SpotEdit: Evaluating Visually-Guided Image Editing Methods

Sara Ghazanfari[1,2*], **Wei-An Lin**[2], **Haitong Tian**[2], **Ersin Yumer**[2]

[1]New York University, US [2]Adobe Inc.

Abstract

Visually-guided image editing, where edits are conditioned on both visual cues and textual prompts, has emerged as a powerful paradigm for fine-grained, controllable content generation. Although recent generative models have shown remarkable capabilities, existing evaluations remain simple and insufficiently representative of real-world editing challenges. We present SpotEdit, a comprehensive benchmark designed to systematically assess visually-guided image editing methods across diverse diffusion, autoregressive, and hybrid generative models, uncovering substantial performance disparities. To address a critical yet underexplored challenge, our benchmark includes a dedicated component on *hallucination*, highlighting how leading models, such as GPT-4o, often hallucinate the existence of a visual cue and erroneously perform the editing task. Our code and benchmark are publicly released at github.com/SaraGhazanfari/SpotEdit.

1 Introduction

Visually-guided image editing enables precise, localized manipulation by combining a reference image with textual instructions to guide generative models. Compared to text-only editing, this multimodal approach provides greater control, stronger semantic alignment, and higher spatial precision, making it valuable for applications such as consistent keyframe editing in the area of long-form video editing [26, 5].

Despite rapid advances in diffusion-based [19, 21] and autoregressive [3, 15] generative methods, rigorous evaluation of visually guided editing remains underexplored. Existing benchmarks [22, 7] focus largely on coarse manipulations, simple object replacements, or single-object scenes (see Fig. 4), offering limited insight into the complexities of real-world editing. As a result, current evaluations fail to capture the nuanced challenges of multimodal guidance, hindering fair model comparison and progress.

To close this gap, we introduce SpotEdit, a comprehensive benchmark for systematic and fine-grained evaluation of visually guided image editing. SpotEdit is built from diverse real and synthetic video frames, enabling controlled variation in object appearance, position, scale, and context. More specifically, each benchmark instance consists of: a reference image, an input image, and a textual instruction, and a ground-truth target image. Crucially, SpotEdit also includes a dedicated *Hallucination* subset that probes failure cases where objects are missing from either the reference or the input image. This component directly evaluates a model's robustness to edge cases, testing its ability to avoid spurious insertions while preserving both spatial coherence and semantic fidelity under adverse conditions.

[*]Work done during internships at Adobe Inc.

The First Workshop on Generative and Protective AI for Content Creation, 39th Conference on Neural Information Processing Systems (NeurIPS 2025).

We evaluate leading open- and closed-source models, including OmniGen2 [25], BAGEL [3] and GPT-4o [1] on SpotEdit. Results reveal that visually guided editing remains fundamentally challenging: the strongest open-source model achieves only 0.685 similarity score to ground truth. Moreover, models exhibit complementary strengths and weaknesses, e.g., OmniGen2 adheres closely to visual guidance but disrupts background consistency, while BAGEL preserves context yet struggles with cue interpretation. Strikingly, in *hallucination* cases, even the proprietary GPT-4o performs poorly, hallucinating object presence and executing incorrect edits despite its strong general image editing capabilities.

2 Related Work

In the following, we present related work on existing visually-guided image editing benchmarks. An extended discussion, including related work on the methods, is provided in App. A.

Visually-guided image editing benchmarks. As a pioneer, Paint by Example[22] introduced exemplar-based editing by inpainting image regions from a reference exemplar, focusing on visual similarity and identity preservation. DreamEdit [7] extended this to subject-driven editing, manipulating a subject's appearance or context while preserving identity. As shown in Fig. 4, prior benchmarks involve simple scenes, few distractors, and near-identical object poses across reference and edited images. In contrast, SpotEdit targets fine-grained, visually-guided editing in complex scenes with multiple objects, sparse textual descriptions, challenging spatial layouts, and pose variations. Its deterministic design with a ground truth enables objective evaluation and, for the first time, tests *hallucination* to incomplete or missing visual cues.

3 SpotEdit Benchmark

In this section, we outline our data generation pipeline, present benchmark statistics, and highlight key characteristics. Additional details are provided in App. B.

3.1 Data Generation Pipeline.

We construct the SpotEdit benchmark through a structured data generation pipeline that ingests keyframes extracted from video sources. Our data comes from two complementary datasets: (1) the StoryStream dataset [23], a large-scale, high-resolution synthetic multimodal collection designed for long-form story generation, and (2) real (non-synthetic) videos from NExT-QA [20], which add diversity and visual realism. While StoryStream provides keyframes directly, NExT-QA requires preprocessing to extract them, details of which are given in App. B. Starting with the key-frames, we then utilize our three-stage data generation process, illustrated in Fig. 1 and described below, to generate our benchmark:

Step 1: Instruction Generation. Both datasets include frame-level captions, which we use to prompt `Llama-3.1-8B-Instruct` to produce fine-grained instructions for object-level image editing. The model outputs an editing prompt, the target object, and the corresponding frames. This stage is entirely text-based, relying solely on the narrative captions; no visual inputs are required.

Step 2: Frame Localization. To refine the *Location* of the target object, i.e., specific frames where the target object appears, we query `InternVL3-8B` [28]. Each frame is individually evaluated by the model to determine the presence or absence of the target object.

Step 3: Consistent Editing. Once the target frames are identified, we perform image editing using the GPT-4o model. More specifically, the first image is edited without any visual guidance. While other frames are edited using the edited version of the first frame as the visual guidance to preserve edit consistency over all keyframes.

We construct benchmark samples by pairing each edit instruction with its corresponding source and edited images. In the standard setting, each sample consists of: a reference frame (providing visual guidance), a source frame (the image to be edited), a textual instruction, and a ground-truth edited frame (the target output). In the *hallucination* setting, we deliberately introduce cases where

[1] `https://openai.com/index/introducing-4o-image-generation/`

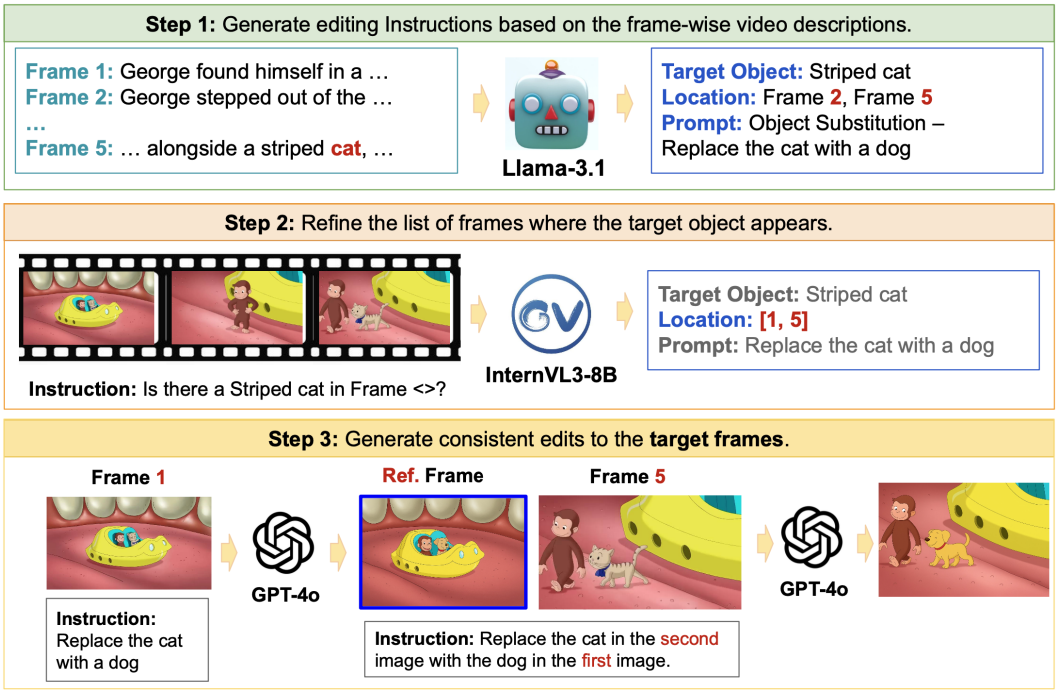


Figure 1: **SpotEdit data generation pipeline.** The pipeline consists of three key stages: (1) generating editing instructions from frame-wise video descriptions, (2) identifying target frames containing the specified object using multimodal queries, and (3) applying consistent edits only to the relevant frames utilizing Visually-guided image editing.

the target object is absent. Specifically, Ref. Robustness samples lack the target object from the reference image, while Inp. Robustness samples lack it from the source image.

3.2 Benchmark Statistics & key features

Our benchmark contains 500 samples spanning both synthetic and real images. As illustrated in Fig. 7, approximately 40% of the samples belong to the *hallucination* setting, while the remaining 60% correspond to the standard setting.

SpotEdit introduces several distinctive features. First, it explicitly evaluates *hallucination* by including cases where the source object is absent, requiring models to correctly refrain from editing. Second, instead of isolated images, it leverages video keyframes that capture diverse object poses, scales, and lighting conditions, making the task more realistic and challenging. Third, editing instructions are intentionally concise, pushing models to rely on visual grounding for consistency with the reference image. Finally, each task is paired with a nearly unique ground-truth target, allowing precise and unambiguous evaluation, unlike traditional benchmarks. Together, these design choices enable systematic assessment of both standard editing performance and robustness to *hallucination* scenarios.

4 Experimental Evaluations

In this section, we evaluate generative models on both *standard* and *hallucination*-focused samples from SpotEdit, analyzing their strengths and limitations. Further details are provided in App. C.

Evaluation Setup. For our baseline models, we include the most recent models that support visually-guided image editing. Specifically, UNO [19] and OmniGen [21] are diffusion-based models; BAGEL [3] and Emu2 [15] are autoregressive generative models; and OmniGen2 [25] adopts a hybrid architecture that couples an autoregressive text decoder with a diffusion-based image decoder. To compute semantic similarity, we use DINOv2 [12] (and later CLIP [14, 2] in Tab. 3 of

App. C) to extract image representations, as they have been widely adopted in prior works [19, 18]. Cosine similarity is then computed between the extracted representations, producing scores in the range $0 \leq \text{score} \leq 1$.

(a) *Instruction*: Replace the striped cat in the second image with the dog in the first image.

(b) Instruction: Match the writing on the man's shirt in the second image with the first image.

Figure 2: Two examples from the SpotEdit *standard* section. Each consists of a reference image, an input image, and an instruction, along with the edited outputs produced by baseline models.

4.1 Standard Evaluation

We first present results for the *standard* category of our benchmark.

Metrics. Qualitative results for both synthetic and real samples in the standard category are presented in Fig. 2 and Fig. 8. For quantitative evaluation, we measure performance across three complementary dimensions. The first, *Global Score*, provides a coarse-grained similarity measure, quantifying the alignment between the edited image and the corresponding ground-truth. The other two are fine-grained metrics: *Object Fidelity* and *Background Fidelity*. More specifically, *Object Fidelity* evaluates whether the model accurately follows the visual guidance from the reference image, preserving the target object's identity and appearance in the output. *Background Fidelity* measures the extent the model maintains the background of the source image while performing the edit.

Table 1: **Standard evaluation.** BAGEL and OmniGen2 emerge as the strongest performers. Moreover, fine-grained analysis uncovers that BAGEL achieves a strong *Background Fidelity* but struggles with visual guidance, while OmniGen2 follows guidance well but preserves backgrounds poorly.

Model	*Global Score*		*Background Fidelity*		*Object Fidelity*	
	Syn	**Real**	**Syn**	**Real**	**Syn**	**Real**
Emu2	0.531	0.543	0.458	0.411	0.567	0.414
OmniGen	0.380	0.252	0.340	0.283	0.391	0.223
UNO	0.535	0.425	0.435	0.371	0.511	0.328
BAGEL	**0.685**	0.611	**0.797**	**0.793**	0.455	0.327
OmniGen2	0.670	**0.617**	0.500	0.455	**0.719**	**0.590**

Evaluation Insights. Our results, illustrated in Tab. 1, reveal that visually-guided image editing remains a challenging task even for leading models: the maximum similarity score achieved does not exceed 0.685. Moreover, from the *Global Score*, BAGEL and OmniGen2 emerge as the strongest performers. However, fine-grained analysis uncovers notable differences in their strengths and weaknesses. BAGEL demonstrates strong *Background Fidelity* but struggles to follow the visual guidance, leading to lower *Object Fidelity*. Conversely, OmniGen2 excels at adhering to the reference image's guidance but exhibits weaknesses in background preservation. These observations are consistent with the qualitative patterns illustrated in Fig. 2 and Fig. 8. Finally, real samples appear to pose greater challenges than synthetic ones across almost all models and metrics. Further techincal details on the compuation of these metrics are provided in App. B.

4.2 *Hallucination* Evaluation.

In our benchmark, we include a dedicated section for *hallucination* evaluation, which tests scenarios where either the reference image or the input image does not contain the object specified in the

instruction. Representative examples from this section are shown in Fig. 3 and Fig. 9. Specifically, Fig. 3a shows a Ref. Robustness, while Fig. 3b illustrates a Inp. Robustness. In such situations, we expect the model to recognize the absence of the required object and output the unmodified input image.

(a) Instruction: Replace the bag in the second image with the backpack shown in the first image.

(b) Instruction: Replace the cat in the second image with the train in the first image.

Figure 3: Two examples from the SpotEdit *hallucination* section.

Evaluation Insights. The *hallucination* results reported in Tab. 2 indicate that this setting is substantially more challenging: all open-source models, except BAGEL, exhibit a marked performance drop relative to their *Global Score* in the standard evaluation (Table1). We also evaluate GPT-4o in this setting and observe that, despite its strong general capabilities in image editing, it introduces unintended modifications to the output image, which leads to a low similarity score.

A closer examination of the qualitative examples in Fig. 2 and Fig. 8 confirms that the models indeed produce hallucinations during the editing process. To better quantify hallucination-induced failures, we employ InternVL3-8B[28] as a binary classifier to assess the presence of the target object in the generated image. As shown in Tab. 2, the failure rates are alarmingly high, with GPT-4o emerging as the most vulnerable model in 2 out of the 4 *Failure Rate* evaluations.

Despite the overall poor performance across models, BAGEL outperforms all others on 6 out of 8 evaluation metrics. First, from the standard evaluation, we observe that BAGEL demonstrates strong background-preservation capabilities, which helps maintain a high *Global Score*. Moreover, BAGEL's unified design for both generation and understanding tasks equips it with stronger visual understanding capabilities for handling such challenging scenarios.

Table 2: ***Hallucination* evaluation.** While GPT-4o performs poorly, emerging as the most vulnerable model in 2 out of the 4 *Failure Rate* evaluations, BAGEL shows strong robustness.

Model	Inp. Robustness				Ref. Robustness			
	Global Score ↑		*Failure Rate* (%)↓		*Global Score* ↑		*Failure Rate* (%)↓	
	Syn	**Real**	**Syn**	**Real**	**Syn**	**Real**	**Syn**	**Real**
GPT-4o	0.710	0.550	81.2	91.7	0.745	0.599	75.5	72.0
Emu2	0.440	0.338	82.7	84.0	0.544	0.402	**54.0**	72.5
OmniGen	0.285	0.260	67.3	62.0	0.371	0.372	70.0	68.6
UNO	0.383	0.350	82.7	76.0	0.419	0.433	60.0	62.7
BAGEL	**0.867**	**0.845**	**61.5**	**56.0**	**0.880**	**0.735**	70.0	74.5
OmniGen2	0.466	0.331	84.6	88.0	0.577	0.513	64.0	**56.9**

5 Conclusion

We presented SpotEdit, a benchmark that brings realistic, fine-grained, and *hallucination*-focused evaluation to visually-guided image editing. Testing leading generative models reveals that while some excel in object fidelity or background preservation, none, even the leading closed-source model GPT-4o, achieve consistent performance across all scenarios, especially when visual cues are incomplete. By exposing these gaps, SpotEdit provides a clear path for advancing models that remain accurate and reliable under real-world editing challenges.

References

[1] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023.

[2] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia 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.

[3] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[4] Sara Ghazanfari, Siddharth Garg, Nicolas Flammarion, Prashanth Krishnamurthy, Farshad Khorrami, and Francesco Croce. Towards unified benchmark and models for multi-modal perceptual metrics. *arXiv preprint arXiv:2412.10594*, 2024.

[5] Hsin-Ping Huang, Yu-Chuan Su, and Ming-Hsuan Yang. Generating long-take videos via effective keyframes and guidance. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3709–3720. IEEE, 2025.

[6] Jinwoo Kim, Sangmin Han, Jinho Jeong, Jiwoo Choi, Dongyeoung Kim, and Seon Joo Kim. Orida: Object-centric real-world image composition dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3051–3060, 2025.

[7] Tianle Li, Max Ku, Cong Wei, and Wenhu Chen. Dreamedit: Subject-driven image editing. *arXiv preprint arXiv:2306.12624*, 2023.

[8] Dong Liang, Jinyuan Jia, Yuhao Liu, Zhanghan Ke, Hongbo Fu, and Rynson WH Lau. Vodiff: Controlling object visibility order in text-to-image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18379–18389, 2025.

[9] Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, et al. Uniworld: High-resolution semantic encoders for unified visual understanding and generation. *arXiv preprint arXiv:2506.03147*, 2025.

[10] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[11] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

[12] 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.

[13] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, A. 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 *ICML*, 2021.

[15] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024.

[16] Zeyi Sun, Ziyang Chu, Pan Zhang, Tong Wu, Xiaoyi Dong, Yuhang Zang, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. X-prompt: Towards universal in-context image generation in auto-regressive vision language foundation models. *arXiv preprint arXiv:2412.01824*, 2024.

[17] Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *The Thirteenth International Conference on Learning Representations*, 2024.

[18] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

[19] Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. *arXiv preprint arXiv:2504.02160*, 2025.

[20] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786, 2021.

[21] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13294–13304, 2025.

[22] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18381–18391, 2023.

[23] Shuai Yang, Yuying Ge, Yang Li, Yukang Chen, Yixiao Ge, Ying Shan, and Yingcong Chen. Seed-story: Multimodal long story generation with large language model. *arXiv preprint arXiv:2407.08683*, 2024.

[24] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26125–26135, 2025.

[25] Luowei Zhang, Hexiang Hu, Zhe Jiang, Yen-Chun Chen, Zhenfang Liu, Zhe Liu, Syed Shukor, Xin Eric Zhang, Kevin Lin, Vivek Natarajan, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2024.

[26] Shuheng Zhang, Yuqi Liu, Hongbo Zhou, Jun Peng, Yiyi Zhou, Xiaoshuai Sun, and Rongrong Ji. Adaflow: Efficient long video editing via adaptive attention slimming and keyframe selection. *arXiv preprint arXiv:2502.05433*, 2025.

[27] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025.

[28] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Extended Related Work

Image editing. Recent advances in image editing have introduced models that take an input image along with a textual edit instruction and produce an edited version of the image [1, 24, 11, 27, 9, 17]. However, to provide more concrete guides for image editing one can provide visual guides as wellas textual guide which lead to a more nuanced task called visually-guided image editing. These visual cues help convey richer, more precise editing intent, especially when consistency across multiple frames is required. For example, when editing keyframes in a video, maintaining temporal and stylistic coherence across frames is crucial—something that textual instructions alone cannot achieve. This motivates the use of visually-guided methods, which incorporate reference visuals to better preserve detail and ensure consistency. As expected the methods mentioned before are not capable of performing these tasks. A few recent models have been proposed to address this more complex task.

Visually-guided image editing methods. (visually-guided) Image editing models typically fall into three categories: diffusion models [21, 19], autoregressive models [15, 16, 3], and hybrid models [25, 13] that combine both approaches. Diffusion models including UNO [19] and OmniGen [21] use iterative denoising to produce high-quality images, with UNO focusing on visually-guided controllability and OmniGen unifying tasks like text-to-image generation and editing. Autoregressive models, including BAGEL [3], X-Prompt [16] and Emu2 [15] generate content sequentially via next-token prediction. Bridging the two paradigms, OmniGen2 [25] adopts a hybrid architecture that couples an autoregressive text decoder with a diffusion-based image decoder, allowing for specialized yet coordinated multimodal generation.

Figure 4: Early Visually-guided image editing benchmarks.

Visually-guided image editing benchmarks. Paint by Example[22] introduced exemplar-based editing by inpainting image regions from a reference exemplar, focusing on visual similarity and identity preservation. DreamEdit [7] extended this to subject-driven editing, manipulating a subject's appearance or context while preserving identity. As shown in Fig. 4, prior benchmarks involve simple scenes, few distractors, and near-identical object poses across reference and edited images.

Unlike previous benchmarks, our work focuses on fine-grained visually-guided image editing, where both the reference and input images contain multiple, distinct objects and exhibit complex spatial structures. Editing models must first detect and identify the target object in the reference image, then accurately locate and modify the corresponding object in the input image; all while preserving the object's identity and maintaining the overall visual coherence of the scene. Another key distinction of our benchmark is its use of minimal prompts: unlike prior datasets that describe the expected final image in detail, our approach relies heavily on visual cues from the reference image, requiring models to reason more deeply about visual context. This makes our benchmark particularly challenging, as it demands both compositional consistency and semantic reasoning across multiple images. Furthermore, in contrast to other editing tasks that allow for open-ended outputs, our benchmark is deterministic—each editing task has a unique, ground-truth result. This enables precise and objective evaluation of model performance.

A parallel line of research [18, 8, 6] has focused on image composition tasks, where a new image is synthesized by combining elements from multiple context images. Unlike visually-guided editing, these works do not involve editing a specific input image while preserving its structure and style; instead, they aim to generate a novel composition based on the contextual images alone.

B SpotEdit Details

In this section, we first explain the preprocessing step i.e., extracting the keyframes from the real videos video. Moreover, we provide the prompts used in each step of the pipeline to query the generative model.

Keyframe Extraction We construct the SpotEdit benchmark using a structured data generation pipeline, as illustrated in Fig. 1. The pipeline takes as input keyframes extracted from various video sources. For synthetic datasets such as StoryStream[23], keyframes are provided directly. For the ShareGPT-4[20] dataset, we employ CLIP-ViT-H-14 [14, 2] combined with cosine similarity to measure the similarity between consecutive frames. Specifically, the selection process begins with the first frame and progressively includes subsequent frames that have a similarity score less than 0.8 compared to the most recently selected keyframe.

Data generation pipeline. In the first step of the pipeline, the frame captions are provided to the `Llama-3.1-8B-Instruct`[2], which generates the corresponding edit instructions. The prompt used in this step is shown in Fig. 5. For Steps 2 and 3, the instructions are shown in Fig. 1.

After running the data generation pipeline, we obtain consistent edits applied to the target frames, as illustrated in Fig. 6. The frames containing the specified object (e.g., Frame 1 and Frame 5) form the standard samples. The remaining frames (e.g., Frame 2), where the specified object is absent, form the *hallucination* samples. We further divide the *hallucination* samples into two subcategories:

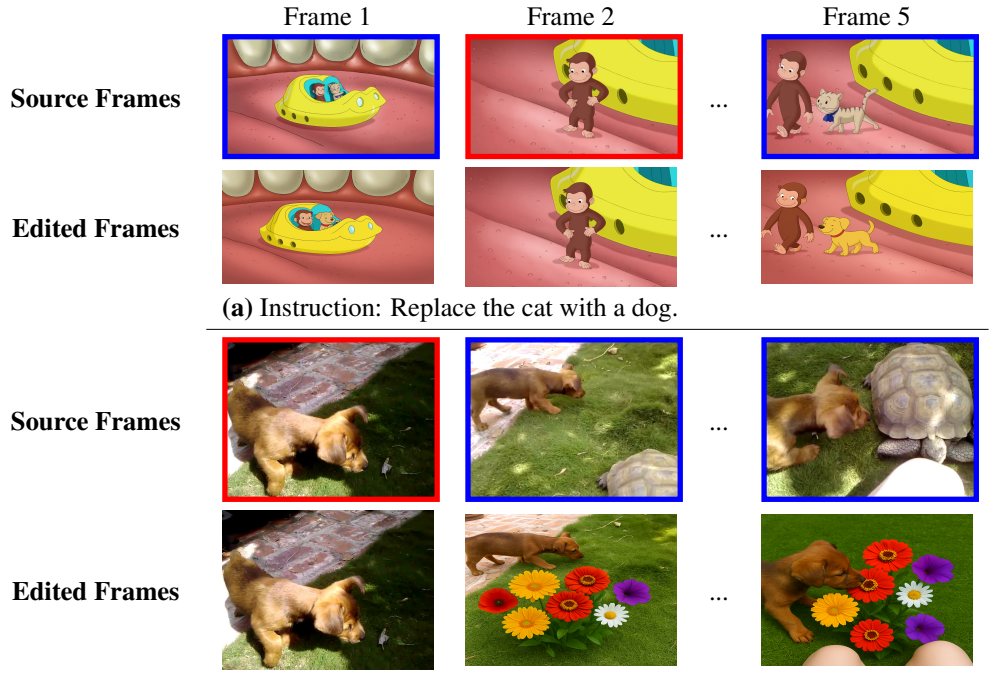
- Inp. Robustness: The reference image contains the target object (e.g., Edited Frame 1 or 5), but the input image does not contain the source object (e.g., Frame 2).
- Ref. Robustness: The reference image does not contain the target object (e.g., Frame 2), but the input image contains the source object (e.g., Frame 1 or Frame 5).

Step 1 Prompt

```
You will be provided with a detailed description of a video,
    including frame-by-frame events.


Your task is to generate a clear, concise, and specific editing
    instructions based on this description.
Focus exclusively on the following types of edits:

- Object Modification (e.g., adjusting color, shape, size, or type)
- Object Removal
- Object Substitution

It is essential that you clearly identify the target object,
    including any distinguishing features, location, or context, so
    that it can be reliably recognized and edited.
Response should contain:
* Target Object:
* Location:
* Instruction:
```

Figure 5: **Step 1 prompt.**.

[2] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`



(a) Instruction: Replace the cat with a dog.

(b) Instruction: Replace the tortoise with a bunch of flowers.

Figure 6: Examples generated by our data pipeline. Target frames (blue) are used to construct the *standard* section of the benchmark, while untargeted frames (red) are used to construct the *hallucination* section of our benchmark.

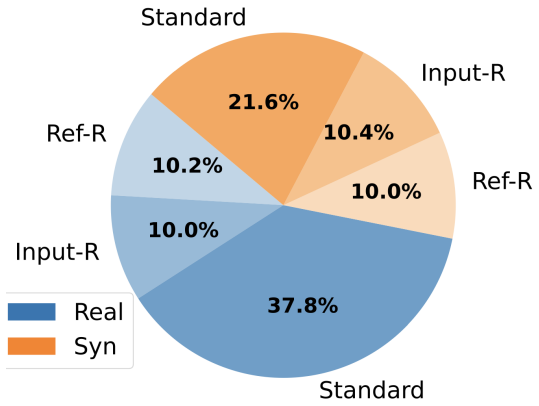
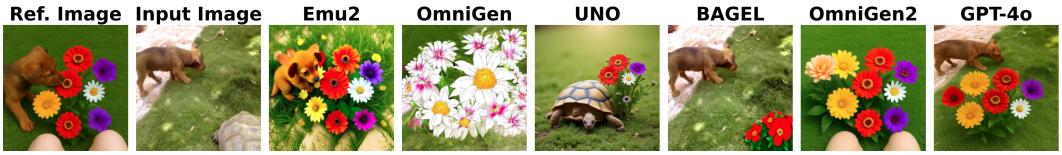
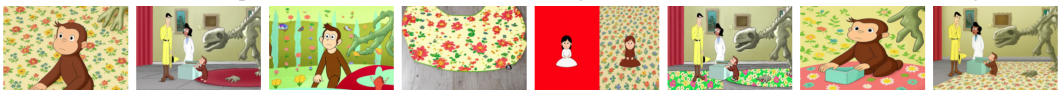


Figure 7: Statistics of the SpotEdit benchmark across 500 samples.

Additional qualitative examples. We provide further qualitative examples from the SpotEdit benchmark. Fig. 8 presents additional cases from the standard section, while Fig. 9 showcases representative samples from the *hallucination* section of the benchmark.



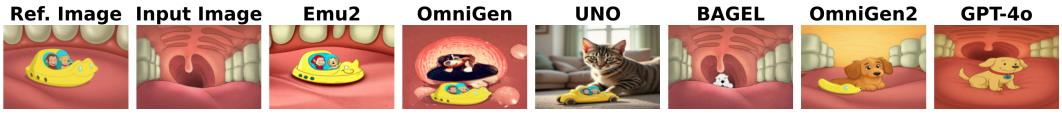
(a) Instruction: Replace the tortoise in the second image with the flowers shown in the first image.



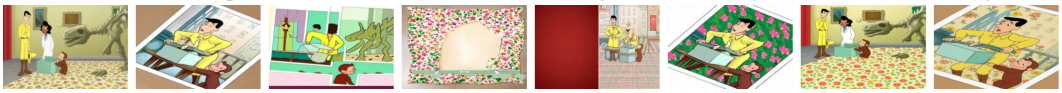
(b) Instruction: Replace the red surface in the second image with a flowery surface in the first image.

(c) *Instruction*: Replace the cat in the second image with the train in the first image.

Figure 8: Examples from SpotEdit *standard* section.



(a) Instruction: Replace the red surface in the second image with a flowery surface in the first image.



(b) Instruction: Replace the red surface in the second image with a flowery surface in the first image.

(c) Instruction: Replace the green fish in the second image with the boat in the first image.

Figure 9: Examples from SpotEdit *hallucination* section.

C Evaluations Details

C.1 Evaluation metrics

- ***Global Score***: As discussed in Section 3, each benchmark sample comprises four components: a reference image (providing visual guidance), a source image (to be edited), a textual prompt, and a ground-truth edited frame as the target output. The target outputs are first generated using GPT-4o and then refined through human supervision to ensure accuracy. To compute the *Global Score*, we measure the similarity between the edited image produced by baseline models and the ground-truth target output. It is important to note that *Global Score* is the only metric that relies on the ground-truth annotations; all subsequent metrics depend solely on the reference and input images for evaluation.
- ***Background Fidelity***: As a more fine-grained evaluation, we assess the model's ability to preserve the background of the input image while performing the required edit and generating the edited (or output) image. To conduct this evaluation, we first employ GroundingDINO [10] to generate bounding boxes around the source object in the input image and the target object in the output image. We then mask out these bounding boxes and compute the similarity score exclusively on the remaining background regions.
- ***Object Fidelity***: As a complementary evaluation to *Background Fidelity*, we assess how well the model preserves the identity and appearance of the target object while following the visual

guidance from the reference image and applying the edit to the input image. To this end, we employ GroundingDINO [10] to extract the target object from both the reference image and the output image. Once isolated, we apply a similarity metric between the two cut-out objects to evaluate how closely they align.

- ***Failure Rate***: This metric specifically evaluates hallucination-induced failures. As discussed in Section 4.2, for both Inp. Robustness and Ref. Robustness samples, the expected behavior is that the output image should remain identical to the input image, meaning that the target object must not be added during editing. To assess whether models avoid such unintended modifications, we employ the multimodal LLM InternVL3-8B as a binary classifier to determine whether the target object appears in the edited image. Since this is a binary classification task, the *Failure Rate* is reported as accuracy.

C.2 Comparison to DreamEdit

As discussed in Section 2 and shown in Fig.4, prior benchmarks typically involve simple scenes, few distractors, and nearly identical object poses across the reference and edited images. To quantitatively assess the increased difficulty of our benchmark compared to the previously proposed DreamEdit benchmark[7], we evaluate models on DreamEdit's object replacement task, which consists of 198 samples. We use these samples to generate edited images with baseline models and then evaluate their performance using the metrics *Global Score*, *Background Fidelity*, and *Object Fidelity*.

For image similarity computation, we employ the CLIP-ViT-H-14 model [14, 2] to extract image representations, given its strong performance across a wide range of semantic similarity tasks [4]. Cosine similarity is then applied to the representations, yielding scores in the range $0 \leq \text{score} \leq 1$.

The results are presented in Tab. 3. As mentioned earlier, SpotEdit provides ground-truth outputs, enabling us to compute *Global Score*. In contrast, the DreamEdit benchmark lacks this property, and thus *Global Score* cannot be measured. However, since *Background Fidelity* and *Object Fidelity* rely only on the reference and input images, they remain computable for DreamEdit. Across all models and metrics, except for a single case, scores on SpotEdit are consistently lower, indicating that it presents a more challenging and complex task than DreamEdit.

Table 3: DreamEdit benchmark complexity compared to our SpotEdit (our) benchmark.

Model	***Global Score***		***Background Fidelity***		***Object Fidelity***	
	DreamEdit	SpotEdit	DreamEdit	SpotEdit	DreamEdit	SpotEdit
Emu2	-	0.639	0.679	0.587 ↓0.09	0.643	0.575 ↓0.07
OmniGen	-	0.429	0.730	0.497 ↓0.23	0.592	0.431 ↓0.16
UNO	-	0.575	0.599	0.503 ↓0.10	0.574	0.528 ↓0.05
BAGEL	-	0.672	0.909	0.828 ↓0.08	0.559	0.496 ↓0.06
OmniGen2	-	0.719	0.735	0.562 ↓0.17	0.679	0.697 ↑0.02