norm.

C.2 The full list of animals

Familiar animals: Beagle, German Shepherd, Labrador Retriever, Dachshund, Bulldog, Ragdoll, Kangaroo, Chicken, Owl, Eagle, Salmon, Catfish, Cod, Orca, Komodo dragon, King cobra, Platypus, Narwhal, Ostrich, cougar.

Unfamiliar animals: Aye-aye, Dik-dik, Tarsier, Gerenuk, Jerboa, Babirusa, Saola, Galago, Vervet, guppy, Celestial Pearl Danio, Herring, Pike, Walleye, Grebe, Spoonbill, Bee-eater, Taipan, ,Copperhead, Anilius, Skink, Bearded Dragon, Ladybug, Scarab, Blue morpho, Cloudless sulphur, Giant anteater

D Implementation Details

We implemented our code using Pytorch (Paszke et al., 2019) and Huggingface libraries (Wolf et al., 2020; von Platen et al., 2022). For each experi-

	Inter annotator agreements			Agreements with automatic annotations		
Question type	#annotations	f1	cohen's kappa	#annotations	f1	cohen's kappa
One object presence (out of 2)	416	72.5%	48.2%	1381	80.6%	63.8%
Relation correct	208	73.7%	61.4%	1319	81.3%	70.1%
One Color presence	208	76.9%	60.7%	1671	85.3%	85.9%
Familiar animals presence	52	94.7%	87.2%	789	85.5%	67.2%
Unfamiliar animals presence	104	84.6%	81.3%	1019	84.3%	72.4%
Unfamiliar animals class presence	260	73.2%	59.5%	1012	91.2%	81.3%
Syntactic structures correct (coco)	357	80.6%	69.7%	2962	80.0%	59.5%

Table 2: A table of agreement between human annotators (left) and between human and automatic annotations averaged over both models. Overall, we see a high agreement between the human annotators and between the human and automatic annotations.For human agreement - the lowest Kappa score is for one object presence, probably due to the ambiguity in early layers, where there is a mix of both objects. For example in fig 5, second line, layer 12.

On this project, you will have to annotate sets of 50 images. For each set, you will have a yes or no question. The questions are written at the start of each task name. They end with a "?". The latter part of the name is in "[ ]" and is not relevant for the questions. For convenience, we start the question with the statement itself, therefore "dog in the image?" means "Is there a dog in the image?" The questions vary from simple questions like "Is there a dog in the image?" to more complicated questions like "Is there a red bird on a green boat?". The images are generated by AI, and might not be realistic. You should answer if the image might be interpreted as the question asks. Examples at the end of this file.

Figure 18: Annotation guidelines.

ment, we generated four images (different seeds) for each layer, and we report the standard division over the seeds in all plots. We use Stable Diffusion v2-1 (CreativeML Open RAIL++-M License) (Rombach et al., 2022) and Deep Floyd (DeepFloyd-IF-License) (StabilityAI, 2023). We ran the experiments on the following GPUs: Nvidia A40, RTX 6000 Ada Generation, RTX A4000 and GeForce RTX 2080 Ti.

Our code is available in the supplementary material.

D.1 Dependency parsing implementation

We conducted a syntactic structure analysis using Stanza (Qi et al., 2020), a Python package. Stanza provides tools for obtaining parts of speech (POS) and syntactic structure dependency parse. To perform this analysis, we executed a Stanza pipeline designed for English. This pipeline returns the tokenized form, POS, lemmatization, and syntactic dependency parsing for a given prompt. We didn't customize any additional parameters and utilized the default settings during the analysis.

E Results on Stable Diffusion

To complement the results in the main paper, we provide Figures A.1, 19–24 from Stable Diffusion.

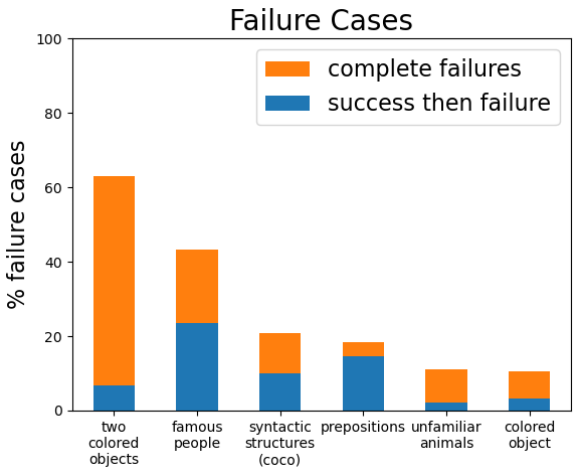

Figure 19: Many cases display successful generations from earlier layers before turning into failures.

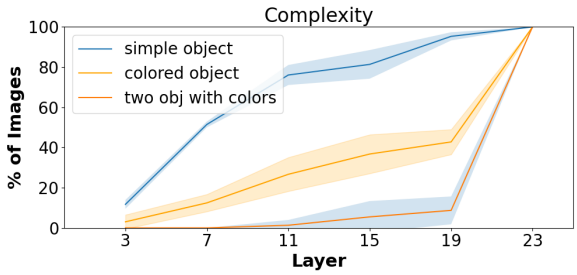

Figure 20: The percentage of images, from each category, for which the prompt matches the generated image, across different intermediate layers. As the prompt is more complex, it takes more layers for DIFFUSION LENS to be able to extract a correct image.

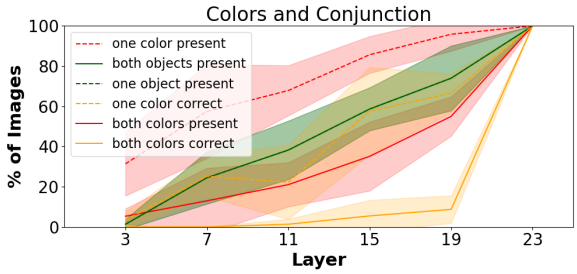

Figure 21: The proportion of images where either the object, the colors, or both were present, and where either the objects or the colors were accurately represented.

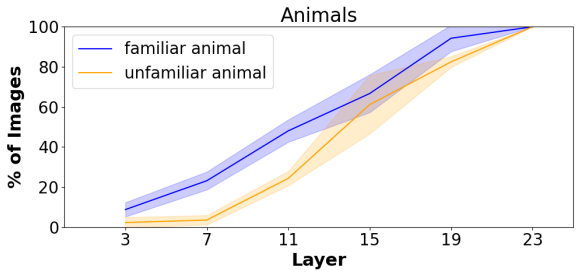

Figure 22: Familiar vs. unfamiliar animals across layers. Familiar animals emerge in much earlier layers.

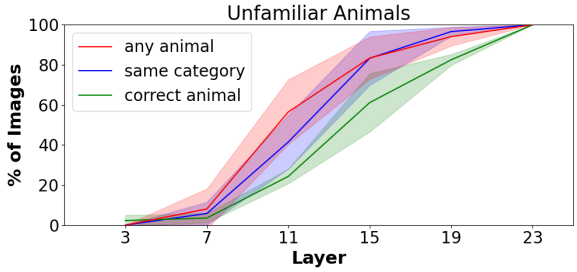

Figure 23: Subset of layers encoding different features in the process of unfamiliar animal generation.

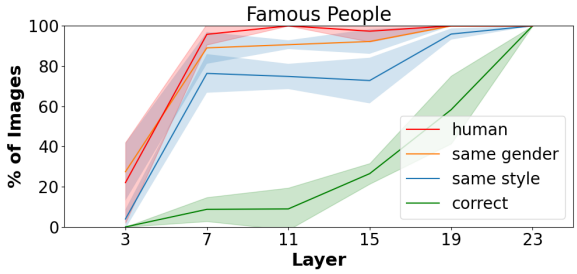

Figure 24: The distribution of feature granularity across layers in generated images.

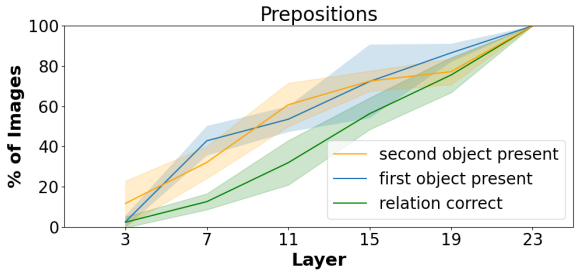

Figure 25: The proportion of images where either the objects, or objects with prepositions, were accurately represented.