Zero-shot Improvement of Object Counting with CLIP

Ruisu Zhang*, **Yicong Chen***, **Kangwook Lee**
{rzhang345, ychen2229, kangwook.lee}@wisc.edu
Department of Electrical and Computer Engineering
University of Wisconsin-Madison

Abstract

We focus on the object counting limitations of vision-language models, with a particular emphasis on Contrastive Language-Image Pre-Training (CLIP) models. We assess the counting performance of CLIP using a custom dataset, which uncovers significant variations across diverse objects. To address this, we introduce a zero-shot, training-free method aimed at improving counting accuracy by manipulating the text embedding space of CLIP. Through comprehensive experiments, we demonstrate that our method not only enhances the counting capabilities of CLIP but also boosts the performance of text-to-image generative models like Stable Diffusion, particularly in generating images with precise object counts.

1 Introduction

Recent advancement of deep learning techniques has led to significant progress in vision-language models [1, 2, 3, 4, 5, 6]. One such breakthrough is the development of Contrastive Language-Image Pre-Training (CLIP) [3], which is trained on a wide range of Internet text-image pairs [7]. CLIP is shown to perform well on a wide range of zero-shot learning tasks, and it has been used as a text-image alignment backbone in many text-to-image generative models such as Stable Diffusion [8].

Despite its extensive deployment, CLIP exhibits limitations in certain areas [3, 9, 10, 11], such as counting objects in images [12]. Counting is a fundamental skill that requires the integration of visual and linguistic understanding, and it plays a crucial role in numerous practical applications.

Our work seeks for a deeper understanding of CLIP's counting ability and attempts to improve it via a simple yet effective zero-shot method. We start by creating a custom dataset containing images with varying quantities of objects. Upon evaluating CLIP's counting ability on this dataset, we find that its counting performance varies significantly across different objects. Our key idea is that if CLIP is effective at counting certain types of objects, it has already learned some counting knowledge, at least for the certain objects. This knowledge has the potential to be transferred to other objects that are harder to count, thereby improving CLIP's counting accuracy on them.

Our approach extracts counting knowledge, represented as a linear direction in the embedding space, from easily countable objects. This knowledge is then applied to the target object by augmenting its embedding with the counting-specific vector. Experiments show that this training-free method significantly boosts CLIP's inherent object-counting ability. We also explore the application of our method to text-to-image generation models, specifically the Stable Diffusion model [8]. The results indicate that our technique can guide Stable Diffusion to generate images with correct number of objects as specified in the prompt.

In sum, our contributions include: (i) we identify disparities in CLIP's ability to count different objects using our custom dataset; (ii) we introduce a zero-shot text embedding editing method, which

*These authors contributed equally to this work

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

substantially enhances CLIP's counting accuracy; and (iii) we show that our approach is also effective in guiding text-to-image generation models to produce images with more accurate object counts.

2 Related Work

Vision-language models Vision-language models (VLMs) have achieved significant success in multimodal tasks by training on massive image-text datasets and operating in a zero-shot or fine-tuning manner in downstream tasks [1, 2, 3, 4, 5, 6]. In this work, we will focus on the Contrastive Language-Image Pre-training (CLIP) model trained by OpenAI [3]. CLIP is trained on 400 million image-caption pairs [7], using a contrastive objective where matching text-image pairs should have a low cosine distance, while mismatched text and images should be far apart. CLIP has demonstrated notable success across a range of visual tasks due to its zero-shot capabilities. It also underpins text-to-image alignment in generative models like Stable Diffusion [8].

Limitations of vision-language models on counting While VLMs show impressive proficiency in many tasks, they have shortcomings in specific tasks [3, 9, 10, 11], like counting objects within pictures [12]. In fact, object counting problem has always been one of the important issues in the visual question answering (VQA) field, and several studies have attempted to address it [13, 14, 15, 16, 17]. Meanwhile, some research focuses on enabling VLMs-driven image generation models to produce images with the correct count of items [12, 18, 19].

While the aforementioned works are more centered on the application of VLMs, what's more relevant to our research are two papers that emphasize directly enhancing the counting capacity of VLMs themselves: CrowdCLIP [20] concentrates on the crowd counting problem, fine-tuning the CLIP in an unsupervised manner to map crowd patches to count text; Another work [12] proposes a counting-contrastive loss for fine-tuning pre-trained VLMs, based on a counting-relevant dataset filtered using object detection from the LAION-400M dataset [7]. It also introduces a new image-text counting benchmark *CountBench*, used to evaluate a model's understanding of object counting, which we also utilized in our experiments. However, both of these works rely on vast additional datasets and training resources to fine-tune CLIP. In contrast, our method requires no extra training and enhances CLIP's counting ability in a zero-shot manner by transferring knowledge between objects.

Text embedding editing Two works have explored the application of text embedding editing methods to image editing. One work [21] discovers editing directions in the text embedding space and applies them to image edits, while leveraging cross-attention guidance to preserve the structure of image content. Another work [22] translates example pairs that represent the "before" and "after" images of an edit back into a text-based editing direction, and then applies it to new images for image editing in a manner similar to [21]. In comparison, our research offers the following distinct contributions: (i) We utilize orthogonal projections to filter out extraneous details, thus achieving a more precise text embedding edit direction; (ii) Instead of concentrating solely on image editing, we focus on transferring CLIP's counting ability between different objects to enhance performance in counting-related image classification, image retrieval, and image generation.

3 Methods

In Section 3.1, we will describe how we created our dataset and specific ways to test CLIP's counting ability. After that, we will introduce our zero-shot method in Section 3.2.

3.1 Evaluation of CLIP's Counting Ability

We first collect our own dataset by manually searching for images of 9 different objects on the Internet. For each type of object, we gather 10 images each with two to five objects. We then modify each image using 10 different operations including rotations, vertical and horizontal flipping, and up and down adjustment of image brightness, contrast, color, and hue. This results in 11 images including the original one. In total, our dataset has $3960 (= 9 \times 10 \times 4 \times 11)$ samples.

We now introduce two counting tasks. The first task – zero-shot image classification – aims to find out the number of specific objects within a given image. For example, an image containing dogs will be classified as having i dogs if the image is more similar to the text "`i dogs`" than others with different counts. For this image classification task, we measure the classification accuracy. The second task type, known as text-based image retrieval, involves searching for and retrieving the most relevant images from a large dataset based on a given textual query. Given a type of object and an equal number of images for each object count, we calculate the probability of successfully retrieving the correct image for object count i as follows: First, we compute the similarity score between the caption "`i objects`" and all images. Then, we apply softmax on all similarity scores to estimate the probability of retrieving an image. Next, we sum up softmax scores of images of the same object count to estimate the probability of successfully retrieving any image of correct object count i. Lastly, we average estimated probability for all counting queries ranging from two to five.

For both tasks, the similarity score between an image and a text is calculated as the cosine similarity between an image embedding vector and a text embedding vector, computed with CLIP's image encoder and text encoder, respectively. The CLIP's counting accuracies are given in Section 4.2.

3.2 Our method: A zero-shot text embedding editing method

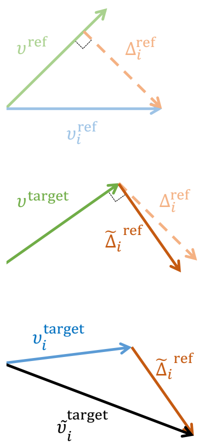


Figure 1: A visual illustration of our zero-shot text embedding editing method.

Our approach is based on the observation that CLIP is more proficient at counting certain types of objects. This strategy involves using an object, which CLIP can count effectively, as a reference to adjust the text embedding vectors that describe the target object.

To begin, we introduce some notations. Let υ^k denote the CLIP text embedding vector (e.g., "an image of dogs") that describes object k in a set of objects. It's important to note that it solely describes the object without any quantity information. We use υ_i^k to denote the text embedding that incorporates additional quantity information about object k, where i is the quantity of object k. For example, the embedding of the text "an image of *three* dogs" could be represented by υ_3^{dog}. We define the counting information direction extracted from any object k as

$$\Delta_i^k := (\upsilon_i^k - \upsilon^k) - \frac{\langle \upsilon_i^k - \upsilon^k, \upsilon^k \rangle}{\langle \upsilon^k, \upsilon^k \rangle} \upsilon^k, \tag{1}$$

such that Δ_i^k captures the information on $\upsilon_i^k - \upsilon^k$ and is also orthogonal to υ^k. The intuition behind this definition is that the counting information is encapsulated in the direction from the base (non-quantitative) representation (υ^k) to the quantitative representation (υ_i^k). Meanwhile, the orthogonality of Δ_i^k to υ^k serves to eliminate information associated with the base representation.

Assume there is an object that CLIP models can accurately predict its count, whose text embedding is denoted as υ^{ref}. In our approach, we use the counting direction extracted by Δ_i^{ref} as a reference direction to refine the counting signal in the representation υ^{target} of any given target object. Specifically, we derive a counting-augmented target object representation $\tilde{\upsilon}_i^{\text{target}} = \upsilon_i^{\text{target}} + \tilde{\Delta}_i^{\text{ref}}$, where

$$\tilde{\Delta}_i^{\text{ref}} := \Delta_i^{\text{ref}} - \frac{\langle \Delta_i^{\text{ref}}, \upsilon^{\text{target}} \rangle}{\langle \upsilon^{\text{target}}, \upsilon^{\text{target}} \rangle} \upsilon^{\text{target}}. \tag{2}$$

Similarly, the orthogonality of $\tilde{\Delta}_i^{\text{ref}}$ to υ^{target} serves to eliminate information associated with target object representation not contributing to object count. Our method is also illustrated in Figure 1.

4 Experiments and Results

4.1 Experimental Setup

Models. We evaluate our method on three versions of CLIP models [3], `clip-vit-base-patch32`, `clip-vit-base-patch16`, and `clip-vit-large-patch14`. These models have progressively smaller patch sizes, implying that each model represents a given image with increasing resolution. Furthermore, `clip-vit-large-patch14` has a larger model size compared to the first two models.

Datasets. We evaluate our method on our custom dataset, as detailed in Section 3.1, and the image counting benchmark, CountBench [12]. CountBench is an object counting dataset, collected from the

LAION-400M dataset [7]. It comprises 540 images, each displaying between two to ten instances of a specific object, with accompanying captions indicating these counts. Each numerical count is represented by 60 respective images. CountBench encompasses a diverse range of objects, and the captions, aside from indicating the counting number, contain extensive additional information.

Tasks. We evaluate our method on both image classification and text-based image retrieval tasks on our own dataset. With Countbench, we only evaluate our methods on the image classification task, since Countbench doesn't contain multiple images for a single type of object. We further divide the task into a four-class task and a nine-class task. The four-class task involves counting objects ranging from two to five, which aligns with the range used in our custom dataset. This allows for a direct comparison of the models' performance on the CountBench dataset and the custom dataset. On the other hand, the nine-class task involves counting objects ranging from two to ten, covering the full range of the CountBench dataset. This division serves to evaluate the models on tasks of varying complexity, providing a more comprehensive understanding of their counting abilities and the effectiveness of our method. The four-class task represents a simpler task, while the nine-class task provides a more challenging test of the models' counting abilities.

Caption template designs. In our experiments, we follow specific templates for text inputs. For each dataset, we have a set of target objects and reference objects. The target objects are the ones we aim to count, while the reference objects are those that CLIP can already count effectively. In our experiements, "cats" and "dogs" are selected as reference objects since all CLIP models can count them consistently more accurate than to count other objects. We use the counting direction extracted from the reference objects to adjust the text embeddings of captions that describe the target objects. Here are the templates we use:

Custom Dataset

Target Captions: "<objects>" vs. "<*i*> <objects>"

Example: "lions" vs. "**three** lions"

Reference Captions: "<objects>" vs. "<*i*> <objects>"

Example: "cats" vs. "**two** cats"

Countbench Dataset

Target Captions: "<context> <objects> <context>" vs. "<context> <*i*> <objects> <context>"

Example: "A set of cartoon calendars" vs. "A set of **four** cartoon calendars"

Reference Captions: "<objects>" vs. "<*i*> <objects>"

Example: "cats" vs. "**two** cats"

In the implementation of our method, we tailor the text embeddings for each target caption, each of which contains a different counting number. For a given counting number i, the adjustment is based on the reference vectors extracted from the reference object text embeddings corresponding to the same counting number i. For instance, for each object in our custom dataset, which contains four captions `"<i><objects>"` for $i \in [2, 3, 4, 5]$, we perform four parallel text embedding editing operations on each caption. Our method ensures that each counting number is more accurately represented in the adjusted embeddings.

4.2 Experimental Results

Table 1: **CLIP's counting accuracy on our custom dataset (%).** The counting accuracy of CLIP varies across diverse objects. We apply our method using "dogs" or "cats" as references. Accuracy is underlined if it is higher than the baseline accuracy and the highest score is highlighted in **bold**.

		average	dogs	cats	lions	chairs	goats	cows	cherries	roses	boats
CLIP-base-32	v_i^{target}	46.89	58.86	66.14	47.73	35.23	42.73	46.36	45.45	32.27	**47.27**
	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	**52.40**	**72.95**	70.23	**58.64**	**42.95**	**43.86**	**48.41**	**50.68**	36.59	**47.27**
	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	49.42	69.09	**70.45**	54.55	37.27	40.00	45.68	46.36	**36.82**	44.55
CLIP-base-16	v_i^{target}	50.33	**74.77**	74.77	54.32	47.05	32.73	55.00	35.00	34.09	45.23
	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	**56.02**	74.00	70.45	68.41	51.36	**52.50**	**58.41**	**39.09**	**42.27**	**47.73**
	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	55.08	69.3	**75.00**	**69.09**	**53.41**	51.36	56.14	37.05	38.86	45.45
CLIP-large-14	v_i^{target}	60.86	**75.23**	79.09	65.45	52.95	44.77	65.00	**53.86**	56.82	54.55
	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	64.29	74.55	**83.41**	66.59	52.50	72.27	68.41	51.59	57.05	52.27
	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	**64.44**	69.55	80.00	**67.27**	**53.41**	**74.77**	65.00	52.05	**64.77**	53.18

Table 2: **CLIP's image retrieval performance on our custom dataset.** The image retrieval performance of CLIP models varies across different objects. We apply our method using "dogs" or "cats" as references. Each row represents a different configuration, and each column represents a different object or the average performance across all objects. Probability is underlined if it is higher than the baseline and the highest score is highlighted in **bold**.

		average	dogs	cats	lions	chairs	goats	cows	cherries	roses	boats
`CLIP-base-32`	$\upsilon_i^{\text{target}}$	0.43	0.57	0.56	0.50	0.38	0.48	0.44	0.32	0.30	0.36
	$\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	**0.51**	**0.63**	0.62	**0.62**	0.47	**0.53**	0.54	**0.39**	**0.36**	0.41
	$\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	**0.51**	**0.63**	**0.63**	0.61	**0.48**	**0.53**	**0.55**	**0.39**	**0.36**	**0.42**
`CLIP-base-16`	$\upsilon_i^{\text{target}}$	0.45	0.56	0.60	0.53	0.39	0.44	0.50	0.38	0.30	0.32
	$\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	**0.53**	0.60	0.69	**0.63**	0.48	0.49	0.60	0.50	**0.37**	0.37
	$\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	**0.53**	**0.61**	**0.70**	0.62	**0.49**	**0.50**	**0.61**	**0.51**	0.36	**0.36**
`CLIP-large-14`	$\upsilon_i^{\text{target}}$	0.62	0.72	0.70	0.66	0.60	0.68	0.67	0.58	0.45	0.47
	$\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	**0.69**	**0.77**	**0.76**	**0.78**	0.61	0.73	**0.74**	**0.70**	**0.53**	0.62
	$\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	**0.69**	0.75	**0.76**	0.76	**0.64**	**0.74**	0.73	**0.70**	**0.53**	**0.64**

4.2.1 CLIP's counting ability on different objects

Table 1, in rows annotated by $\upsilon_i^{\text{target}}$, presents the unmodified performance accuracy of various CLIP models in matching a given image to the prompt with the correct number. Each column displays the accuracy of counting a specific object, with the object name used as the column header. The average accuracy across all objects is also calculated and displayed under the "average" column. We observe a positive correlation between model size and average counting accuracy, with accuracies ranging from 46.89% to 60.86%. However, the models' counting abilities significantly vary across different object types. This indicates that CLIP's counting ability is object-dependent. Notably, all models consistently perform best when counting "dogs" and "cats", while their performance with other objects lacks consistency.

Table 2, also in rows annotated by $\upsilon_i^{\text{target}}$, shows the probability of CLIP models, without modifications, retrieving an image with correct object count as specified in the caption. Similar to the image classification tasks, the performance varies across object types, with the highest probability consistently associated with correctly counting "dog" or "cat" images.

4.2.2 Effectiveness of our method

We tested our zero-shot method using "dog" and "cats" as reference objects to extract counting knowledge, based on their consistently high results in Table 1. The results of choosing "dog" and "cats" are shown in rows annotated by $\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{dog}$ and $\upsilon_i^{\text{target}} + \tilde{\Delta}_i^{cat}$. The results demonstrate that the counting accuracy of CLIP models can be improved by adjusting the target object's text embeddings with the counting direction extracted from either "dogs" or "cats". The improvement is observed across all models and for most of the objects. For instance, in the case of `CLIP-base-32`, the average counting accuracy improves from 46.89% to 52.40% when the counting direction extracted from "dogs" is used for adjustment. Similarly, for `CLIP-base-16`, the average counting accuracy improves from 50.33% to 56.02% with the same adjustment.

From Table 2 , we can observe that the use of our method improves the retrieval accuracy across all models and for most object types. For example, in the `CLIP-base-32 model`, the average retrieval accuracy improves from 0.43 to 0.51 when using either "dogs" or "cats" as the reference object. Both tables also show that the performance improvement is consistent across different model sizes, indicating the scalability of our method.

Table 3 demonstrates that our method's ability to improve counting accuracy extends to the CountBench dataset, which contains a diverse range of common real-world objects. For instance, in the case of `CLIP-base-32`, the accuracy for the 4-class task improves from 45.63% to 63.11% when the counting direction extracted from "dogs" is used for adjustment. However, as the task complexity increases from a four-class task to a nice-class task, our method becomes less effective.

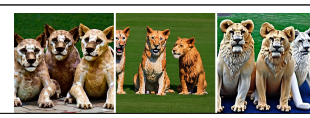
Table 3: **CLIP's counting accuracy on the CountBench dataset (%).** The table compares the accuracy of three CLIP models on two tasks (four-class and nine-class). We apply our method using "dogs" or "cats" as references. Accuracy is <u>underlined</u> if it is higher than the baseline accuracy and the highest score is highlighted in **bold**.

	`CLIP-base-32`			`CLIP-base-16`			`CLIP-large-14`		
	v_i^{target}	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	v_i^{target}	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$	v_i^{target}	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{dogs}}$	$v_i^{\text{target}} + \tilde{\Delta}_i^{\text{cats}}$
four-class	45.63	<u>63.11</u>	<u>**66.99**</u>	**56.94**	54.74	<u>**56.94**</u>	62.14	<u>62.75</u>	<u>**65.05**</u>
nine-class	21.03	<u>22.56</u>	<u>**27.18**</u>	29.46	<u>**29.86**</u>	<u>**30.66**</u>	22.56	<u>28.21</u>	<u>**30.00**</u>

4.3 Effectiveness of our method in improving text-to-image models' counting fidelity

We also apply our methodto see if it can improve the counting fidelity of Stable Diffusion model [8]. Shown in Table 4 are sample images generated with our method. More results are in Appendix A. One can note that after applying our method, Stable Diffusion's counting fidelity increases. Note that our method can be used in conjunction with any existing methods for improving the fidelity of text-to-image models, e.g., reinforcement learning-based algorithms [23, 24, 25].

Table 4: **Selected results from Stable Diffusion [8].** Images in the "Original" column are generated based on input prompt in the same row, using different sees. Images in the "Embedding edited" column are generated after applying our zero-shot method (using the same seeds), with the selection of "dog" as reference. We observe that our method is also effective on Stable Diffusion models.

Input Prompt	**Original**	**Embedding edited**
`"three lions"`		
`"An old building with ruined walls and four antique pink armchairs"`		
`"vintage silver plate tablespoons, serving spoon set of two"`		

5 Conclusion and Discussion

In this work, we have investigated the counting ability of CLIP models and proposed a novel zero-shot text embedding editing method. Our method extracts the counting knowledge embedded in a reference object's embedding and transfers this knowledge to others objects. Our experimental results demonstrate that CLIP's counting ability varies significantly across different object types, with the best performance observed when counting "dogs" and "cats". This observation led us to select these two objects as reference objects to extract counting knowledge. We found that our approach can significantly improve CLIP's counting accuracy, in both image classification tasks and image retrieval tasks. This improvement is consistently observed across all models and for most of the objects.

However, our work has some limitations. Firstly, the performance improvement varies across different objects, and for some objects, the improvement is not significant. This suggests that the counting knowledge extracted from "dogs" and "cats" may not be fully applicable to all objects. Secondly, our method requires prior knowledge or evaluation to first identify a good reference object, which may not always be feasible. Thirdly, our method does not work effectively if the image contains more than five objects, limiting its applicability to images with larger object counts.

Looking ahead, there are several promising directions for future research. First, we could explore other methods for extracting and transferring counting knowledge. For example, we could consider using multiple reference objects to extract a more general counting direction. Second, we could investigate whether the counting direction can be learned in a supervised manner using a large labeled dataset. Third, we could extend our method to other tasks beyond object counting and text-to-image generation to further explore its potential. Finally, we could explore the theoretical aspects of our method, such as why it works and under what conditions it is expected to work. This could lead to a deeper understanding of the counting ability of CLIP models and potentially inspire new methods for enhancing their performance.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, and Weicheng Kuo. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong 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.

[4] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15617–15629. IEEE, 2022.

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *CoRR*, abs/2304.08485, 2023.

[6] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *CoRR*, abs/2306.00890, 2023.

[7] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[9] Nan Liu, Shuang Li, Yilun Du, Josh Tenenbaum, and Antonio Torralba. Learning to compose visual relations. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 23166–23178, 2021.

[10] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 5228–5238. IEEE, 2022.

[11] Roni Paiss, Hila Chefer, and Lior Wolf. No token left behind: Explainability-aided image classification and generation. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XII*, volume 13672 of *Lecture Notes in Computer Science*, pages 334–350. Springer, 2022.

[12] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. *arXiv preprint arXiv:2302.12066*, 2023.

[13] Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. *CoRR*, abs/2305.07304, 2023.

[14] Jingyi Xu, Hieu Le, Vu Nguyen, Viresh Ranjan, and Dimitris Samaras. Zero-shot object counting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 15548–15557. IEEE, 2023.

[15] Yan Zhang, Jonathon S. Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[16] Duy-Kien Nguyen, Vedanuj Goswami, and Xinlei Chen. Movie: Revisiting modulated convolutions for visual counting and beyond. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[17] Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 8076–8084. AAAI Press, 2019.

[18] Yumeng Li, Margret Keuper, Dan Zhang, and Anna Khoreva. Divide & bind your attention for improved generative semantic nursing. *CoRR*, abs/2307.10864, 2023.

[19] Wonjun Kang, Kevin Galim, and Hyung Il Koo. Counting guidance for high fidelity text-to-image synthesis. *CoRR*, abs/2306.17567, 2023.

[20] Dingkang Liang, Jiahao Xie, Zhikang Zou, Xiaoqing Ye, Wei Xu, and Xiang Bai. Crowdclip: Unsupervised crowd counting via vision-language model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 2893–2903. IEEE, 2023.

[21] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In Erik Brunvand, Alla Sheffer, and Michael Wimmer, editors, *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, pages 11:1–11:11. ACM, 2023.

[22] Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: Image editing via visual prompting. *CoRR*, abs/2307.14331, 2023.

[23] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: reinforcement learning for fine-tuning text-to-image diffusion models. *CoRR*, abs/2305.16381, 2023.

[24] Ying Fan and Kangwook Lee. Optimizing DDPM sampling with shortcut fine-tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan 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 9623–9639. PMLR, 2023.

[25] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *CoRR*, abs/2305.13301, 2023.

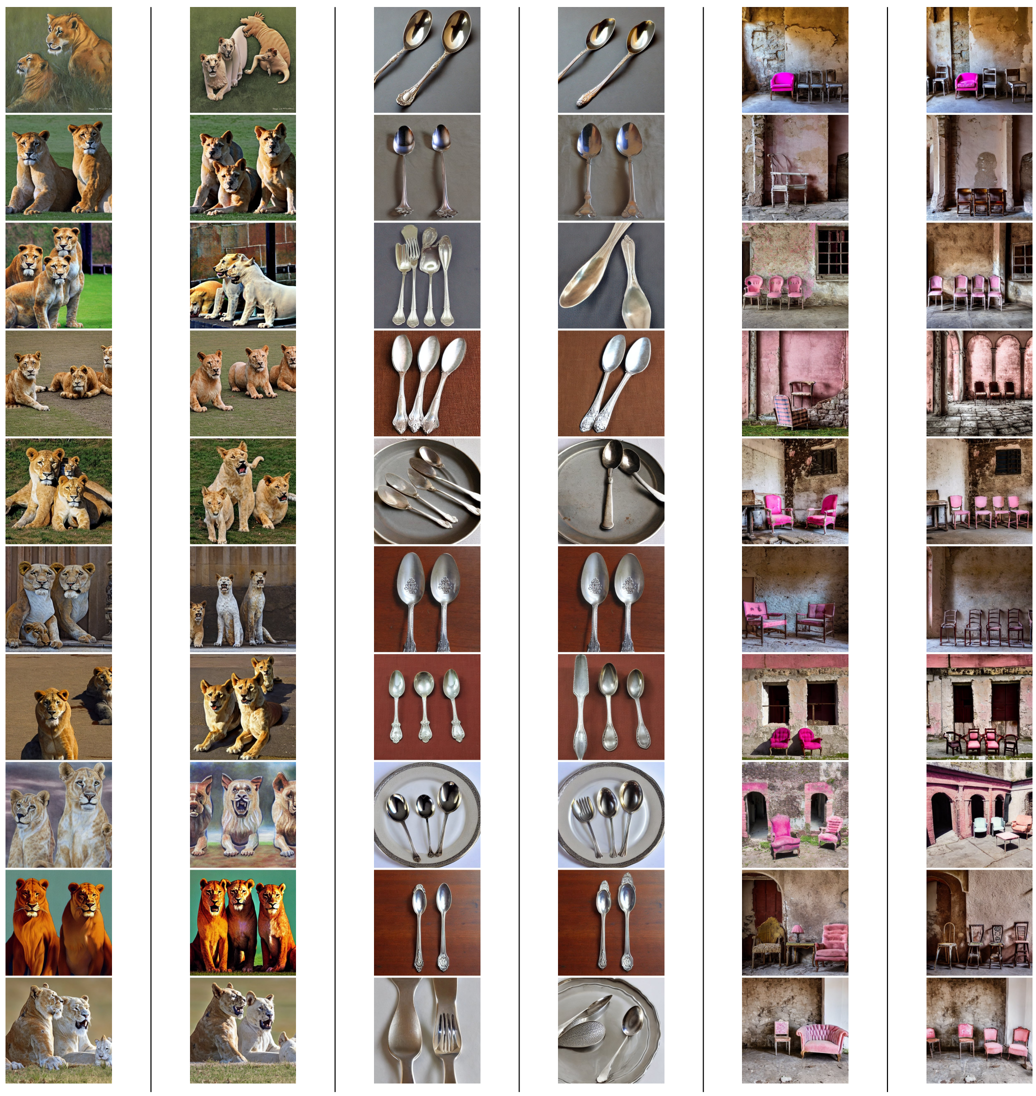
Appendix

A Effectiveness of our method in improving text-to-image models' counting fidelity

We provide more examples to show the effectiveness of applying our method to Stable Diffusion [8] to see if it can improve the counting fidelity of the text-to-image generation model. We show results from 3 prompts, where for each prompt 30 images are generated with 30 unique random seed. To compare our method with unmodified Stable Diffusion baseline, images in the same row are generated using the same random seed. It's worth noting that our method is not always effective. However, it increases the frequency of Stable Diffusion generating images with correct object count.

‘‘three lions’’		‘‘vintage silver plate tablespoons, serving spoon set of two’’		‘‘An old building with ruined walls and four antique pink armchairs’’	
Original	Embedding edited	Original	Embedding edited	Original	Embedding edited
					

“three lions”		“vintage silver plate tablespoons, serving spoon set of two”		“An old building with ruined walls and four antique pink armchairs”	
Original	Embedding edited	Original	Embedding edited	Original	Embedding edited



“three lions”		“vintage silver plate tablespoons, serving spoon set of two”		“An old building with ruined walls and four antique pink armchairs”	
Original	Embedding edited	Original	Embedding edited	Original	Embedding edited