In this supplementary material, we present additional experimental results. The supplementary comprises the following subsections:

1. Sec. 1, Comparison to Baselines: Qualitative Results
2. Sec. 2, Additional Qualitative Results
3. Sec. 3, collapse of inference-time optimization
4. Sec. 4, detailed quantitative analysis
5. Sec. 5.1, CLIP loss coefficient ablation study
6. Sec. 5.2, learning rate ablation study
7. Sec. 5.3, YOLO confidence threshold ablation study

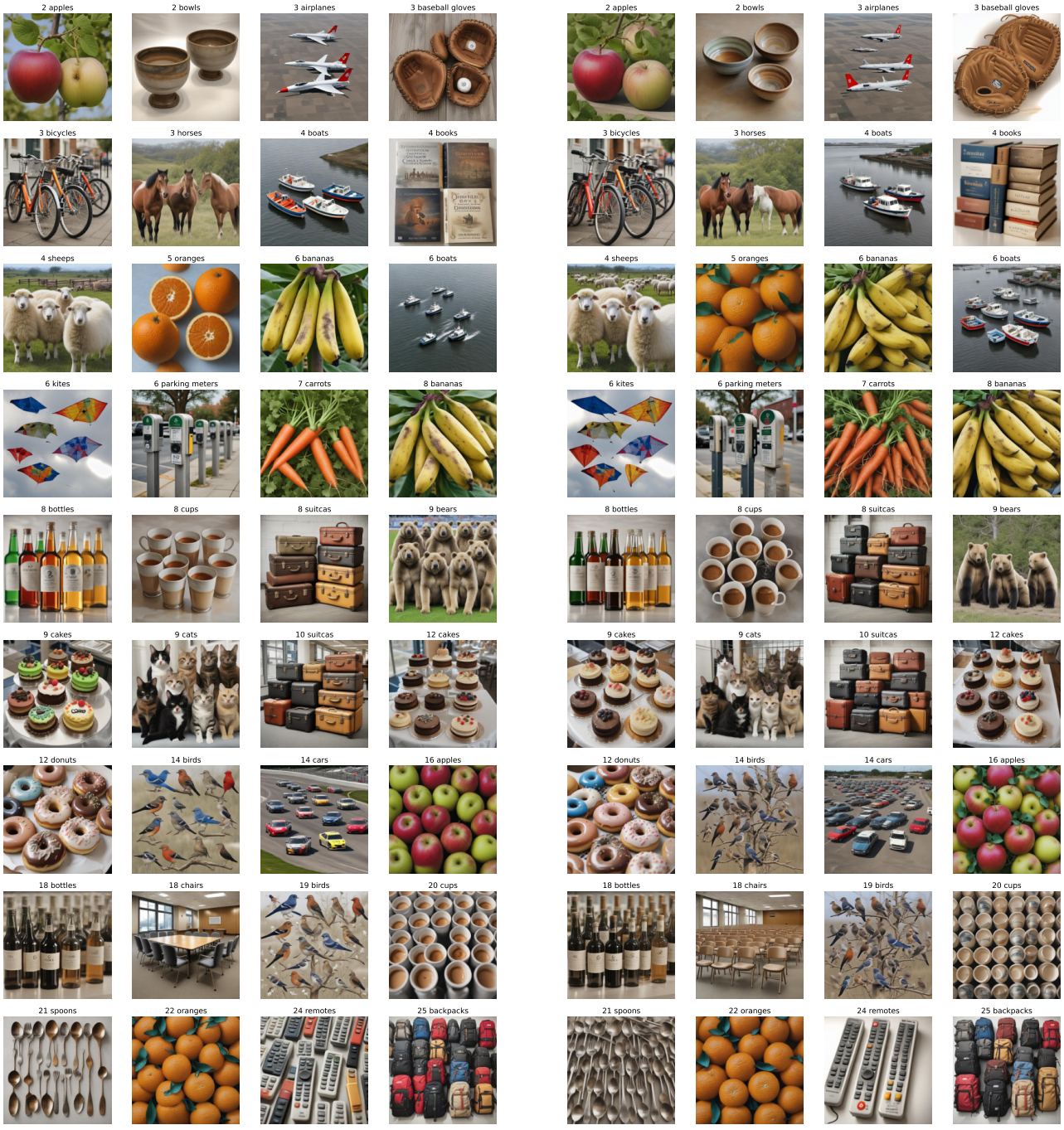
1. Comparison to Baselines: Qualitative Results

Figure 1 presents qualitative comparisons between baseline methods and various variants of our approach. Our method effectively corrects object counts while preserving the overall image content. In contrast, Counting-Guidance often fails to adjust the count, resulting in images that remain largely unchanged. CountGen, on the other hand, frequently alters the overall composition by aggressively intervening in the denoising process, leading to the addition or removal of objects.

Figure 1. Comparison to baselines of our method with different scaling variants, as well as Counting-Guidance and CountGen. For each example, we show the original image generated by the backbone model (top) and the optimized version with corrected object count (bottom). Prompts used for generation are shown on the left.

2. Additional Qualitative Results

In Fig. 2, we show additional results of our method using DETR-based scaling, compared to the backbone. Samples shown in increasing order. In many cases, SD-Turbo depicts partial objects, such as multiple heads or broken chairs. Our method often corrects these issues, enhancing naturalism and counting accuracy. We find that for large quantities, the number of objects often does not differ meaningfully; for instance, the outputs for 14 and 19 birds are almost identical.



(a) Ours (b) SD

Figure 2. Additional results of our method using DETR-based scaling, compared to the backbone. Samples shown in increasing order.

3. Stability and Failure Modes of Inference-Time Optimization

Inference-time optimization can suffer from training collapse, where the image either fails to change or becomes distorted. We observed these issues across the baselines. These issues may arise because their method does not incorporate any penalty, such as our CLIP penalty, to ensure semantic accuracy.

Counting-Guidance [1] often struggles with large object counts. As shown in Fig. 3, the method frequently applies minimal correction (left and middle) or collapses into noisy, indistinct patterns (right).

CountGen [2] exhibits a different failure mode (Fig. 4). It often enforces the correct count, but at the cost of visual realism. The middle image shows unrealistic object shapes, while the rightmost image alters the scene semantics, despite achieving a successful count.

We encountered similar collapses in our early experiments using **InstaFlow** [3]. As Fig. 5 shows, removing the CLIP alignment penalty resulted in adversarial changes, e.g., color distortions or content shifts, despite correct counts. Adding a CLIP penalty preserved prompt alignment. Notably, such failures did not occur with **SD-Turbo,** which remained stable even without CLIP guidance.

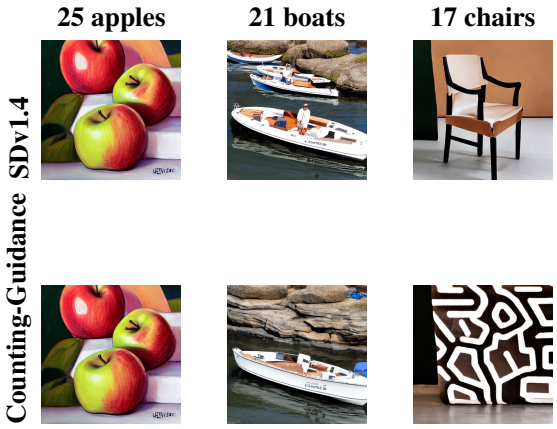


Figure 3. Failure cases of Counting-Guidance. The left and middle show little to no correction for large counts; the rightmost image collapses into a noisy pattern.

Figure 4. Failure cases of CountGen. Left: incorrect count. Middle: unrealistic appearance. Right: accurate count and realism, but altered scene.

Figure 5. Collapse in early experiments with InstaFlow [3] without CLIP regularization. Top: original generation. Bottom: optimized but adversarial.

4. Detailed quantitative analysis

In Fig. 6, we show the YOLO-based counting error for our method compared to the baselines. Although the comparison is not entirely fair, as each baseline uses a different backbone, our method achieves the highest accuracy for large object counts, despite relying on SD Turbo, the weakest backbone. For smaller quantities, our performance is comparable to other methods, even though they benefit from stronger backbones.

In Fig. 7, we show detailed scores of our benchmark for each number of objects from two to 25. Our method consistently outperforms others in accuracy. The error grows linearly as the number of objects increases from two to 10. Interestingly, the error remains relatively constant beyond ten objects. This could be due to the challenges the YOLO-based counting method faces with larger numbers, affecting the metric's accuracy.

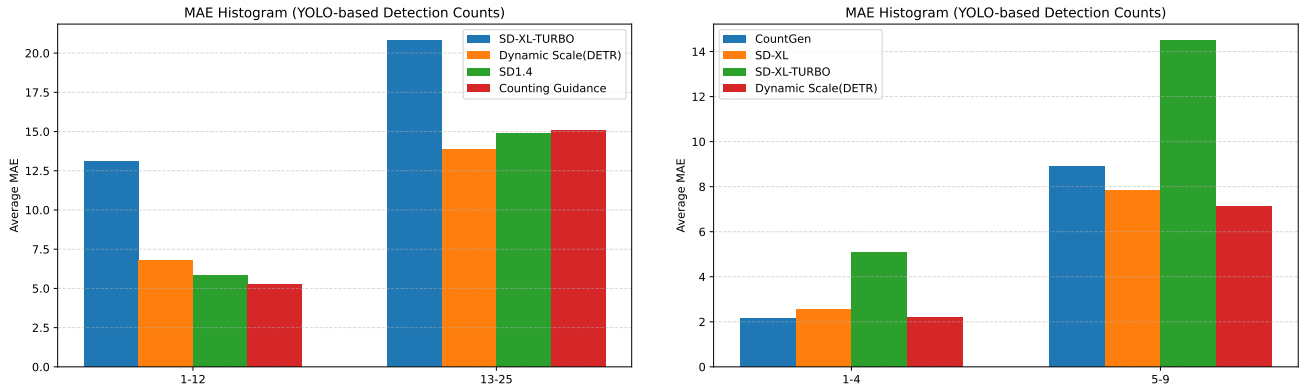


Figure 6. Comparison of different methods on sets of varying quantities. Since CountGen supports only up to nine objects, we limit the evaluation to 1-4 for small quantities and 5-9 for large quantities. We use YOLO to count the number of objects and report the mean absolute error (MAE).

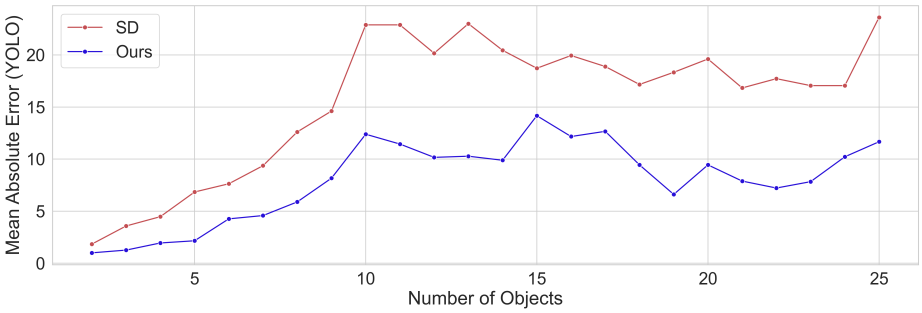


Figure 7. MAE scores as a function of the number of objects ranging from 2 to 25.

5. Ablation study

5.1. CLIP penalty

Next, in Fig. 8, we assess the influence of the CLIP penalty term. We find that setting it to five improves the MAE score. Beyond five, while the CLIP score improves, there is a trade-off with the MAE.

5.2. Learning rate

In Fig. 9, we experimented with our method using learning rates of 0.00025, 0.00125, 0.01, 0.05, and 0.1. The learning rate of 0.01 yielded the best results, suggesting that our method benefits from a higher learning rate.

5.3. detection confidence threshold

Our dynamic scale factor method employs the detection mechanism to accurately scale the object's density map. Detectors defines a threshold for object detection.

In Fig. 10 and Fig. 11, we ablate this parameter and find that a threshold of 0.3 yields the best results for our detectors.

The detection models struggle to be confident in images with many objects. Since our benchmark also consists of a relatively high number of objects, a lower threshold helps achieve more accurate results.

We hypothesize that dynamically changing this threshold for different counting requirements (e.g., higher confidence for smaller quantities below five and lower confidence for higher quantities beyond twenty) could lead to further improvements.

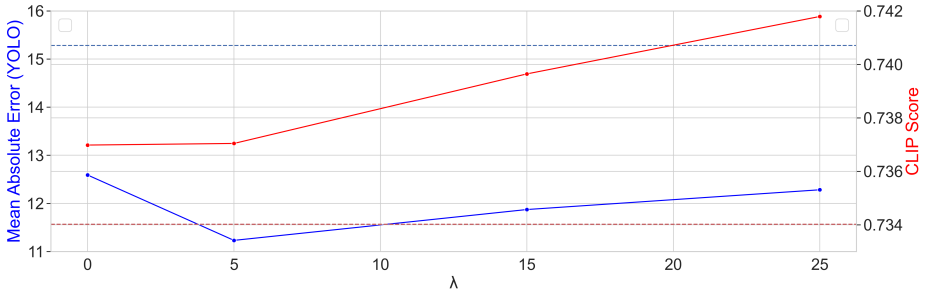


Figure 8. MAE (blue) and CLIP (red) scores as a function of λ. We show the SD-Turbo result baseline in a dotted line.

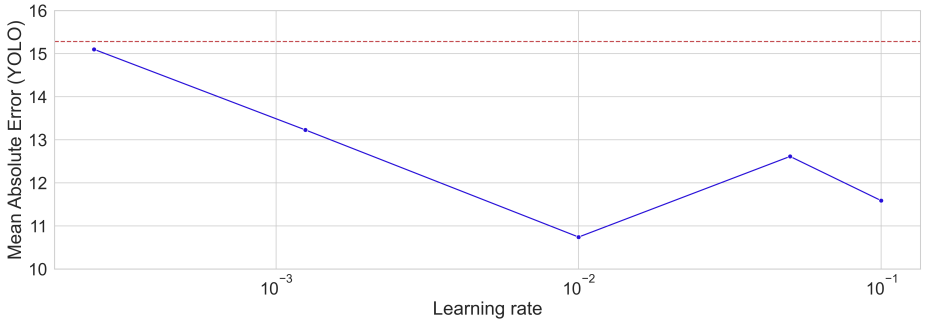


Figure 9. MAE score as a function of the learning rate. The MAE of SD-Turbo is displayed in red.

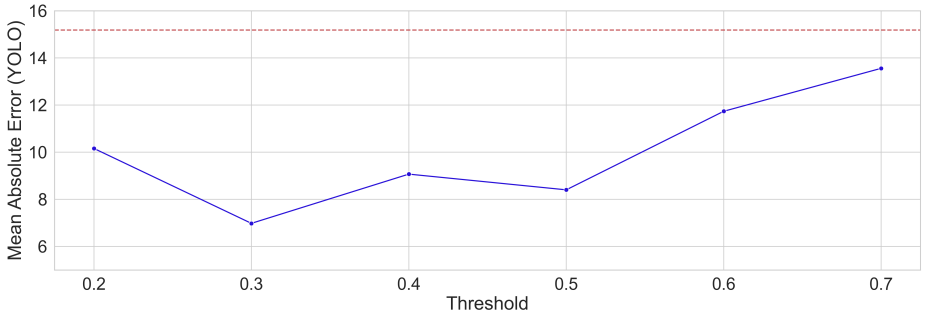


Figure 10. MAE score as a function of YOLO's confidence threshold. The MAE of SD-Turbo is displayed in red as a baseline.

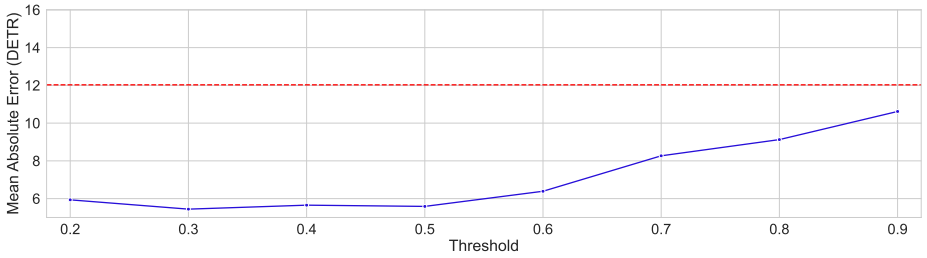


Figure 11. MAE score as a function of DETR's confidence threshold. The MAE of SD-Turbo is displayed in red as a baseline.

References

[1] Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make it count: Text-to-image generation with an accurate number of objects. *arXiv preprint arXiv:2406.10210*, 2024. 6

[2] Wonjun Kang, Kevin Galim, Hyung Il Koo, and Nam Ik Cho. Counting guidance for high fidelity text-to-image synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 899–908. IEEE, 2025. 6

[3] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024. 6