

LLM-guided Instance-level Image Manipulation with Diffusion U-Net Cross-Attention Maps

Andrey Palaev¹
palaev@aidecisions.ai

Adil Khan²
a.m.khan@hull.ac.uk

Syed M. Ahsan Kazmi³
ahsan.kazmi@uwe.ac.uk

¹ AIDecisions, SW20 ODS, London, UK

² School of Computer Science, University of Hull, HU6 7RX, Hull, UK

³ Department of Computer Science, University of the West of England, BS16 1QY, Bristol, UK

1 Additional examples

Fig. 1 shows additional examples of manipulating position with our method, Self-Guidance [1] and Dragon Diffusion [2].

2 Quantitative evaluation

In addition to visual evaluation, we have calculated the following metrics:

1. Average number of objects removed by the pipeline. This is calculated using IoU-based matching of bounding boxes between the original and the resulting images. In other words, it means the number of objects that are present in the original image but are not in the resulting one. The best value for this metric is 0 since the number of objects is not supposed to change.
2. Average number of objects added by the pipeline. This is calculated the same way as the previous metric. In other words, it means the number of objects that are not present in the original image but are in the resulting one. The best value for this metric is 0 since the number of objects is not supposed to change.
3. Percentage of successful manipulations. The manipulation is considered to be successful if the target object has a match after performing IoU-based matching. The best value for this metric is 1 since we want the target object to be moved.
4. Average CLIP [3] and DINOv2 [4] similarity scores for matched detected objects. For each pair of matched objects described above, we calculate CLIP and DINOv2 similarity scores to measure how the detected objects are preserved by our pipeline.

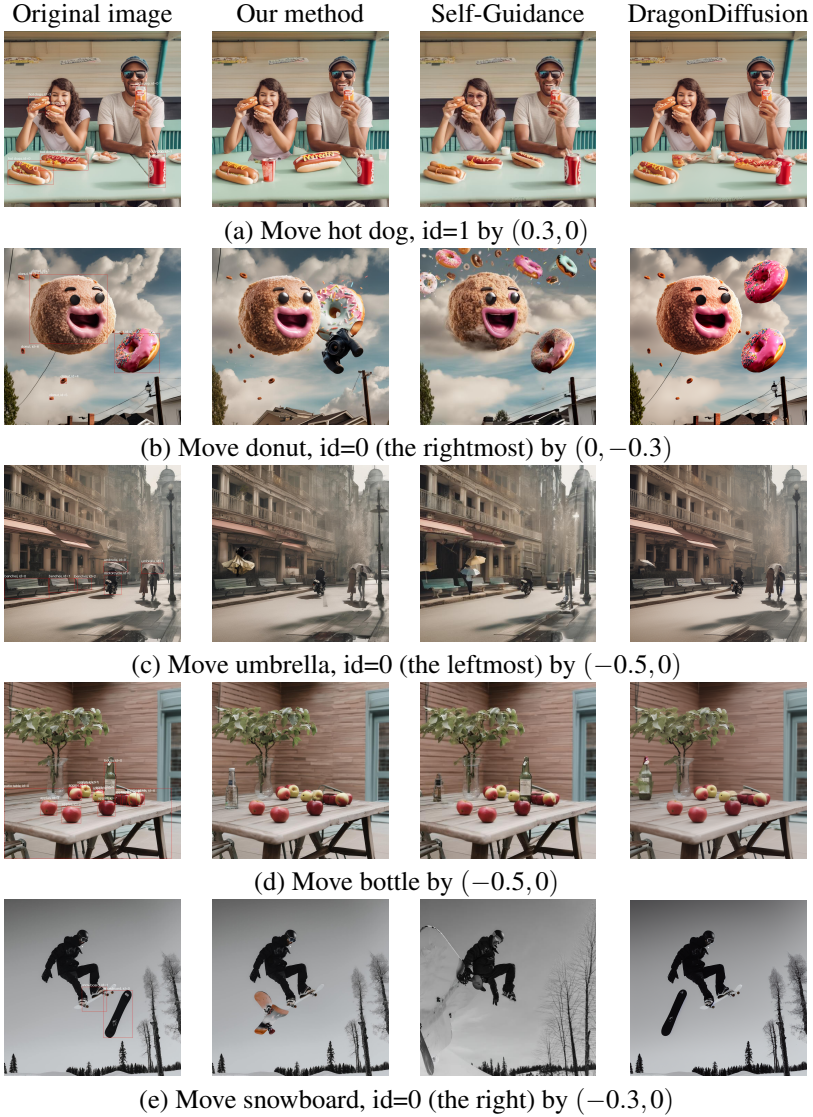


Figure 1: Additional examples of the position manipulations. Coordinates shift is represented by (x, y) .

5. CLIP [15] and DINOv2 [16] similarity scores between original and resulting images. This metric shows how well the overall image is preserved, including the background.

For calculating these metrics, we have selected 20 random captions from the COCO captions dataset [17] and applied our pipeline to each of the captions selecting a random manipulation, i.e. the manipulated object and the target positions are chosen randomly for the uniform distribution. For this experiment, we have set $w_0 = 30.0$, $w_1 = 8.0$. The pipeline achieved good object and image preservation, indicated by the high CLIP and DINOv2 similarity scores. However, the number of removed and added objects is high as well as the portion of

successful manipulations is very low. This might be caused by three possible reasons:

1. For each manipulation, the pipeline requires a different set of manipulation coefficients. Hence, setting a single set of weights for each manipulation may lead to successful manipulations in some cases, but unsuccessful in others.
2. Since the position term in our manipulation pipeline focuses on maximizing the pixel values in the target location in the cross-attention maps, sometimes the target object appears in that location to be much larger than it was originally, making it not matched to the original object and the manipulation considered to fail.
3. The object detector sometimes fails to detect some objects, leading to mismatches and manipulations being considered failures.

Examples of failed manipulations can be seen in Fig. 2.

Table 1: Metrics for our pipeline calculated on COCO captions dataset

Average number of removed objects	0.185
Average number of added objects	0.214
Portion of successful manipulations	0.3
Average CLIP similarity score for matched objects	0.969
Average DINOv2 similarity score for matched objects	0.891
Average CLIP similarity score for images	0.991
Average DINOv2 similarity score for images	0.992

References

- [1] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.
- [2] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion Self-Guidance for controllable image generation. In *Advances in Neural Information Processing Systems*, volume 36, pages 16222–16239, 2023.
- [3] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. DragonDiffusion: Enabling drag-style manipulation on diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.
- [4] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [5] Alec Radford et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763, 18–24 Jul 2021.

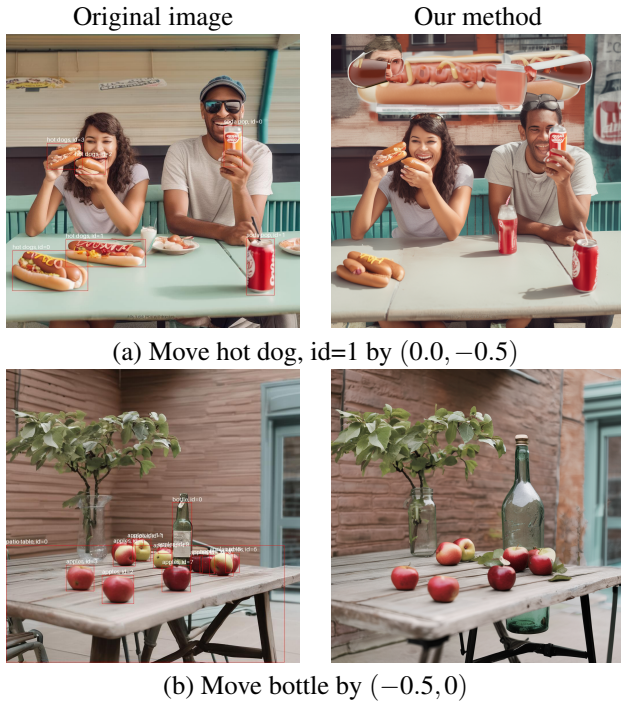


Figure 2: Failure examples for our method.