Language-Guided Fine-Grained Text-Guided Image Editing with Diffusion Models

Chao-Jung Lai chl299@ucsd.edu
Eddy (Yi-Ting) Chu yic147@ucsd.edu
Ming-Kai Liu mil151@ucsd.edu
Yen-Ting Lee yel050@ucsd.edu

Table 1: GPT-4o Text-guided editing example

Source

Target

Prompt: "Help me remove the guy at the rightmost corner of the image who's wearing a black top and pants."

Observation: The model removes the specified figure, but also changes the central subject's face and alters background semantics.

1 Introduction

Text-guided image editing has grown increasingly popular, largely due to the accessibility of natural language interfaces, as demonstrated by models like GPT-4 (Brown et al., 2020). Despite this convenience, existing approaches often struggle with fine-grained control. Rather than confining edits to a specific object or region, they frequently alter the entire image—affecting global structure, style, or semantics. (See the GPT-generated output above table 1; image credit: my friend's thread post, which inspired this work.) While segmentation masks provide a mechanism for spatial precision, manually drawing them remains labor-intensive and impractical for most users. Recent attempts to address this limitation, such as DiffEdit (Couairon et al., 2022) and SmartBrush (Xie et al., 2022), aim to combine textual guidance with region-based control. However, these methods still face challenges, including inaccurate mask localization and weak alignment between text prompts and visual edits.

To address these limitations, we present a modular, heavily engineered pipeline for fine-grained, language-guided image editing. Given an input image and a natural language prompt (e.g., "replace the dog on the man's lap with a cat"), our system performs: (1) object mask generation, (2) spatial description via LLaVA, (3) inpainting the background using Stable Diffusion XL, (4) predicting the replacement object's mask, and (5) final inpainting to insert the new object. This approach provides precise, controllable edits with strong visual coherence.

Due to an unexpected loss of access to critical GPU resources, we were unable to finish fine-tuning a custom satisfying mask generator. However, inspired by the promising direction shown in Adobe's recent work (Singh et al., 2023), we believe that future development in mask generation will further enhance this pipeline. For proof-of-concept, we include results using hand-crafted masks that simulate predicted outputs, and we will continue to work on it.

2 Related work

2.1 Diffusion Models for Image Generation and Editing

Diffusion probabilistic models (DDPM) (Ho et al., 2020) generate images by modeling the reverse of a gradual noise process. In the forward process, noise is incrementally added to an image:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I),$$

while the model learns to reverse this corruption through a denoising network:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)).$$

This foundational mechanism has enabled a wide range of image generation and editing applications. DDIM (Song et al., 2022) builds on DDPM by introducing a deterministic, non-Markovian variant of the reverse process, making it well-suited for controllable and efficient sampling in editing tasks.

Leveraging this framework, InstructPix2Pix (Brooks et al., 2023) introduces a model that performs semantic edits by conditioning on textual instructions, effectively enabling instruction-following behavior in a single denoising pass. DiffEdit (Couairon et al., 2022) extends DDIM's capabilities by computing semantic difference masks between source and target prompts, guiding the editing process via masked inpainting. While effective, DiffEdit still inherits some limitations of global noise propagation, occasionally altering unintended regions.

To improve spatial precision, newer models like DiffEditor (Mou et al., 2024) and InstDiffEdit (Zou et al., 2024) incorporate attention-based localization and region-specific score guidance. These advances make the editing process more controllable and accurate, particularly for fine-grained modifications.

Commercial systems like ChatGPT's image editing interface—powered by the GPT-4o or DALL·E 3 backend—also rely on diffusion pipelines. However, such systems often edit entire images unless users explicitly define regions via masking tools in the UI (Wiggers and Zeff, 2025), underscoring the need for finer control in practical applications.

2.2 Inpainting and Fine-Grained Editing with Masks

Mask-guided inpainting models like Stable Diffusion Repaint (Lugmayr et al., 2022) restrict diffusion to specified regions:

$$x_{t-1} = M \odot \text{Denoise}(x_t, y) + (1 - M) \odot x_t,$$

where M is a binary mask and y is a text prompt. This avoids modifying unmasked areas.

Google's Imagen and its evaluation benchmark EditBench (Wang et al., 2023) support high-fidelity editing within masked regions. Further work like CAT-Diffusion (Zeng et al., 2024) extends inpainting to attribute transfer.

Other notable methods include: - Inpainting (LaMa-series) (Lugmayr et al., 2022), - SmartBrush (closed-source), - Inpaint Anything (SAM+inpainting, GitHub), - HD-Painter, TF-ICON, and task-prompt guided inpainting frameworks.

2.3 Phrase Grounding and Interactive Segmentation

Fine-grained editing relies on isolating objects. Phrase grounding methods like Grounded DINO (Liu et al., 2024) map noun phrases to image regions, while SAM (Kirillov et al., 2023) provides potent mask proposals. The combination (Grounded-SAM) (Ren et al., 2024) enables prompt-conditioned mask extraction—e.g., "dog on a couch."

SAM-CP (Chen et al., 2025) further enhances mask precision by composing prompts, aligning closely with our use of Grounded-SAM to generate removal masks.

2.4 Visual Understanding for Prompt Grounding

Describing object location and context is crucial. Image captioning and visual description models (e.g., BLIP (Li et al., 2022), LLaVA) translate image regions into natural language, enabling spatial understanding.

In our pipeline, we use LLaVA to generate positional prompts ("on the lap behind the man's hand") for subsequent mask prediction and inpainting guidance.

3 Method and Experiments

Our final pipeline emerged through a series of incremental experiments conducted over the course of the quarter. In this section, we present these developments in chronological order to convey the underlying intuition and iterative decision-making that shaped the project.

3.1 Problem Statement

Table 2: Example of an editing instruction and the corresponding input image.

Instruction: Change the **dog** that the man is holding into a fluffy orange **cat**.

Language-guided fine-grained image editing is an inherently ill-posed and under-defined task. The umbrella of "image editing" encompasses a vast range of goals—including style transfer, object relocation, content dragging, object replacement, resizing, pasting, and more. In this work, we narrow our focus to the task of *object replacement*, which we argue is a generalizable formulation that subsumes many of the aforementioned tasks. For instance, object resizing can be viewed as replacing an object with a larger or smaller version of itself. Similarly, content dragging may be modeled as removing an object, shifting its position, and re-generating it with contextual consistency. An illustrative example is shown above see table 2.

3.2 System Design and Ideation

Our final pipeline emerged through iterative experimentation, guided by observations of failure cases and analysis of prior work. In this section, we walk through the key design choices that shaped our system, grouped by the editing strategy they represent.

Instruction-conditioned full-image editing. One popular approach for image editing is to condition a diffusion model on a natural language instruction and synthesize a new image end-to-end. Models such as InstructPix2Pix (Brooks et al., 2023) and the DALL·E backend in GPT-4o adopt this paradigm. These methods demonstrate strong semantic alignment with instructions but often apply global changes to the image—both structural (e.g., layout, pose, spatial configuration) and stylistic (e.g., color tone, saturation, softness, or sharpness).

Table 3: Visualizing global changes from instruction-conditioned editing. The left is the edited result, and the right shows a side-by-side comparison of specific image details, such as the logo on the hat and the texture of the chair. Nearly all pixels are subtly altered, revealing global changes beyond the intended edit.

Edited Target (Cat)

Overlay: Source vs. Target

This limitation is clearly illustrated in Table 3, where an instruction like "replace dog with cat" causes the model to redraw large portions of the scene. Such behavior, while effective for general transformations, is problematic for fine-grained, object-specific editing. This insight motivated our exploration of localized, mask-guided editing methods.

Table 4: DiffEdit applied to a dog→cat edit. The generated mask is localized but imprecise; the result is semantically weak.

Generated Mask

Edited Output

Mask-guided diffusion with zero-shot localization (DiffEdit). To constrain edits to localized regions, we investigated DiffEdit (Couairon et al., 2022), which generates a mask indicating the region to be edited. The mask is derived by computing the normalized difference between predicted noise vectors conditioned on a reference prompt R (e.g., "dog") and a query prompt Q (e.g., "cat") across multiple noisy samples. A threshold is applied to this difference map to produce a binary mask M.

This mask is then used to constrain the denoising trajectory: a DDIM schedule encodes the image into an intermediate latent $\mathbf{x}_r$, and decoding

is performed with mask-wise correction, blending edited and preserved regions. As shown in Table 4, the mask is spatially constrained, but imprecise—leaking into the background and failing to cleanly capture object boundaries. The final result partially resembles the target object but lacks full semantic fidelity.

Table 5: RES + Gaussian dilation enables more plausible replacements when target objects are spatially incompatible with the source.

Original Source Mask

Edited Output (w/o Dilation)

Dilated Source Mask

Edited Output (w/ Dilation)

Referring Expression Segmentation (RES) for accurate source localization. To improve mask accuracy, we adopted referring expression segmentation (RES) via Grounded-SAM, allowing us to generate precise source object masks from a natural language phrase (e.g., "the dog on the couch"). This enabled more reliable localization than DiffEdit's implicit mask.

However, RES-based editing still faces challenges when the source and target objects differ significantly in shape or size (e.g., dog → mouse or TV). A naïve RES mask may be insufficient to reserve enough space for the target. To partially address this, we follow Zhuang et al. (Zhuang et al., 2024) and apply Gaussian dilation to expand the source mask. Table 5 shows an example where dilation yields better spatial accommodation and allows for more plausible object replacement. Nonetheless, this approach is not entirely language-guided. The generated mask with careless hyperparameter can cause nullify the benefit of extracting precise mask from SAM in the first place.

This approach constitutes our first zero-shot pipeline—segmenting the source with RES see Figure 1, dilating the mask, and applying a pretrained diffusion inpainting model. While effective in some cases (e.g., dog → cat), this strategy still fails when the target object's spatial footprint deviates drastically from the source.

Table 6: Failure cases for the zero-shot pipeline

Instruction: Replace the woman with a giant elephant.

Instruction: Replace the yellow car with a mouse sitting on the ground.

Prompt-Tuned Two-Stage Editing. InstDiffEdit (Zou et al., 2024) proposes generating fine-grained masks on-the-fly by extracting cross-attention maps from diffusion models. This attention-based localization demonstrates significantly higher mask precision and editing quality compared to DiffEdit, as shown in their benchmarks. However, despite its elegance, such approaches still face fundamental limitations when the source and target objects differ substantially in size or semantics. In these cases, even attention-based masks tend to over-segment, introducing editing artifacts or context mismatch.

Motivated by this, we adopt a two-stage pipeline that **decouples *object removal* from *object insertion***. Our key insight is that state-of-the-art diffusion models, such as Stable Diffusion XL, are remarkably effective at inpainting when provided with well-localized masks and care-

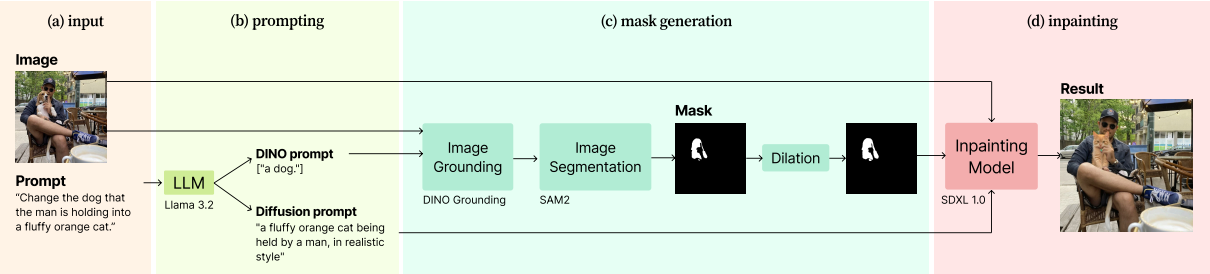

Figure 1: Overview of our zero-shot editing pipeline. (a) Given an input image and text prompt, (b) an LLM (LLaMA 3.2) interprets the prompt and generates two outputs: a grounding prompt for identifying the source object (e.g., "a dog") and a diffusion prompt describing the target (e.g., "a fluffy orange cat being held by a man, in realistic style"). (c) The system then performs image grounding and segmentation to produce a source object mask, which is further dilated to provide more flexibility. (d) Finally, a pretrained diffusion inpainting model (e.g., SDXL) generates the edited output.

fully crafted prompts. We leverage LLaVA with prompt-based conditioning (see Appendix A for prompt details) to automatically generate localized environmental descriptions (e.g., "the grass under the dog" or "a person standing on a sidewalk"), which are then used to guide background restoration after source object removal. The full pipeline can be found at Figure 2

Once a clean background is synthesized, we use a secondary prompt to describe the target object and perform targeted inpainting at a new or resized location. This avoids the inherent ambiguity of editing large-to-small object replacements using a single-step mask.

This two-step strategy allows for more interpretable, prompt-driven control and eliminates the need for fragile semantic difference masks. Furthermore, it generalizes better across object scales and categories, as shown in Figure 4.

Automated Mask Generation. Our two-stage editing pipeline relies on a target mask generator that can understand the global scene context and localize the desired insertion region based on a target prompt. To this end, we first use LLaVA to describe the location and context of the removed source object. This spatial guidance is then used to constrain the generation of a corresponding target mask in the same vicinity. The formulation can be found in the equation below:

$$f(I_{\text{bg}}, p_{\text{tgt}}) = M_{\text{tgt}} \tag{1}$$

where f denotes the mask generator, I_{bg} is the source-removed image, p_{tgt} is the target prompt, and M_{tgt} is the resulting target mask.

To realize this capability, we fine-tune a diffusion model using ControlNet, conditioning on the source-removed image and the target prompt. A similar strategy was recently explored in Adobe's proprietary system, which further supports the feasibility of this pipeline. While we were unable to complete the training of a high-quality generator due to time constraints, we constructed a large-scale dataset specifically for this task. The next section outlines our data generation pipeline.

Dataset Construction. We build our training dataset on top of SEED (AILab-CVC, 2023), a curated collection of over 50,000 human-edited, Photoshop-style image pairs. For each edited image, we first apply LLaVA to extract object-level descriptions corresponding to prominent visual elements. These descriptions are passed to Grounded-SAM to generate a segmentation mask of the referred object.

Next, we use our Stable Diffusion XL-based background removal pipeline to erase the source object while preserving contextual realism. Each resulting datapoint comprises: (1) a background-removed image I_{bg}, (2) a natural language target prompt p_{tgt}, and (3) the corresponding target mask M_{tgt}. To scale data generation efficiently, we distributed the process across 10 compute nodes, each equipped with at least 24 GB of GPU memory. See Table 7 for examples of data points in the generated dataset.

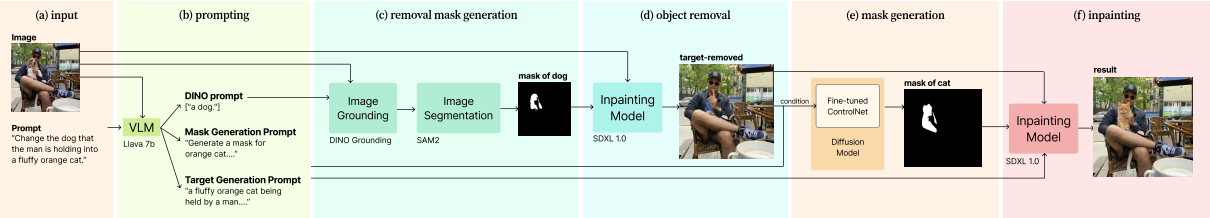

Figure 2: Overview of our two-stage editing pipeline. (a) Given an input image and prompt, (b) a vision-language model generates three types of prompts: a grounding prompt for identifying the source object, a target generation prompt describing the object to be inserted, and a mask generation prompt for predicting where the target object should appear. (c) The system performs grounding and segmentation to obtain a source object mask. (d) An inpainting model removes the source object based on the mask and reconstructs the background in the missing area. (e) A separate mask is generated for the target object by a prompt-conditioned mask generation model pretrained with ControlNet, which predicts the appropriate region for insertion. (f) Finally, an inpainting model inserts the new object into the predicted region.

Table 7: Examples from our training dataset. Each datapoint consists of an original image, the background-removed version, and the associated target prompt.

Original Image **Source Removed**

Target Prompt: "Generate a mask for tail feathers, the tail feathers are in the foreground of the image, with water droplets splashing around them, and a blurred background."

Original Image **Source Removed**

Target Prompt: "Generate a mask for sunglasses, the sunglasses are worn by a person with a yellow and black furry face, who is standing in front of a blue and pink background with a purple and pink neon sign."

3.3 Benchmark

Given our problem formulation—fine-grained object replacement—we design a benchmark to evaluate a model's ability to plausibly replace one object with another while preserving global coherence. Specifically, we assess performance using three complementary metrics: FID, Local FID, and CLIP directional similarity. Our benchmark protocol is adapted from DiffEdit (Couairon et al., 2022), and uses two validation datasets: COCO2017-val (5,000 images) and OpenImages validation (41,620 images).

For each image, we first detect all object instances and their categories using a pretrained object detector. Two instances are then randomly selected—one as the source object to be replaced, and one providing the target category. These are chosen from different semantic classes, and a fixed random seed ensures that all methods are evaluated on the same source/target pairs.

The editing model receives the original image, the bounding box or mask of the source object, and the desired target category. Its task is to synthesize a new object of the target class in place of the source, with seamless integration into the surrounding context.

We quantitatively evaluate the generated results as follows: (1) **FID** measures the distributional distance between edited images and real images of the target class, assessing overall visual realism. (2) **Local FID** is computed over the masked (edited) region only, focusing on localized fidelity of the replacement. (3) **CLIP Directional Similarity** computes the semantic alignment between the intended transformation (e.g., "cat $\rightarrow$ dog") and the actual visual difference using CLIP embeddings; higher scores reflect better target compliance.

Since prior works did not release their evaluation splits, we re-run all baselines using our fixed source/target pairings to ensure a fair and consis-

tent comparison.

4 Results

We evaluate our proposed pipelines both quantitatively and qualitatively. Our experiments compare the performance of DiffEdit and our zero-shot pipeline across two benchmark datasets: COCO and OpenImages. In addition, we present proof-of-concept results from our two-stage editing pipeline with manually defined target masks.

4.0.1 Quantitative Evaluation

Table 8 summarizes the performance of each method using average CLIP directional similarity, FID, and Local FID. Our zero-shot pipeline significantly outperforms DiffEdit across all metrics, suggesting stronger semantic alignment and higher perceptual quality. However, we note that the absolute CLIP scores are lower than those reported in the original DiffEdit paper.

This discrepancy arises because the original benchmark splits were not publicly released. As a result, we generate our own evaluation set using a fixed random seed. Despite this, certain source-target pairs are inherently difficult (e.g., replacing "flower" with "burrito" or "airplane" with "saxophone"), creating semantically and spatially mismatched editing scenarios. These challenging cases likely fall outside the distribution seen during pretraining for most diffusion models, leading to lower overall scores.

Table 8: Quantitative comparison between DiffEdit and our zero-shot pipeline. CLIP is reported per dataset; FID and Local FID are averaged across all samples.

Metric	DiffEdit	Zero-shot (Ours)
CLIP (COCO)	0.2073	0.2292
CLIP (OpenImages)	0.1776	0.2310
FID (Combined)	199.6818	37.8510
Local FID (Combined)	259.6376	78.4503

4.0.2 Qualitative Evaluation

(a) replace the meal with waffle

(b) replace the airplane with a saxophone

(c) replace the flower with a burrito

(d) replace the man with a girl

(e) replace the beer with a pencil sharpener

(f) replace the monkey with a door handle

Figure 3: Representative qualitative results from our zero-shot editing pipeline.

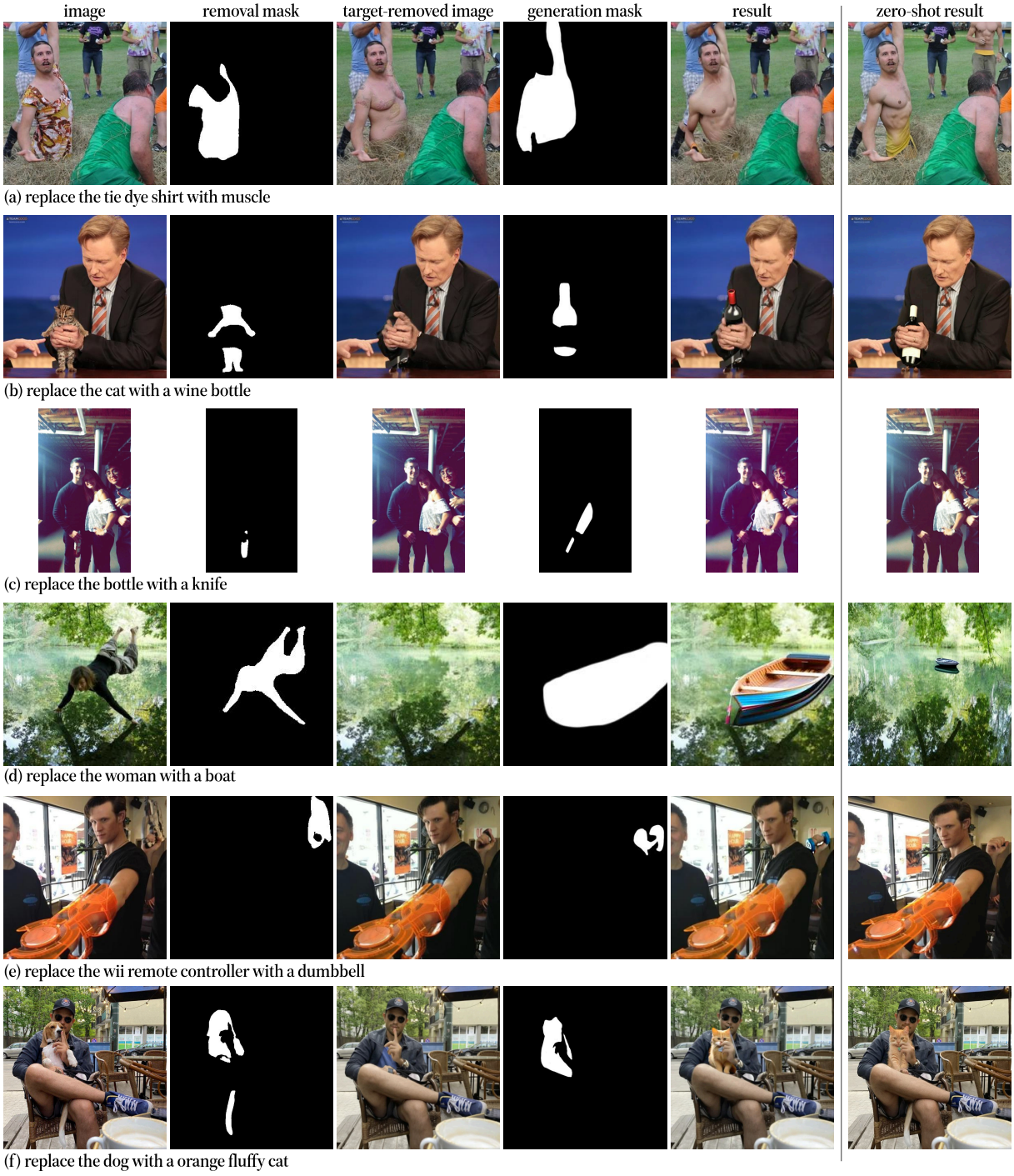

Figure 4: Representative qualitative results from our two-stage editing pipeline.

We present qualitative results for both our zero-shot and two-stage editing pipelines. The **zero-shot pipeline** performs effectively when the source object is relatively large and the target object is semantically and spatially aligned. Several representative outputs from this setting are shown in Figure 3. However, the method struggles in cases involving extreme scale mismatches or abstract category substitutions.

For example, when replacing an airplane with a saxophone, the model attempts to fit the saxophone into the unusual shape of the airplane, leading to visually implausible results. Similarly, substituting a monkey with a door handle—where the target object is significantly smaller—yields poor background blending and spatial coherence.

In contrast, our **two-stage pipeline**, though still a proof-of-concept, demonstrates improved robustness and visual consistency. By explicitly decoupling source object removal and target insertion, it allows for better spatial control and cleaner inpainting, particularly in complex editing scenarios. Figure 4 shows examples where small objects such as a knife or dumbbell are successfully added to a person's hand—something the single-stage approach struggles with due to discrepancies between the source and target object footprints.

References

AILab-CVC (2023). Seed: Semantic editing with editable descriptions. https://huggingface.co/datasets/AILab-CVC/SEED-Data-Edit-Part2-3. Accessed: 2025-06-10.

Brooks, T., Holynski, A., and Efros, A. A. (2023). Instructpix2pix: Learning to follow image editing instructions.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Chen, P., Xie, L., Huo, X., Yu, X., Zhang, X., Sun, Y., Han, Z., and Tian, Q. (2025). Sam-cp: Marrying sam with composable prompts for versatile segmentation.

Couairon, G., Verbeek, J., Schwenk, H., and Cord, M. (2022). Diffedit: Diffusion-based semantic image editing with mask guidance.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment anything.

Li, J., Li, D., Xiong, C., and Hoi, S. (2022). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., and Zhang, L. (2024). Grounding dino: Marrying dino with grounded pre-training for open-set object detection.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Gool, L. V. (2022). Repaint: Inpainting using denoising diffusion probabilistic models.

Mou, C., Wang, X., Song, J., Shan, Y., and Zhang, J. (2024). Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing.

Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., and Zhang, L. (2024). Grounded sam: Assembling open-world models for diverse visual tasks.

Singh, J., Zhang, J., Liu, Q., Smith, C., Lin, Z., and Zheng, L. (2023). Smartmask: Context aware high-fidelity mask generation for fine-grained object insertion and layout control.

Song, J., Meng, C., and Ermon, S. (2022). Denoising diffusion implicit models.

Wang, S., Saharia, C., Montgomery, C., Pont-Tuset, J., Noy, S., Pellegrini, S., Onoe, Y., Laszlo, S., Fleet, D. J., Soricut, R., Baldridge, J., Norouzi, M., Anderson, P., and Chan, W. (2023). Imagen editor and editbench: Advancing and evaluating text-guided image inpainting.

Wiggers, K. and Zeff, M. (2025). Chatgpt's image-generation feature gets an upgrade. *TechCrunch*. ChatGPT can now leverage the company's GPT-4o model to natively create and modify images and photos.

Xie, S., Zhang, Z., Lin, Z., Hinz, T., and Zhang, K. (2022). Smartbrush: Text and shape guided object inpainting with diffusion model.

Zeng, J., Song, D., Nie, W., Tian, H., Wang, T., and Liu, A. (2024). Cat-dm: Controllable accelerated virtual try-on with diffusion model.

Zhuang, J., Zeng, Y., Liu, W., Yuan, C., and Chen, K. (2024). A task is worth one word: Learning with task prompts for high-quality versatile image inpainting.

Zou, S., Tang, J., Zhou, Y., He, J., Zhao, C., Zhang, R., Hu, Z., and Sun, X. (2024). Towards efficient diffusion-based image editing with instant attention masks.

A Prompt Templates

We constructed prompts through an iterative process of prompt refinement. The final versions include role prompting, instructional constraints, and exemplar-based prompting.

A.1 Zero-shot Editing Pipeline

We use LLaMA 3.2 to extract structured inputs from natural language instructions to guide Grounding DINO and the diffusion model.

```
You are an AI assistant that processes user
    instructions for image editing tasks.
Given a user's prompt, extract and format the
    necessary inputs for two models: Grounding
    DINO and a diffusion model.

Your output must be a JSON object with the
    following two fields:

1. "dino_input": list of objects to detect or mask.
  - Format:
    - Single object: ["a cat."]
    - Multiple objects: [["a face.", "a car."]]
  - Rules:
    - Lowercase
    - Singular
    - Start with "a "
    - End with "."

2. "diffusion_input": a descriptive sentence of the
    desired edited image region.
  - Focus on what should appear in the masked area.
  - Natural language description, e.g., "a fluffy
      orange cat being held by a man."

Example Input:
User: "Change the dog that the man is holding into
    a fluffy orange cat."

Example Output:
{{
  "dino_input": ["a dog."],
  "diffusion_input": "a fluffy orange cat being
      held by a man."
}}

Output only the JSON. Do not include any
    explanations or markdown.
```

A.2 Dataset Generation Prompt and Fine-Tuning Conditioning Prompt

Given a dataset image, we use LLaVA to identify removable objects and generate the corresponding prompts for each object. Specifically, for each detected object, LLaVA produces: (1) the object name, (2) a Grounding DINO prompt for object segmentation, and (3) a mask generation prompt describing the object that should be subsequently restored in the removed region, where we explicitly describe the object's location, pose, and contextual surroundings. This serve as conditioning inputs for ControlNet-based mask generation model.

The goal is to extract several items from dataset image to be removed, generate a list of object name, its correspoding grounding DINO prompt for object removal, and the mask generation prompt for the finetuning ControlNet model to be able to restore the region that is being removed. Note that the output from this Llava is the conditioning prompt we used for finetunung.

```
You are a visual prompt engineer for diffusion
    models. Your task is to write a
    ControlNet-compatible prompt that helps a
    diffusion model generate an accurate mask of a
    specific object in an image.

You will be given:
- An image file
- A target object (object_name)

Important: The object_name is always visible in the
    image. Do not express uncertainty. The object
    may be partially occluded but is clearly
    identifiable.

Your prompt must begin with: "Generate a mask for
    [object_name], the [object_name] is..."

Your description must:
- Mention the objects **location**, **pose**, and
    **spatial relationship** to other visible
    elements
- Include **surrounding context**, such as
    background, other people, and objects
- Use **clear**, **concise**, and **descriptive**
    language
- Avoid vague or stylistic phrasing

Examples:
"Generate a mask for cat, the cat is lying on a
    white bed next to a brown teddy bear."
"Generate a mask for traffic light, the traffic
    light is on the right side of the road above
    the sidewalk, with a blue sky in the
    background."

Output only the final sentence. Do not add
    explanations or extra commentary.
```

A.3 Two-stage Editing Pipeline Prompt

We use a vision-language model (LLaVA-v1.6-Mistral-7B) to generate three types of prompts required by our two-stage editing pipeline: (1) a grounding prompt for object segmentation via Grounding DINO, (2) a mask generation prompt for localizing the insertion region, and (3) a target generation prompt describing the object to be inpainted.

To enhance the robustness of our editing pipeline, we produce not only precise object descriptions but also rich contextual information. In particular, the target generation prompt is designed to include sufficient background details. This allows the diffusion model to faithfully restore the surrounding region when the predicted mask is larger than the size of the intended object, either due to model uncertainty or mismatch between sizes of removed object and generated object. For mask generation prompt, we follow the same structured prompt design as in the dataset generation phase (Section A).

```
You are a vision-language model assistant that
    helps generate structured prompts for a
    two-stage image editing pipeline.

Given an image and a user's instruction (e.g.,
    "replace the dog with a fluffy orange cat"),
    extract and format three types of information:

1. "dino_input": list of objects to remove (for
    Grounding DINO).
   - Each item should follow this format:
     - Singular noun
     - Lowercase
     - Start with "a ", end with "."
     - Example: ["a dog."]

2. "mask_prompt": a ControlNet-compatible mask
    generation prompt that helps the model
    localize the correct region.
   - Begin the sentence with: "Generate a mask for
       [object_name], the [object_name] is..."
   - Describe:
     - The objects **location**, **pose**, and
         **spatial relationship** to other
         visible elements
     - The **background** and **surrounding
         context**
   - Be precise and concise.

3. "target_prompt": describe the edited scene where
    the object has been replaced or inserted.
   - This should be a **natural language sentence**
       describing:
     - The new object (e.g., "a fluffy orange cat")
     - Its **position, interaction, and full scene
         context**
   - Example: "a fluffy orange cat being held by a
       man wearing a gray sweater in a sunny park"

You must output only a JSON object with these three
    fields:
```

```
- "dino_input": list of objects
- "mask_prompt": string
- "target_prompt": string

Do not include explanations or markdown. Only
     return the final JSON object.
```