Understanding Visual Concepts Across Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large multimodal models such as Stable Diffusion can generate, detect, and classify new visual concepts after fine-tuning just a single word embedding. Do models learn similar words for the same concepts (i.e. `<orange-cat>` = orange + cat)? We conduct a large-scale analysis on three state-of-the-art models in text-to-image generation, open-set object detection, and zero-shot classification, and find that new word embeddings are model-specific and non-transferable. Across 4,800 new embeddings trained for 40 diverse visual concepts on four standard datasets, we find perturbations within an ϵ-ball to any prior embedding that generate, detect, and classify an arbitrary concept. When these new embeddings are spliced into new models, fine-tuning that targets the original model is lost. We show popular soft prompt-tuning approaches find these perturbative solutions when applied to visual concept learning tasks, and embeddings for visual concepts are not transferable. Code for reproducing our work is available at: anonymous-visual-words.github.io.

1 Introduction

Fine-tuning prompts is a widely successful technique for adapting large pretrained models to new tasks from limited data [20, 23, 41, 9]. In language modelling, these prompts can efficiently teach pretrained language models specialized tasks, such as reading tables [23]. In text-to-image generation, they can embed subjects with unique, often hard-to-describe appearances into the generations of a diffusion model [9, 38]. Large multimodal models, such as Stable Diffusion [37], OWL-v2 [30], and CLIP [34, 3, 8], can generate, detect, and classify diverse visual concepts not present in their training data after fine-tuning just a single word embedding representing that concept in their prompt [43]. *Do these models learn similar words for the same visual concept?* There is an emerging hypothesis in multimodal machine learning that text-based models learn to process visual information [25, 18], and acquire similar representations for visual information [13, 27], despite training purely on text. This investigation aims to determine if the hypothesis extends to visually-grounded soft prompts, and whether these prompts converge to a solution that can be re-used by other models, akin to Figure 1.

For example, do text-based models that can generate, detect, and classify various species of cats learn similar words for orange cats (i.e. `<orange-cat>` = orange + cat)? We conduct a large-scale analysis on three state-of-the-art models in text-to-image generation, open-set object detection, and zero-shot classification, and find that new word embeddings are model-specific and non-transferable. We optimize 4,800 new embeddings for Stable Diffusion [37], OWL-v2 [30], and CLIP [34, 3, 8] to generate, detect, and classify 40 diverse visual concepts in four standard datasets with high fidelity. Interestingly, `<orange-cat>` $\neq$ orange + cat for any model and concept tested. Instead, for all tested models we find perturbations within an ϵ-ball to any prior embedding that generate, detect, and classify an arbitrary visual concept. We refer to this behavior as **fracturing** of the embedding space. Fractured models have several noteworthy properties. First, their prompts are difficult to interpret: prompts for orange cats may be close to prompts for blue cars, and far from prompts for black cats. Second, their prompts are not transferable: when embeddings trained for one model are spliced into the prompt of a new model, the second model ignores fine-tuning that targets the original model.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

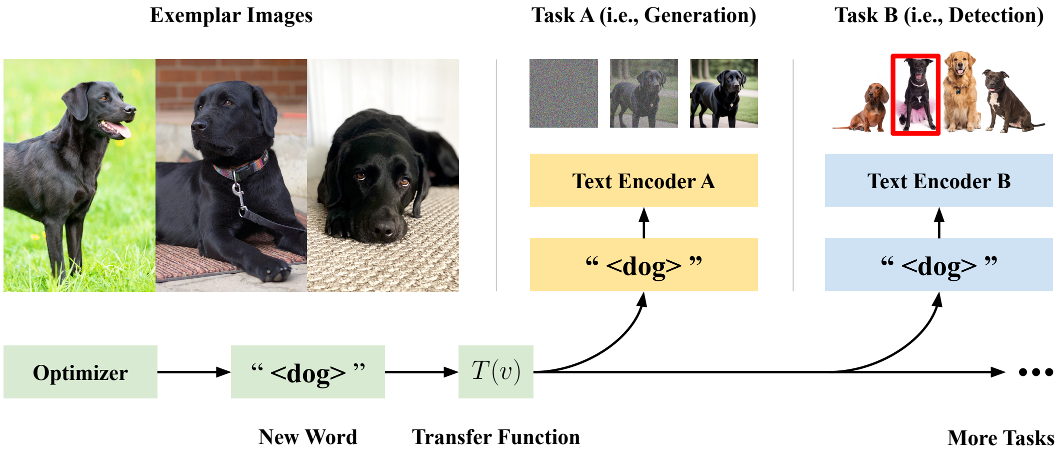

Figure 1: Large multimodal models can learn new words that represent specific concepts, like `<black-dog>` for the black Labrador retriever on the left in the figure. Do models learn similar words for the same concept? We study the interoperability of new word embeddings that encode visual concepts across three models and tasks, and show that popular soft prompt-tuning approaches find model-specific and non-transferable solutions.

2 Transfer Evaluation Methodology

Finding words for visual concepts across models involves finding a map between the embedding spaces of different models. We call this mapping the Transfer Function $T(v)$, depicted in Figure 1. The goal of the Transfer Function is to map word representations for visual concepts from the vector space $\mathcal{X} = \mathbb{R}^{d_x}$ for word embeddings in one model, to the vector space $\mathcal{Y} = \mathbb{R}^{d_y}$ for word embeddings in another model. $\mathcal{X}$ may correspond to Stable Diffusion [37] word embeddings for a generation task, and $\mathcal{Y}$ may be OWL-v2 [30] word embeddings for a detection task. Given these vector spaces, the Transfer Function predicts the representation $\vec{x}(w)$ in the vector space $\mathcal{X}$ for a word w originally from the vector space $\mathcal{Y}$ given just the word vector representation $\vec{y}(w)$.

$$T^{\,y\rightarrow x} : \mathcal{Y} \rightarrow \mathcal{X} = \arg\min_{T} \; \mathbb{E}_{w\sim p_w} \|\vec{x}(w) - T(\vec{y}(w))\|_2^2 \tag{1}$$

The Transfer Function $T(v)$ minimizes the average prediction error between transferred word embeddings $T(\vec{y}(w))$ and real word embeddings $\vec{x}(w)$ from the vector space $\mathcal{X}$. We average this prediction error over a uniform distribution p_w of the words that exist in both vector spaces $\mathcal{X}$ and $\mathcal{Y}$. In our experiments on Stable Diffusion 2.1 [37] and OWL-v2 [30], the number of words in p_w is large (> 40,000), much larger than the number of components d_x and d_y in each vector space.

2.1 Evaluating Words On Transferred Tasks

Using Equation 4, we estimate Transfer Functions between all six ordered subsets of three state-of-the-art models, and evaluate words optimized for visual concepts on one task (such as generation), and transferred to the same visual concepts on another task (such as classification). Consider a dataset D of images I depicting a specific visual concept, such as a black Labrador retriever, and task-specific annotations a_y, such as bounding boxes ($a_y \in \mathbb{R}^{b\times 4}$), or class labels ($a_y \in \mathbb{N}$). We first optimize word vector embeddings $\vec{v}_y \in \mathcal{Y}$ to minimize a task-specific loss function $\mathcal{L}_y$. We then zero-shot transfer $\vec{v}_y$ to task x using the linear map $\vec{v}_x = T^{\,y\rightarrow x}\vec{v}_y$, and evaluate a task-specific performance metric $\mathcal{M}_x$. Loss functions and performance metrics used for each task are shown in Table 1.

$$\mathbb{E}_{I,a_x\sim D_{\text{test}}} \; \mathcal{M}_x(T^{\,y\rightarrow x}\vec{v}_y, I, a_x) \;\; \text{s.t.} \;\; \vec{v}_y = \arg\min_{\vec{v}} \; \mathbb{E}_{I,a_y\sim D_{\text{train}}} \; \mathcal{L}_y(\vec{v}, I, a_y) \tag{2}$$

We use standard loss functions and performance metrics adapted from recent literature when training and evaluating words optimized for visual concepts. Each loss function and performance metric is discussed further in Section I. Now equipped for training, evaluating, and transferring words across models, we can ask our motivating question: *do models learn similar words for the same concepts?*

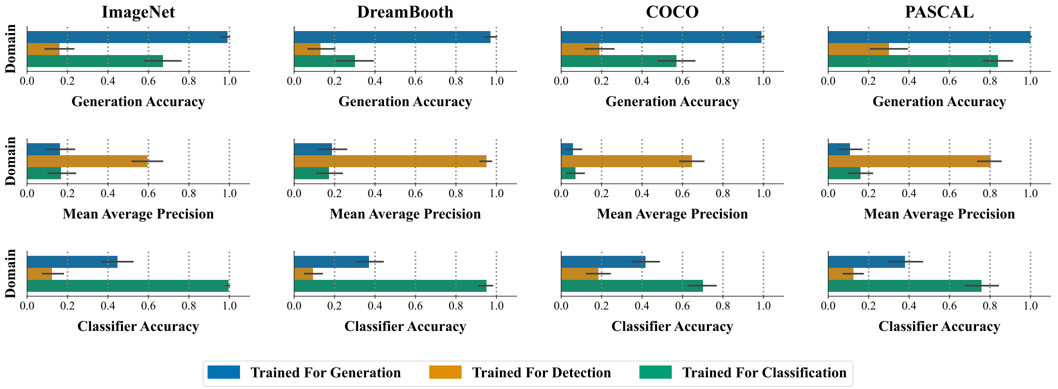

Figure 2: Visual word embeddings trained for one task (i.e. generation) perform well on that task, but may not perform well when transferred to another task (i.e. generation → detection). In certain directions, such as classification → generation, transfer works better than others. To understand when transfer fails, we perform extensive ablations across four standard datasets, and three models in generation, detection, and classification.

3 Soft Prompts Are Model-Specific

Prompts optimized for visual tasks can perform great in-domain, but are typically not re-usable. In most transfer scenarios in Figure 2, words optimized for one task can't solve a different task than they were trained on with comparable fidelity to in-domain training. Words optimized for classification transfer best, achieving up to 84% of the performance of in-domain training for generation (PASCAL), and up to 28% of the in-domain detection performance (ImageNet). Words optimized for detection are least transferable, achieving up to 26% of in-domain classification performance (COCO), and up to 30% of in-domain generation performance (PASCAL). Words optimized for generation are in the middle in terms of their transferability, attaining up to 59% of the in-domain classification performance (COCO), and up to 28% of the in-domain detection performance (ImageNet). Generation shows a significant difference in performance between words transferred from classification vs. detection, what's happening here?

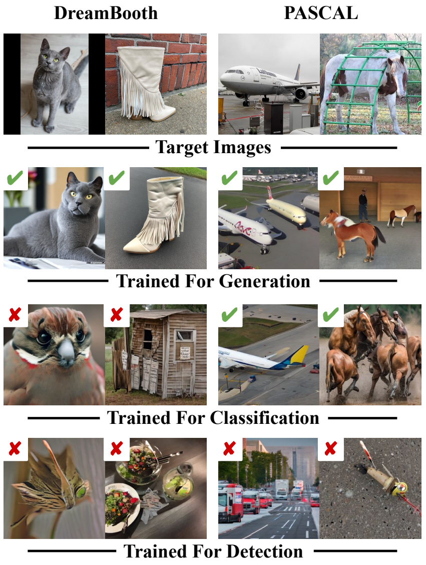

Figure 3: Generations (rows 2-4) from Stable Diffusion for target concepts (top row) from the DreamBooth and PASCAL datasets. The second row trains word embeddings for generation. The third row transfers word embeddings from classification to generation. The final row transfers from detection.

Understanding The Results Using generation as a case study, we show images generated by Stable Diffusion 2.1 in Figure 3 using prompts trained for generation (second row), transferred from classification (third row), and from detection (fourth row). We select two fine-grain concepts from the DreamBooth dataset, and two common concepts from the PASCAL dataset. Prompts trained for generation succeed at learning both fine-grain details for subjects in the DreamBooth dataset, and common classes in PASCAL. Prompts trained for classification miss fine-grain details, but learn common classes. Prompts from detection miss fine-grain details, and common classes when transferred to generation, explaining trends in Figure 2.

4 The Embedding Space Is Fractured

Results in Section 3 show that most embeddings become random when transferred. We explore this phenomenon by considering a constrained objective for soft prompts in Equation 3, where given an anchor word w_{anchor} that we initialize $\vec{v}$ to, and a threshold δ, we constrain solutions for $\vec{v}$ to an l2-ball of radius δ using projected gradient descent. Transfer and evaluation remain the same as discussed in Section I. We conduct a large-scale experiment, optimizing 4,800 prompts for 40 visual concepts

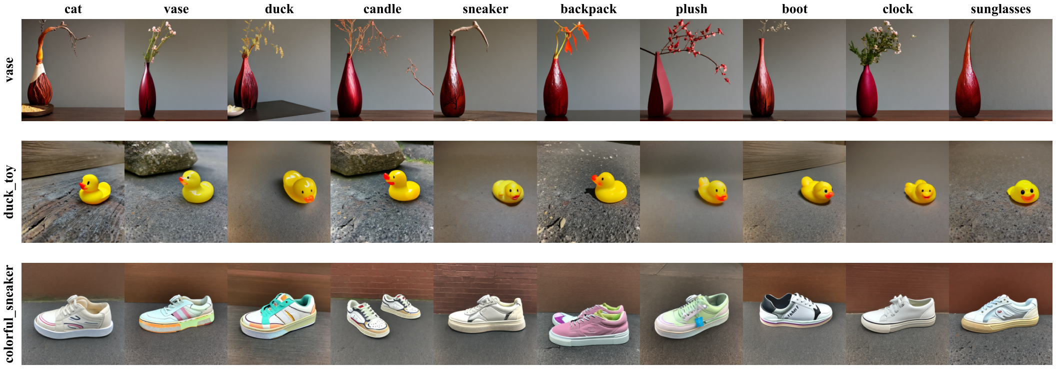

Figure 4: Example generations from Stable Diffusion 2.1 [37] for various concepts (row labels) using solutions found in the immediate neighborhood of unrelated words (column labels). We consistently find new words for generating arbitrary concepts near unrelated anchor words across DreamBooth (first three rows), ImageNet, COCO, and PASCAL VOC (examples in Appendix N). In several cases, near-identical images are generated by the diffusion model from two distinct prompts constrained to unrelated words in the embedding space.

across four standard datasets, three models, and four constraint thresholds $\delta \in \{0.1, 0.2, 0.5, 1.0\}$.

$$\vec{v}_y = \arg\min_{\vec{v}} \; \mathbb{E}_{I,a_y \sim D_{\text{train}}} \; \mathcal{L}_y(\vec{v}, I, a_y) \;\; \text{s.t.} \;\; \frac{\|\vec{v} - y(w_{\text{anchor}})\|_2}{\min_{w \neq w_{\text{anchor}}} \|y(w) - y(w_{\text{anchor}})\|_2} \leq \delta \tag{3}$$

This experiment controls where solutions are located in the embedding space, to help us understand the relationship between their location, and what gets transferred. Equipped with this tool, we can ask *where performant solutions are located, and why some solutions transfer better than others*.

4.1 Performant Solutions Are Everywhere

Near the representation for any word in embedding space, there is a perturbation ϵ that causes models to generate, detect, and classify an arbitrary unrelated visual concept. For example, the representation in the top-left of Figure 4 is closest to the cat vector, but Stable Diffusion generates a red vase. This behavior is consistent across three tested models, four standard datasets, and 40 diverse visual concepts, suggesting it may be a general phenomenon in Large Multimodal Models. We name this phenomenon **fracturing** of the vector embedding space, as the set of word vectors that encode (i.e. generate) an arbitrary visual concept is disconnected, and parts of the set are close to every anchor word tested. Examples of these solutions are shown in Figure 4, where each row corresponds to a visual concept from a standard dataset, and each column represents an anchor word. In several cases, an *identical image* is generated by perturbations near two unrelated anchor words, such as generations for the duck concept (second row) for the vase (column two) and candle anchors (column four).

5 Discussion

This work contributes a large-scale study of word embeddings that encode specific visual concepts across generation, detection, and classification tasks. We provide a benchmark for training soft prompts on a diverse set of visual concepts, and evaluating their transferability across three models. We show that certain embeddings are transferable between certain models, such as common concepts on the PASCAL task that transfer from classification → detection. In the majority of cases, soft prompts for visual concepts are model-specific, and to understand why, we conduct a large-scale ablation, training soft prompts constrained to the immediate neighborhood of different anchor words. We show that initialization does not matter as performant solutions are located everywhere in the embedding space, and non-transferable solutions resemble perturbations akin to adversarial examples.

Our work aims to galvanize the interoperability of large multimodal models following Figure 1, allowing prompts trained for generating black Labradors to be re-used for detection, and other tasks. Transferring prompts can significantly improve the adaptability and cost of machine learning systems by eliminating the need to re-train prompts when new models are released. We highlight the difficulty of transferring soft prompts for current multimodal models, and study why transfer often fails.

References

[1] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017.

[2] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

[3] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[4] K. Clark and P. Jaini. Text-to-image diffusion models are zero shot classifiers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[8] A. Fang, A. M. Jose, A. Jain, L. Schmidt, A. Toshev, and V. Shankar. Data filtering networks, 2023.

[9] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[11] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[12] J. Ho and T. Salimans. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022.

[13] M. Huh, B. Cheung, T. Wang, and P. Isola. The platonic representation hypothesis, 2024.

[14] T. Ju, Y. Zheng, H. Wang, H. Zhao, and G. Liu. Is continuous prompt a combination of discrete prompts? towards a novel view for interpreting continuous prompts. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7804–7819. Association for Computational Linguistics, 2023.

[15] H. Kannan, A. Kurakin, and I. J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018.

[16] D. Khashabi, X. Lyu, S. Min, L. Qin, K. Richardson, S. Welleck, H. Hajishirzi, T. Khot, A. Sabharwal, S. Singh, and Y. Choi. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In M. Carpuat, M. de Marneffe, and I. V. M. Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3631–3643. Association for Computational Linguistics, 2022.

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[18] J. Y. Koh, R. Salakhutdinov, and D. Fried. Grounding language models to images for multimodal inputs and outputs. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17283–17300. PMLR, 2023.

[19] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[20] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics, 2021.

[21] A. C. Li, M. Prabhudesai, S. Duggal, E. Brown, and D. Pathak. Your diffusion model is secretly a zero-shot classifier. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 2206–2217. IEEE, 2023.

[22] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu. BERT-ATTACK: adversarial attack against BERT using BERT. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6193–6202. Association for Computational Linguistics, 2020.

[23] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics, 2021.

[24] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

[25] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[26] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. *CoRR*, abs/2303.05499, 2023.

[27] K. Lu, A. Grover, P. Abbeel, and I. Mordatch. Frozen pretrained transformers as universal computation engines. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7628–7636. AAAI Press, 2022.

[28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[30] M. Minderer, A. A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby.

Simple open-vocabulary object detection with vision transformers. *CoRR*, abs/2205.06230, 2022.

[31] Z. Novack, J. J. McAuley, Z. C. Lipton, and S. Garg. Chils: Zero-shot image classification with hierarchical label sets. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 26342–26362. PMLR, 2023.

[32] P. Passigan, K. Yohannes, and J. Pereira. Continuous prompt generation from linear combination of discrete prompt embeddings. *CoRR*, abs/2312.10323, 2023.

[33] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli. Adversarial robustness through local linearization. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13824–13833, 2019.

[34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[35] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022.

[36] A. Robey, E. Wong, H. Hassani, and G. J. Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *CoRR*, abs/2310.03684, 2023.

[37] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.

[38] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 22500–22510. IEEE, 2023.

[39] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, R. Gontijo-Lopes, B. K. Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[40] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[41] T. Shin, Y. Razeghi, R. L. L. IV, E. Wallace, and S. Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics, 2020.

[42] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. R. Bach and D. M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015.

[43] B. Trabucco, K. Doherty, M. Gurinas, and R. Salakhutdinov. Effective data augmentation with diffusion models. *CoRR*, abs/2302.07944, 2023.

[44] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[45] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *CoRR*, abs/2302.03668, 2023.

[46] Z. Wu, Y. Wu, and L. Mou. Zero-shot continuous prompt transfer: Generalizing task semantics across language models. *CoRR*, abs/2310.01691, 2023.

[47] J. Yang, C. Li, P. Zhang, B. Xiao, C. Liu, L. Yuan, and J. Gao. Unified contrastive learning in image-text-label space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 19141–19151. IEEE, 2022.

[48] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3):24:1–24:41, 2020.

[49] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043, 2023.

[50] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Wang, L. Wang, J. Gao, and Y. J. Lee. Segment everything everywhere all at once. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

A Limitations & Safeguards

We employ pretrained diffusion models, object detectors, and classifiers in this work, and these models are known to have biases, obtained from their training data. Diffusion models in-particular can generate harmful or dangerous content, including graphic imagery of violence, and pornography. We employ the Stable Diffusion safety checker to flag generations after transferring soft prompts for unsafe content as a mitigation strategy for this potential limitation. Transferring soft prompts currently does not perform very well outside of certain common concepts, and one limitation of this paper is its scope: we do not propose new methodology for transferring soft prompts with high fidelity. Rather, we benchmark popular methods for soft prompt-tuning on three recent models, and show that most prompts are not transferable. Our experiments suggest that non-transferable prompts have certain properties that can be used to identify them, but turning this identification strategy into a mitigation method is outside the scope of this paper, and a challenge that we leave for future research.

B Ethical Considerations

Diffusion models currently require pristine data showing a subject in clear view in order to generate new photos of that subject. Transferring soft prompts from an object detector has the potential to allow for training on less pristine data that shows the subject amidst many distracting objects. One potentially harmful consequence of transfer between object detection models and generative models is related to privacy. Individuals that don't upload photos of themselves online are currently protected from their likeness being generated by diffusion models. However, transfer from object detectors to generative models would allow for their likeness to be generated, even when photos only show them in crowded spaces with many other people. Likewise, transferring prompts from generation to detection allows for the rapid creation of detectors for specific individuals. This technology could be used by malicious actors to track the activity of specific individuals, invading their privacy.

C Broader Impacts

Transferring prompts for specialized tasks significantly improves the adaptability and cost of machine learning systems by removing the need to re-train when new models are released. The cadence of multimodal machine learning is such that new models are released every month, and the state-of-the-art is in constant flux. Currently, soft prompts trained for older models are discarded when newer models are released, or when the task changes (i.e. classification becomes detection). Enabling the re-use of soft prompts would allow users to download prompts trained by someone else, like plugins, even when the original use-case for that soft prompt was for a different task (such as generation).

One negative broader impact that results from improved transferability is that soft prompts encoding negative and harmful behaviours become easier to use and maintain. Currently, harmful prompts become obsolete quickly as newer models are released, but once they can be transferred, they become permanent. Mitigation strategies for this risk could involve moderating online databases containing soft prompts to remove ones that perpetuate harmful behaviors, and filtering the outputs of models using the soft prompts to directly remove the harmful content (in the same vein as a safety checker).

D Related Works

Text-To-Image Generation. With the advent of diffusion-based architectures, large-scale generative models have developed impressive photo-realism. Approaches like Stable Diffusion [37], DALL-E 2 [35], and Imagen [39] employ diffusion-based approaches [11, 42] that start from an initial Gaussian noise map, and iteratively denoise the image over several denoising diffusion steps. These approaches incorporate pretrained text-encoders, such as CLIP [34] in Stable Diffusion [37], to guide generation in the diffusion process. Guidance is typically applied through Classifier-free Guidance [12], which allows the influence of the text-encoder to be increased, at the expense of generation quality. Diffusion models have remarkable flexibility, and can generate new subjects from a handful of examples by learning embeddings for pseudo tokens representing the subject in the prompt [9, 43]. Fine-tuning both the model and the prompt, as in Dreambooth [38], leads to improved generation of subjects, while retaining the controllability of pseudo tokens. These pseudo tokens for diffusion models are an instance of prompts encoding visual concepts, and our analysis applies to them.

Open-Vocabulary Object Detection. Parallel to work in generation, large-scale object detection models have developed a comparable strong versatility, and can detect new objects from short descriptions of their appearance (i.e. detect *black dog*) [50, 26, 30]. Models like Grounding DINO [26], SEEM [50], and OWLv2 [30] employ a pretrained text encoder to produce representations for classifying bounding boxes. In OWLv2 [30], representations from a pretrained CLIP [34] text encoder are contrasted with region-based representations from a vision transformer backbone. Grounding DINO [26], and SEEM [50] employ representations from a pretrained text encoder (BERT [6], and UniCL [47], respectively) to directly guide bounding box proposal. We show open-vocabulary object detectors can detect new objects from a handful of examples by optimizing a single new word embedding for the object in their prompt. Furthermore, our analysis shows many of the properties of these new words in open-vocabulary object detection are the same as for text-to-image generation.

Zero-Shot Classification. We use CLIP [34] for zero-shot classification. We insert new word embeddings optimized for classifying new visual concepts in the prompt of the CLIP text encoder, and contrast text representations with image representations from the CLIP vision encoder on test images. Prior work shows CLIP is an effective zero-shot classifier on open-vocabulary tasks [34, 31]. We use checkpoints from OpenAI CLIP [34], OpenCLIP [3], and Data Filtering Networks [8], trained on LAION-5B [40]. Diffusion models can also be used as zero-shot classifiers [21, 4], but we focus on CLIP for better task coverage. Our analysis shows that soft prompts learned for zero-shot classification share properties with open-vocabulary object detectors and text-to-image models.

Prompt-Tuning. The word embeddings we optimize for visual concept learning tasks are closely related to prompt-tuning [20, 23]. Prompt tuning aims to find a prefix or an entire prompt that causes a pretrained language model to perform a specialized task, such as reading tables [20, 23]. These methods treat the prompt as a trainable parameter, and optimize the embeddings of the prompt to minimize a task loss function. Prior work has shown the resulting soft prompts in language modelling tasks are hard to interpret [16], as their closest discrete prompts are often unrelated to the desired task. Transferring learned prompts is an important task in jail-breaking LLMs [49, 36], and researchers are searching over discrete prompts [45, 41, 49, 36]. In pure language modelling tasks, researchers have shown that certain soft prompts can transfer between models with the same architecture and task, but different weights [32, 14, 46]. We extend this investigation to visual tasks, and models with different architectures, trained on different label modalities (images, bounding boxes, and class labels).

Adversarial Examples. The perturbative structure of word embeddings that encode visual concepts are akin to an adversarial attack on the embeddings of text encoders. Adversarial robustness is an extensively studied field in computer vision [10], with a variety of attack methods, including [10, 1, 28, 19, 2], and defense methods, including [28, 33, 44, 15]. Adversarial attacks in computer vision traditionally focus on modifying the pixels in an image, whereas we modify word embeddings. Adversarial attacks on language are in their infancy, including jail-breaking approaches [49, 36], and typically involve searching over discrete prompts [48, 22], rather than continuous embeddings.

E Finding A Linear Transfer Function

Solving the optimization problem given by Equation 1 is hard in general, and to simplify the investigation, we restrict our focus to the class of linear Transfer Functions. This restriction transforms the hard problem in Equation 1 into a Linear Least Squares estimator, which has a closed-form solution. Consider a pair of matrices $X \in \mathbb{R}^{n \times d_x}$ and $Y \in \mathbb{R}^{n \times d_y}$, where each pair of rows in X and Y is a pair of word vector embeddings $\vec{x}(w)$ and $\vec{y}(w)$ for a word w contained in the support of the distribution p_w. The Linear Least Squares estimator we employ for $T^{\,y \rightarrow x}$ is given below.

$$T^{\,y \rightarrow x} = \arg\min_{T} \; \mathbb{E}_{w \sim p_w} \| \vec{x}(w) - T \vec{y}(w) \|_2^2 = (Y^T Y)^{-1} Y^T X \tag{4}$$

One can interpret the map $T^{\,y \rightarrow x}$ as lining up the directions in the vector spaces $\mathcal{X}$ and $\mathcal{Y}$ that correspond to the same visual concepts. Word embeddings often have algebraic relationships [29], and a linear Transfer Function preserves these relationships by distributing over addition.

Task	Loss Function	Performance Metric
Generation	$\mathbb{E}_{I \sim D_{\text{train}}} \, \|\epsilon - \epsilon_\theta(\sqrt{\alpha_t} I + \sqrt{1 - \alpha_t}\epsilon, t, \vec{v})\|^2$	$\mathbb{E}_{I \sim p_\theta(\cdot \mid \vec{v})} \; \mathbb{1}[\; I \text{ has the concept }]$
Detection	$\mathbb{E}_{I,b,w \sim D_{\text{train}}} \, [w \cdot (\vec{e}_{\text{object}}(I, b)^T \vec{e}_{\text{text}}(\vec{v}))]$	Mean Average Precision
Classification	$\mathbb{E}_{I,w \sim D_{\text{train}}} \, [w \cdot (\vec{e}_{\text{image}}(I)^T \vec{e}_{\text{text}}(\vec{v}))]$	Classifier Accuracy

Table 1: Loss Functions and Performance Metrics. We benchmark transfer of word embeddings for visual concepts across generation, detection, and classification. In each row, I corresponds to an image, b to an object bounding box, and $w \in \{-1, 1\}$ to a weight multiplied onto the loss function. This weight controls whether the objective is maximized or minimized, where $w = -1$ when the image and bounding box contain the target concept, and $w = 1$ otherwise. The functions $\vec{e}$ are image and text encoders that return vector representations: $\vec{e}_{\text{image}}$ is the CLIP vision encoder, $\vec{e}_{\text{text}}$ is the CLIP text encoder, and $\vec{e}_{\text{object}}$ is the OWL-v2 region feature proposer.

F Evaluation Metrics

For tuning prompts with Stable Diffusion 2.1 [37], we use the denoising loss function originally proposed in Ho et al. 2020 [11], where the goal is to predict a noise map ϵ added to an image I at a particular timestep in the diffusion process t. We optimize the word vector embedding $\vec{v}$ so that Stable Diffusion generates images of a particular class (such as black Labrador). This optimization uses a training dataset D_{train}, and a separate dataset D_{test} that contains different images of the same visual concept (such as black Labrador) is used for evaluation. For evaluating generation, we measure the probability that generations contain the target visual concept, measured by OpenAI's pretrained CLIP L-14 model given the prompt "a photo of {`visual_concept_name`}". We build on Textual Inversion [9] with a transfer step, shown in Figure 5. Losses and metrics are listed in table 1.

G Dataset Preparation

We employ the 2014 ImageNet detection dataset [5], the DreamBooth dataset [38], COCO [24], and PASCAL VOC [7]. For each dataset, we select 10 concepts uniformly at random from the available classes to use for benchmarking, and select 8 images per concept from the training set. See Appendix L These cover a wide range of concepts likely to be encountered in the wild. For ImageNet, each image is annotated with an integer class label, and a set of bounding boxes that contain the target concept. For the DreamBooth dataset, bounding box labels are missing. To obtain bounding box labels, we ran a pretrained OWL-v2 on every image using the name of the subject as the prompt, and manually verified the labels as correct. For COCO and PASCAL VOC, class labels are not present, so we assign each image a class label equal to the class of the largest bounding box.

H Model Details

We analyze three state-of-the-art models in text-to-image generation, open-set object detection, and zero-shot classification. Each model accepts a text-based prompt as input, containing the new word to be optimized (such as `<dog>` for the dog concept). For generation, we choose Stable Diffusion 2.1 [37], a latent diffusion-based generative model. For detection, we select OWL-v2 [30], a two-stage object detection model with a region proposal stage, and a classification stage that contrasts region features with text encodings of class names. For classification, we employ Data Filtering Networks [8], which use CLIP-based [34] contrastive training on a filtered dataset. These models have different input requirements. We resize images to 768x768 pixels when optimizing for Stable Diffusion [37], 960x960 for OWL-v2 [30], and 224x224 for Data Filtering Networks [8].

I Experiment Details

Training We take all combinations of models, datasets, and concepts, and perform 10 randomized trials, where we vary the initialization word used to seed the optimization algorithm. Initialization words are selected as the closest single token in the model's tokenizer to the name of a concept in the dataset. For example, 'sombrero' tokenizes to multiple subwords, so we use 'hat' for its initialization word. This choice ensures that our experiment accounts for both good and poor initializations. We

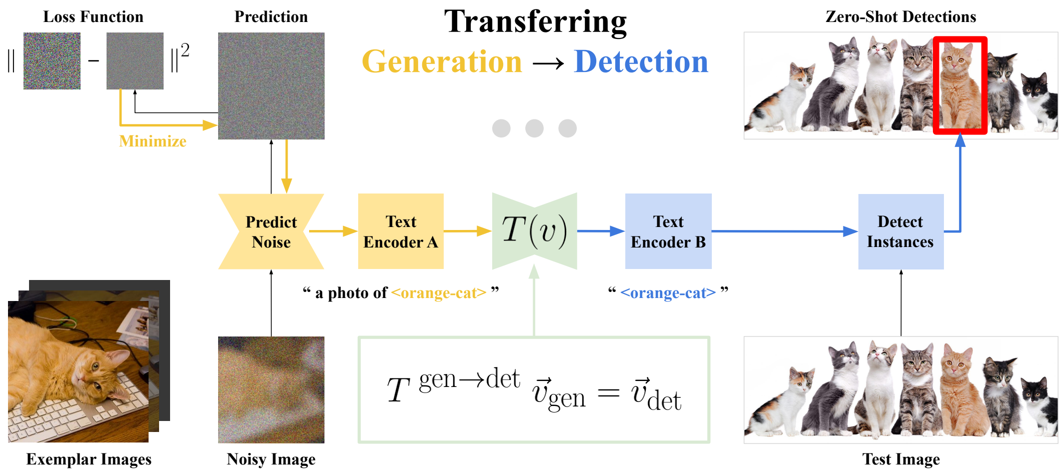

Figure 5: Transferring words optimized for generation to detection tasks. We fine-tune the vector embeddings for new words (such as `<orange-cat>` for the orange cat in the figure) to minimize a noise prediction loss for generation. Vector embeddings are transferred from generation to detection using the Transfer Function $T(\vec{v})$, and used to produce zero-shot instance detections for the target visual concept (in this case, orange cats).

optimize the embeddings for new word tokens using the Adam [17] optimizer with a learning rate of 0.0001, and a batch size of 8 (these hyperparameters are shared across all models). We train for 1000 gradient descent steps, and report final performance metrics using the optimized word embedding.

Loss Functions For generation, we employ the standard reparameterized denoising objective, introduced by Ho et al. in DDPM [11]. For detection, we maximize the cosine similarity between the text and region feature containing the target object, and minimize cosine similarity to all other region features proposed by OWL-v2 [30] in the image. For classification, we maximize cosine similarity between text and images of the target concept, and minimize cosine similarity to images that don't contain the target concept. Table 1 shows the exact loss definitions.

Metrics For generation, we report the rate at which an OpenAI CLIP L-14 [34] classifier predicts that generations are the target class (the set of class labels is the set of concepts names for that dataset from Appendix L), which we call Generation Accuracy in Table 1. For detection, we report the Mean Average Precision of bounding box predictions from OWL-v2 [30] on images from held-out validation sets, annotated with bounding boxes. For classification, we report DFN CLIP-based [8, 34] classifier accuracy given images of the target concept, and unrelated concepts, from validation sets. All metrics are reported as 95% confidence intervals over 100 randomized trials.

J Performance Of Constrained Soft Prompts

Performance Quickly Saturates We measure performance of solutions in the fractured embedding space for different constraint levels $\delta \in \{0.1, 0.2, 0.5, 1.0\}$, and find their performance is indistinguishable from unconstrained solutions. Figure 6 shows that performance saturates at a constraint level of $\delta = 0.5$, when the nearest neighbor is still the anchor word w_{anchor}. We observe that for all constraint levels $\delta > 1$, in-domain performance does not improve, despite the larger set of possible solutions. These results suggest that initialization is not very important, as performant solutions are likely close to any initialization. Instead, the data provided to the optimizer is likely more important.

K Perturbations Target The Final Layers

Results in Section 4.1 show that performant solutions are located everywhere in the embedding space, and most of these solutions are non-transferable. How can we tell these solutions apart from the cases in Section 3 that are transferable? One characteristic that identifies non-transferable solutions is their effect on the activations of the text encoder. Perturbative solutions like in Figure 6 generally

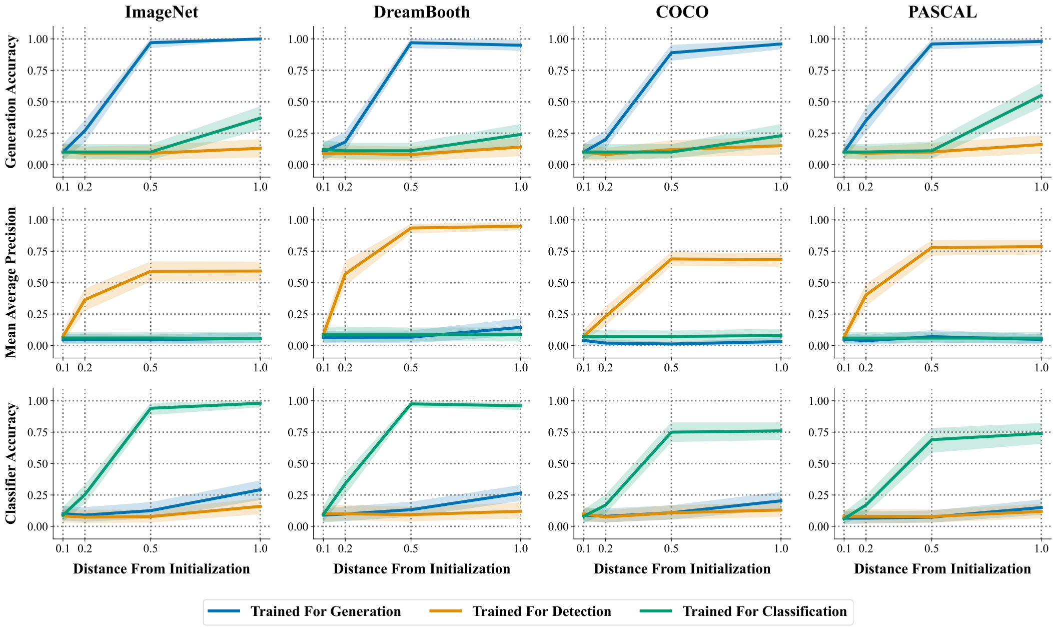

Figure 6: Performance (y-axis) of word vectors optimized to cause generation, detection, and classification of new visual concepts, for different constraint levels (x-axis). In-domain performance saturates at a constraint level of $\delta = 0.5$, which corresponds to solutions where the nearest existing word vector is the anchor w_{anchor}. Constrained solutions perform well in-domain, but typically don't perform well on transferred tasks for $\delta < 1$. Each line in the figure corresponds to the 95% confidence interval of 100 randomized trials for 10 concepts, and 10 anchor words per dataset. Refer to Appendix L for the concepts and anchor words used for each dataset.

target the final layers of the text encoder, and lead to a disagreement between early and later layers. Figure 7 shows generations from Stable Diffusion [37] when truncating the text encoder to just the first N transformer blocks (block = Norm $\rightarrow$ Attention $\rightarrow$ Residual $\rightarrow$ Norm $\rightarrow$ MLP $\rightarrow$ Residual). The bottom row shows TSNE visualizations of the pooling token activations at four evenly spaced layers in the text encoder of Stable Diffusion when generating concepts from the ImageNet [5] task. Activations initially cluster around the anchor concept, i.e. strawberry, and generations from early layers yield the anchor concept, strawberry, instead of the target concept we optimized for, sombrero. When transferred (visualizations in Appendix O), clusters and generations stay mismatched.

L Selected Concepts & Anchor Words

In this section, we discuss the concepts that were selected from ImageNet [5], COCO [24], PASCAL [7], and the DreamBooth dataset [38]. These concepts were selected uniformly at random without replacement from the available classes in each dataset. Ten classes were sampled per dataset in order to reduce the computational complexity of the experiments in the paper (results take 3 days to produce on just 40 visual concepts). These classes cover a diverse set of visual concepts.

On the ImageNet dataset [5], we select [’strawberry’, ’harp’, ’sturgeon’, ’gorilla’, ’throne’, ’pelican’, ’honeycomb’, ’barrel’, ’sombrero’, ’scuba diver’] as target concepts.

On the DreamBooth Dataset [38], we select [’cat2’, ’vase’, ’duck_toy’, ’candle’, ’colorful_sneaker’, ’backpack_dog’, ’grey_sloth_plushie’, ’fancy_boot’, ’clock’, ’pink_sunglasses’] as target concepts.

On the COCO dataset [24], we select [’laptop’, ’scissors’, ’donut’, ’bear’, ’cup’, ’dog’, ’bottle’, ’umbrella’, ’cat’, ’remote’] as target concepts.

On the PASCAL VOC dataset [7], we select [’airplane’, ’bicycle’, ’bird’, ’boat’, ’person’, ’train’, ’car’, ’cat’, ’horse’, ’cow’] as target concepts.

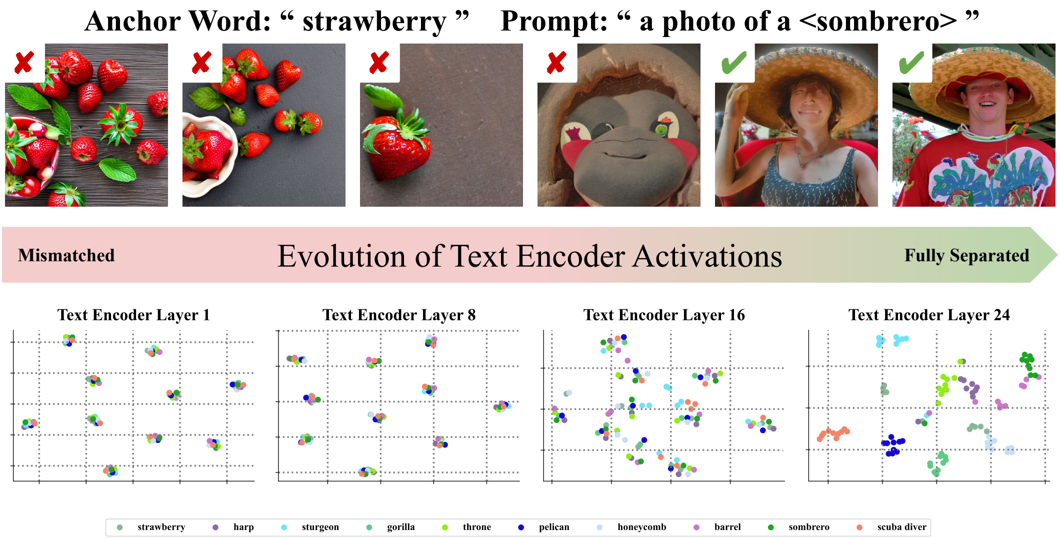

Figure 7: Prompts optimized for visual concepts target the final layers in text encoders. We show images generated by Stable Diffusion when truncating the text encoder to the first N layers, and create TSNE visualizations of the text encoder activations for the pooling token at four evenly spaced layers. Each color represents a different visual concept. Clusters in plots 1-16 represent anchor words from Section 4.1, which the activations cluster around instead of the target concept. When truncating the text encoder to just these layers, the anchor word (i.e. strawberry) is generated instead of the target concept (sombrero). Only by the final layers are clusters and generations correct. When transferred (visualizations in Appendix O) clusters and generations stay mismatched.

In addition to selecting concepts, we select anchor words that tokenize to a single token across all of the tested models. These are derived from the above target concepts.

On the ImageNet dataset [5], we select `[’strawberry’, ’harp’, ’sturgeon’, ’gorilla’, ’throne’, ’pelican’, ’honeycomb’, ’barrel’, ’hat’, ’scuba’]` as anchor words.

On the DreamBooth Dataset [38], we select `[’cat’, ’vase’, ’duck’, ’candle’, ’sneaker’, ’backpack’, ’plush’, ’boot’, ’clock’, ’sunglasses’]` as anchor words.

On the COCO dataset [24], we select `[’laptop’, ’scissors’, ’donut’, ’bear’, ’cup’, ’dog’, ’bottle’, ’umbrella’, ’cat’, ’remote’]` as anchor words.

On the PASCAL VOC dataset [7], we select `[’airplane’, ’bicycle’, ’bird’, ’boat’, ’person’, ’train’, ’car’, ’cat’, ’horse’, ’cow’]` as anchor words.

M Hyperparameters

In this section, we enumerate the hyperarameters used in the experiments in the paper. We choose hyperparameters agnostic to the model and task, so that results in the experiments are general, and not specific to the model. In Table 2 we note the HuggingFace model ID used, model configuration details, and hyperparameters from training, and evaluation.

N More Examples

In this section, we show more examples of generations from Stable Diffusion for perturbations to various unrelated anchor words in the embedding space. We show results for all combinations of 10 target concepts (row labels) and 10 anchor words (column labels) on ImageNet [5], COCO [24], PASCAL [7], and the DreamBooth dataset [38]. In several cases, nearly identical images are generated by Stable Diffusion for perturbations near to different unrelated anchor words.

Hyperparameter Name	**Hyperparameter Value**
Generation Model Name	Stable Diffusion 2.1 [37]
Generation Model HuggingFace ID	`stabilityai/stable-diffusion-2-1`
Generation Image Size	768 x 768
Detection Model Name	OWL-v2 [30]
Detection Model HuggingFace ID	`google/owlv2-base-patch16-ensemble`
Detection Image Size	960 x 960
Classification Model Name	Data Filtering Networks [8]
Classification Model HuggingFace ID	`apple/DFN2B-CLIP-ViT-L-14`
Classification Image Size	224 x 224
Examples Per Concept	8
Embedding Vectors Per Concept	4
Denoising Steps	50
Batch Size	8
Learning Rate	1e-04
Gradient Descent Steps	1000
Optimizer	Adam
Adam Beta1	0.9
Adam Beta2	0.999
Adam Epsilon	1e-08
Weight Precision	float16

Table 2: Hyperparameters used in the experiments of the paper. These parameters are held constant across all datasets and models. These choices are adapted from relevant prior work.

O More Visualizations

We provide more TSNE visualizations of the text encoder activations for different models and datasets in this section. Trends discussed in Section K hold across all models and datasets. Perturbative soft prompts like those found in Section 4.1 target the final layers in text encoders, and early activations in text encoders disagree with later activations. Generating images when truncating the text encoder to the first N layers leads to generations of the anchor work, instead of the target concept we are optimizing for (see Figure 7). When perturbative solutions are transferred, this transition stops.

Fine-tuning that targets the final layers of text encoders does not transfer, and Figure 18 shows that activations stop clustering by concept (color) when soft prompts are transferred.

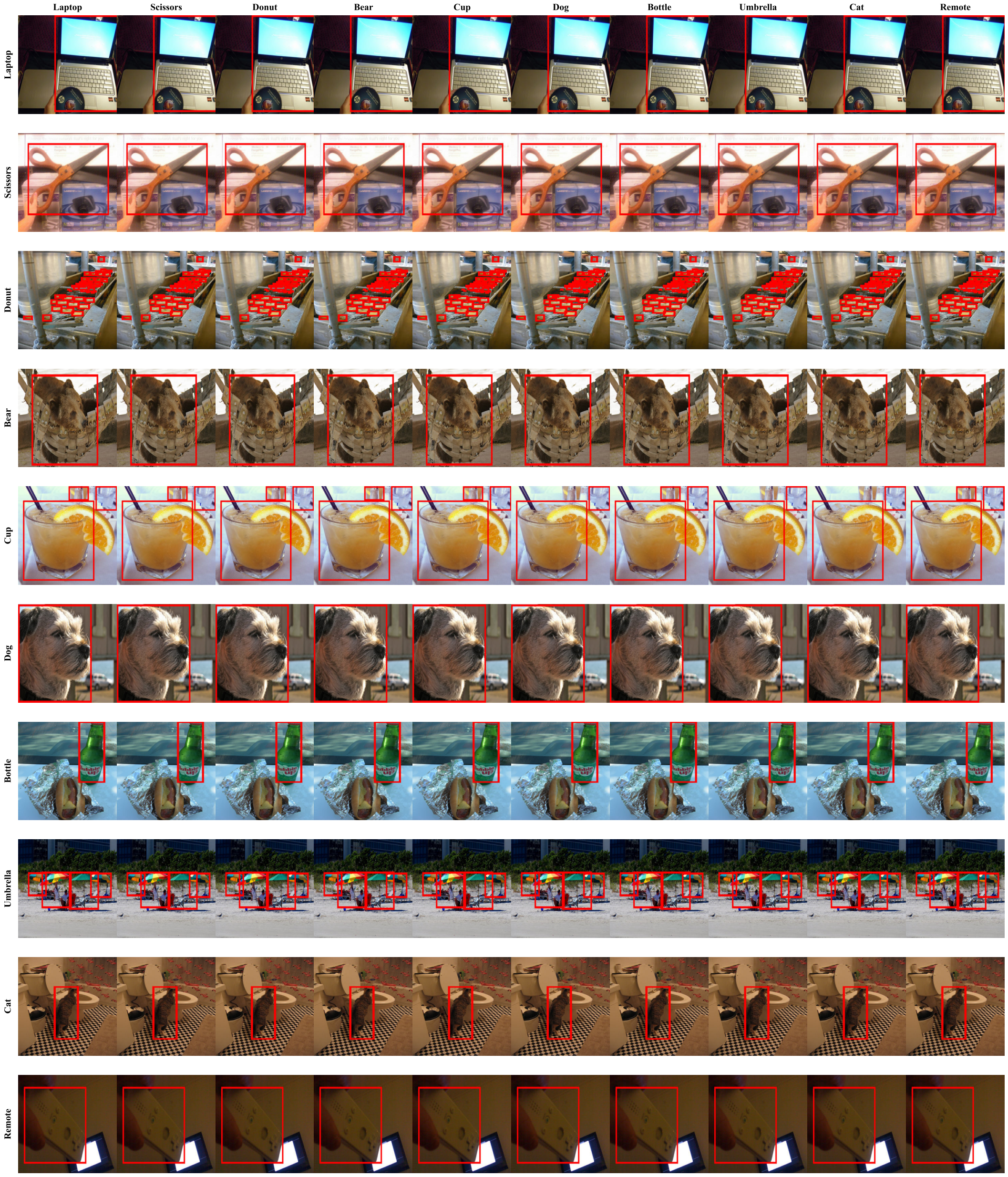

Figure 8: Visualizations of detections from OWL-v2 [30] using new embeddings optimized for detecting visual concepts on COCO [24]. Performant solutions for detecting arbitrary target concepts (row labels) are found with a constraint threshold $\delta = 0.5$ of unrelated anchor words (column labels).

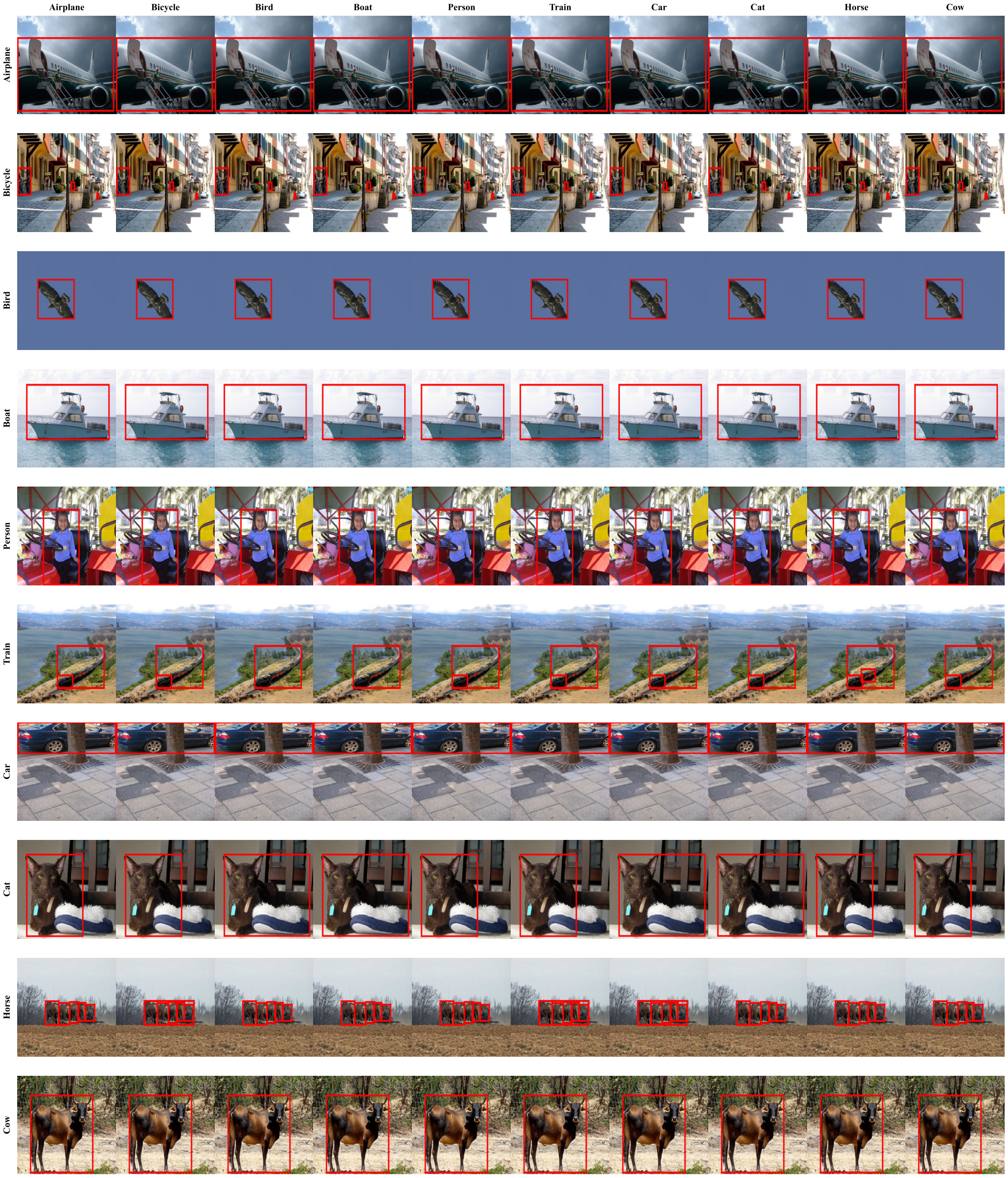

Figure 9: Visualizations of detections from OWL-v2 [30] using new embeddings optimized for detecting visual concepts on PASCAL [7]. Performant solutions for detecting arbitrary target concepts (row labels) are found with a constraint threshold $\delta = 0.5$ of unrelated anchor words (column labels).

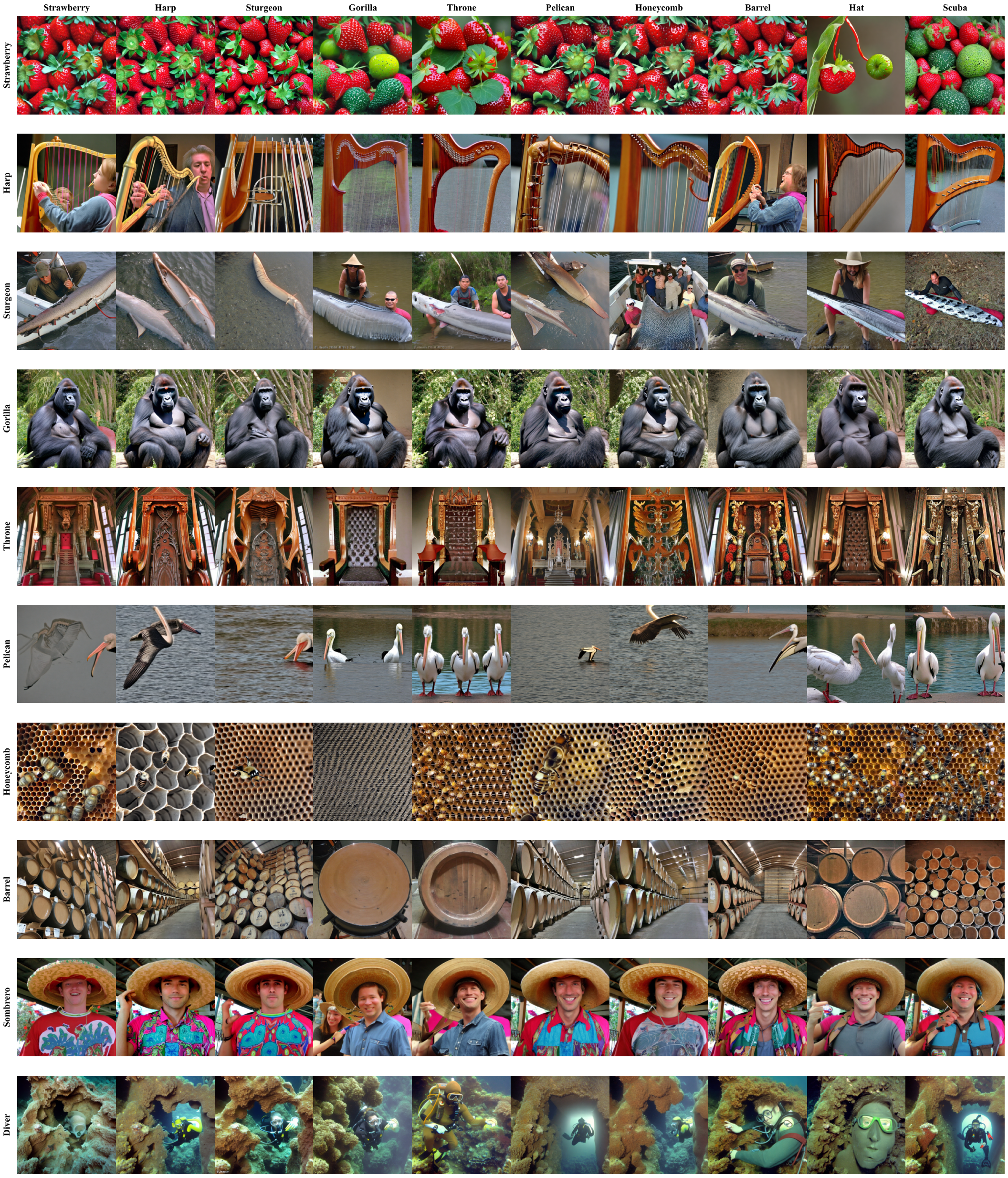

Figure 10: Visualizations of generations from Stable Diffusion 2.1 [37] using new embeddings optimized for generating visual concepts on ImageNet [5]. Performant solutions for generating arbitrary target concepts (row labels) are found with a constraint threshold $\delta = 0.5$ of unrelated anchor words (column labels). In several cases, different solutions far apart generate the same image.

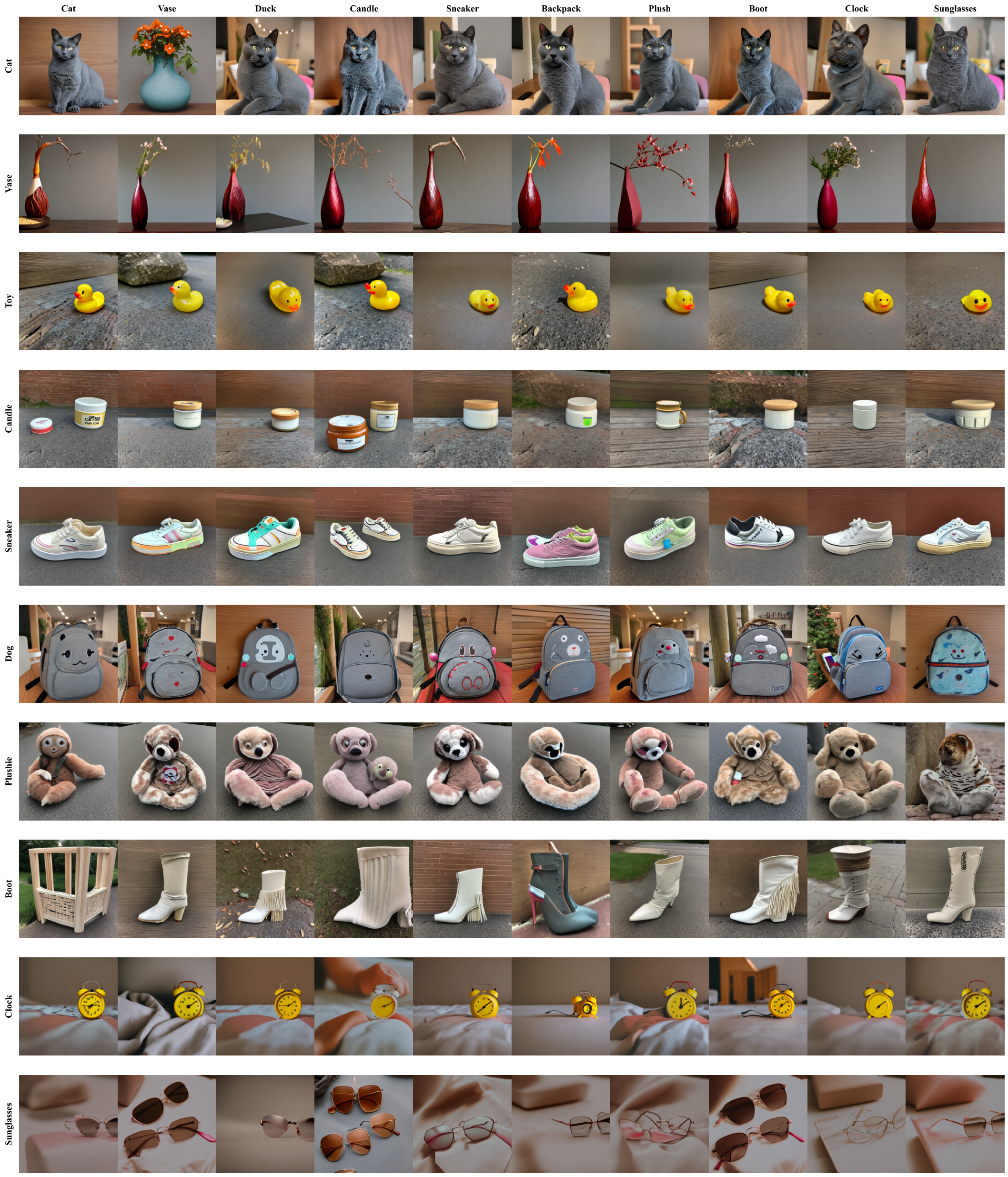

Figure 11: Visualizations of generations from Stable Diffusion 2.1 [37] using new embeddings optimized for generating visual concepts on DreamBooth [38]. Performant solutions for generating arbitrary target concepts (row labels) are found with a constraint threshold $\delta = 0.5$ of unrelated anchor words (column labels). In several cases, different solutions far apart generate the same image.

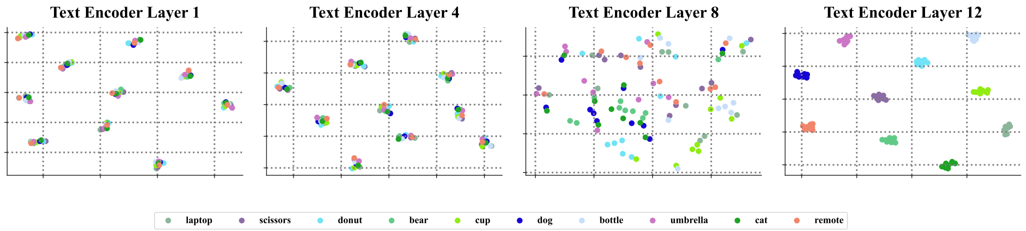

Figure 12: Visualizations of text encoder activations for OWL-v2 [30] on COCO [24] at four evenly spaced layers when optimizing soft prompts for detecting visual concepts (colored points), constrained to the neighborhood of various anchor tokens (clusters in plots 1-8).

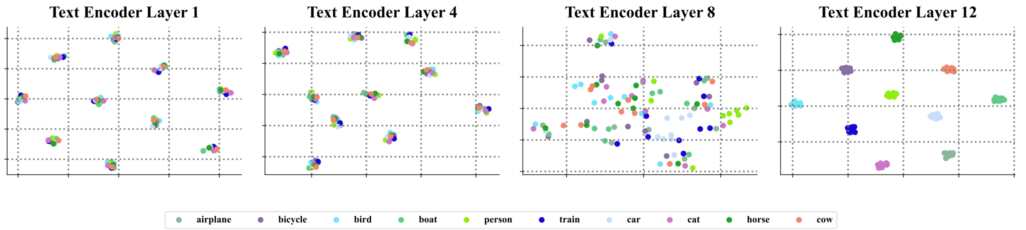

Figure 13: Visualizations of text encoder activations for OWL-v2 [30] on PASCAL [7] at four evenly spaced layers when optimizing soft prompts for detecting visual concepts (colored points), constrained to the neighborhood of various anchor tokens (clusters in plots 1-8).

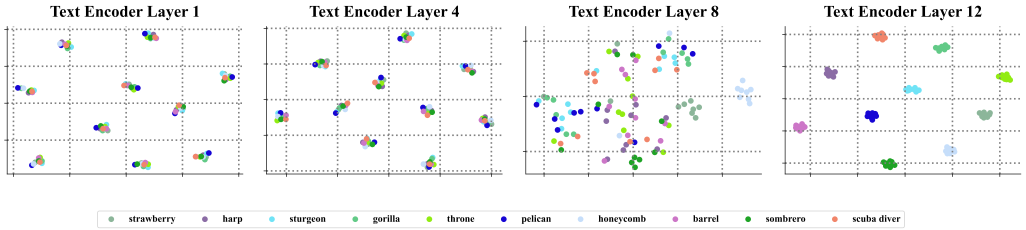

Figure 14: Visualizations of text encoder activations for DFN CLIP [8, 34] on ImageNet [5] at four evenly spaced layers when optimizing soft prompts for classifying visual concepts (colored points), constrained to the neighborhood of various anchor tokens (clusters in plots 1-8).

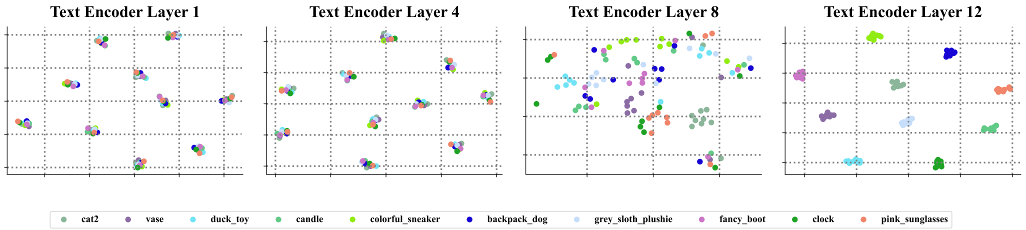

Figure 15: Visualizations of text encoder activations for DFN CLIP [8, 34] on DreamBooth [38] at four evenly spaced layers when optimizing soft prompts for classifying visual concepts (colored points), constrained to the neighborhood of various anchor tokens (clusters in plots 1-8).

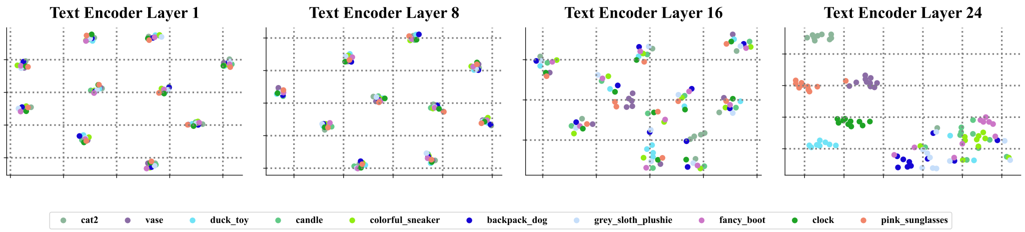

Figure 16: Visualizations of text encoder activations for Stable Diffusion 2.1 [37] on DreamBooth [38] at four evenly spaced layers when optimizing soft prompts for generating visual concepts (colored points), constrained to the neighborhood of various anchor tokens (clusters in plots 1-16).

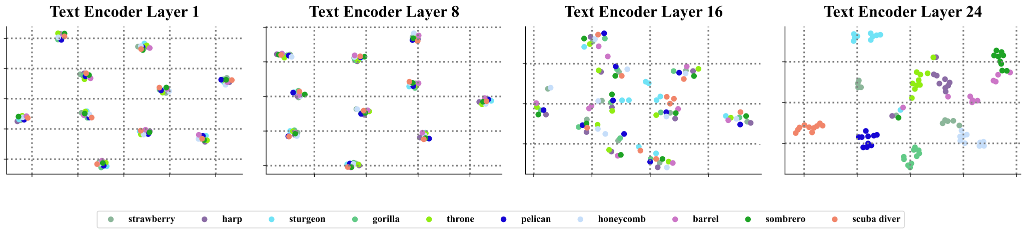

Figure 17: Visualizations of text encoder activations for Stable Diffusion 2.1 [37] on ImageNet [5] at four evenly spaced layers when optimizing soft prompts for generating visual concepts (colored points), constrained to the neighborhood of various anchor tokens (clusters in plots 1-16).

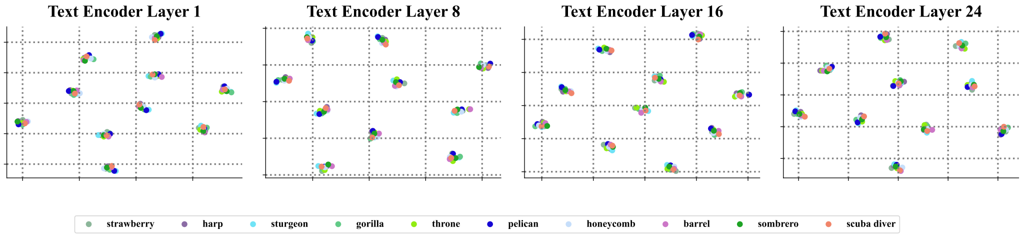

Figure 18: Visualizations of text encoder activations for Stable Diffusion 2.1 [37] on ImageNet [5] at four evenly spaced layers when optimizing soft prompts for classifying visual concepts (colored points) and transferring to generation, constrained to the neighborhood of various anchor tokens (clusters in plots 1-24). The evolution of clusters towards clean separation for in-domain evaluation stops when soft prompts are transferred. Fine-tuning that targets the original model is lost.