

A APPENDIX

PROJECT PAGE FOR VIEWING VIDEO EXAMPLES: <https://veditbench.github.io>

A.1 DETAILS ON EDIT PROMPT GENERATION

To generate the six task-specific edit prompts in our dataset, we followed the process outlined below. Prompts are designed using the model input shown in Figure 7. Leveraging GPT-4o, we create tasks focused on *object insertion*, *object removal*, *object swap*, *scene replacement*, *motion change*, and *style translation*. For the style translation task, we ensured diversity by randomly selecting a style from a curated set of eighty styles⁴. This approach guarantees the generation of coherent and high-quality prompts across all tasks.

Please generate five command-based modifications based on the provided base description. Use the following guidelines to create specific instructions for each modification type, taking the example of “a black and white dog is laying on a bed with a blanket on the floor and a ball under his paw”:

1. **Object Insertion:** Instruct to add a specific object to the original description.
2. **Object Removal:** Instruct to remove a specific object from the original description.
3. **Object Change:** Instruct to change the main object in the original description to another object.
4. **Scene Change:** Instruct to alter the background setting of the original description.
5. **Motion Change:** Instruct to modify the action or state of the main object in the original description.

Instructions: Use the base description to create each instruction according to the guidelines provided above. Each modification should clearly incorporate the specified change while remaining coherent with the overall context of the original description.

reply MUST in JSON format:

```
{
  "Object Insertion": "Have the dog a cowboy hat.",
  "Object Removal": "Remove the ball.",
  "Object Change": "Change the dog to a cat.",
  "Scene Change": "Make it on the beach.",
  "Motion Change": "Make the dog standing."
}
```

Base description:

Figure 7: Prompt used for edit task generation.

A.2 DETAILS ON MOTION SIMILARITY

To quantify the motion similarity between two videos, we develop Motion Similarity score (Section 4.1) that leverages object trajectory data. We first extract positional and velocity information for each object in the videos using CoTracker. The initial points are arranged in a 10x10 grid layout to ensure uniform coverage across the frame. As shown in Figure 8, a grid that is too small may result in sparse sampling, potentially missing key objects, while a grid that is too large can become overly dense, skewed toward background elements, and significantly increase computational costs. Empirical results demonstrate that a 10x10 grid achieves an optimal balance, effectively capturing the motion of both foreground and background elements while maintaining computational efficiency. The video in Figure 8 can be found in supplementary material at [videos/multi_edit.mp4](#).

As detailed in Algorithm 1, we compare these trajectories to quantify the similarity of their motion patterns. The algorithm takes two videos, \mathcal{V}^A and \mathcal{V}^B , as input, along with a parameter α that

⁴<https://openaijourney.com/stable-diffusion-styles>

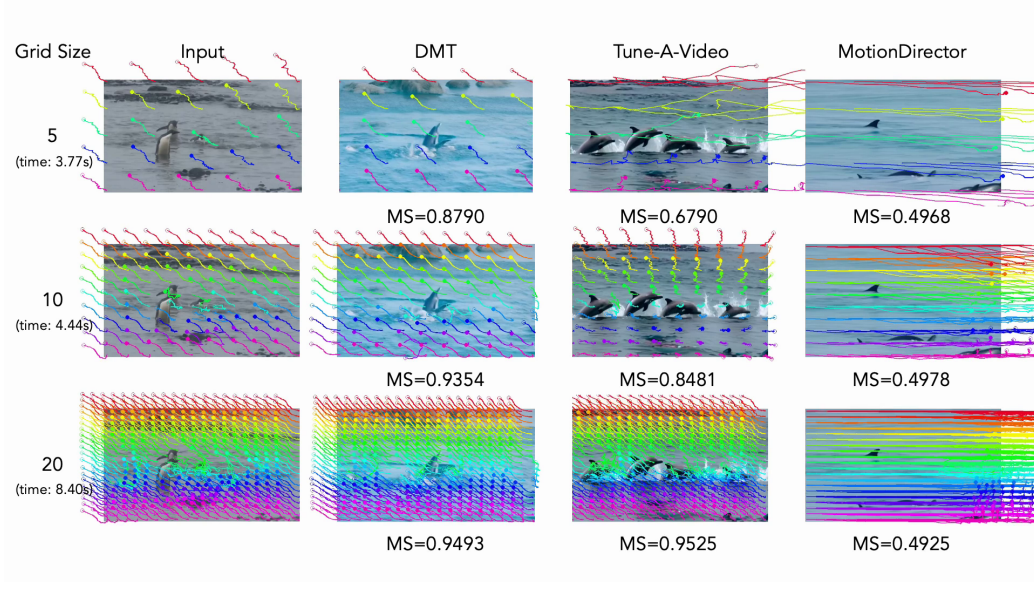


Figure 8: **Effect of grid size on Motion Similarity (MS).** Using a grid that is too small can result in sparse sampling, potentially missing important objects. Conversely, a grid that is too large becomes overly dense, dominated by background elements, and increases computational costs. Empirically, we found that a grid size of 10 strikes an optimal balance, effectively capturing the motion of both foreground and background while maintaining computational efficiency.

controls the relative importance of positional versus directional information. The output, $S_{\text{MotionSim}}$, is a similarity score ranging from 0 to 1, where higher values indicate greater similarity.

Algorithm 1 Motion Similarity Score

Require: Videos V^A and V^B , weighting parameter α

- 1: **Extract trajectories** $\mathbf{T}^A = \{(\mathbf{p}_i^A, \mathbf{v}_i^A)\}_{i=1}^N$ from V^A using CoTracker
- 2: **Extract trajectories** $\mathbf{T}^B = \{(\mathbf{p}_j^B, \mathbf{v}_j^B)\}_{j=1}^N$ from V^B using CoTracker
- 3: **Compute positional cost matrix** C_{pos} :

$$C_{\text{pos}}(i, j) = \frac{\|\mathbf{p}_i^A - \mathbf{p}_j^B\|_2}{D_{\text{max}}}$$

- 4: **Compute directional cost matrix** C_{dir} :

$$C_{\text{dir}}(i, j) = 1 - \frac{\mathbf{v}_i^A \cdot \mathbf{v}_j^B}{\|\mathbf{v}_i^A\|_2 \|\mathbf{v}_j^B\|_2 + \epsilon}$$

- 5: **Compute combined cost matrix** C :

$$C = \alpha \cdot C_{\text{pos}} + (1 - \alpha) \cdot C_{\text{dir}}$$

- 6: **Apply Hungarian algorithm** to C to find optimal assignment σ

- 7: **Compute motion similarity score:**

$$S_{\text{MotionSim}} = 1 - \frac{1}{N} \sum_{i=1}^N C(i, \sigma(i))$$

- 8: **return** $S_{\text{MotionSim}}$
-




Video	Source Prompt	Multiple Edits	Target Prompt
	The sky is cloudy and there are mountains covered in snow with trees in the foreground.	(1) Change the setting to a sunny seaside, (2) replace the mountains with a desert landscape, and (3) make the clouds move out of the screen.	The sky is cloudless. A sunny seaside with a desert landscape in the background and trees in the foreground.
	Close-up of a hand inserting a credit card into an ATM machine, with a green light illuminating the card insertion slot.	(1) Add a Batman sticker above the credit card insertion slot, (2) remove the green light illuminating card insertion slot, and (3) render the scene in an Analog Film style.	Close-up of a hand inserting a credit card into an ATM machine, with a Batman sticker just above the credit card insertion slot, rendered in an Analog Film style.
	A rocket launches into space on a clear day with a cloud of smoke.	(1) Change the rocket to a missile, (2) make it descending back to Earth, and (3) depict the scene in an Afrofuturism style.	A missile descends back to Earth on a clear day with a cloud of smoke, Afrofuturistic style.

Figure 9: Extending VEditBench to Compositional Editing task.

A.3 SAMPLE VIDEOS

We provide an anonymous webpage showcasing video examples at <https://veditbench.github.io>. The site includes videos and prompts from VEditBench, alongside edited videos produced by baseline methods and their corresponding scores across nine evaluation dimensions.

A.4 COMPOSITIONAL EDITING

VEditBench is extendable to more complex editing tasks, such as compositional editing, which involves simultaneously modifying multiple elements within a scene—such as objects, motion, and style—using a single prompt (Figure 9). We plan to evaluate the performance of existing methods on this task in the final paper. A demonstration video can be found in the supplementary materials at [videos/multi_edit.mp4](#) within the zip file.

A.5 EDIT MAGNITUDE IN OBJECT SWAP

We provide a more detailed evaluation of model performance in the Object Swap task by categorizing scenarios based on the extent of edits required. Our dataset already includes varying levels of edits for object swapping, as illustrated in Figure 10. These scenarios are divided into two distinct subcategories: **small edit**, where objects are of similar size (e.g., *replacing a tiger with a lion*), and **large edit**, which involve significant shape changes (e.g., *swapping a van with a motorcycle*). The small edit subset comprises 200 videos, while the large edit subset includes 220 videos.

A.6 MOTION MAGNITUDE

We use RAFT (Teed & Deng, 2020) to estimate the magnitude of motion in input videos, categorizing them into three distinct groups: small, medium, and large motion, based on their estimated dynamic degree. This categorization ensures an even distribution across the three motion levels. Examples of videos from each category are illustrated in Figure 11. For a clearer understanding of the motion dynamics, the videos can be viewed in the supplementary material at [videos/motion_degree.mp4](#). We will provide more analysis on the varying levels of motion in the final paper.

A.7 EXPLORING THE USE OF MASK IN EVALUATION METRICS

Incorporating object masks could enable a more fine-grained evaluation of text alignment and related dimensions. However, obtaining accurate masks for videos presents significant challenges. Manual annotation is highly labor-intensive and demands substantial human effort to ensure preci-

sion. Although automatic segmentation tools such as SAM (Kirillov et al., 2023) and SAM 2 (Ravi et al., 2024) offer potential solutions, they often introduce errors, particularly with small or partially occluded objects (Figure 12)

For instance, SAM 2 fails to segment the soccer ball accurately and even misclassifies it as a human (Figure 12, row 1). Additionally, these tools struggle with edited videos (rows 2 and 3), especially when temporal consistency is not maintained. Such inaccuracies can compromise the robustness of the evaluation metric, potentially introducing bias or noise into the results and reducing overall reliability.

We see this as an important avenue for future research and plan to explore efficient and effective approaches for incorporating high-quality object masks in future iterations of VEditBench.

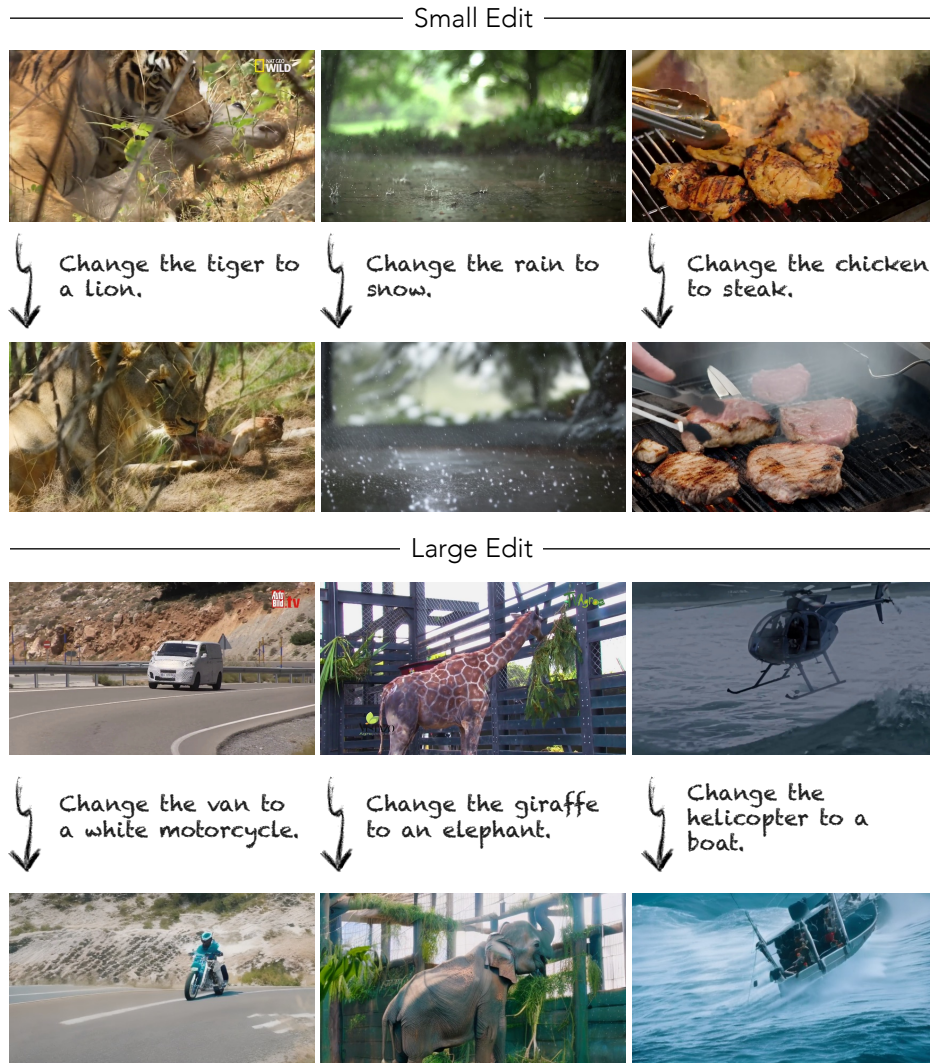


Figure 10: Examples of small and large edit in Object Swap task.

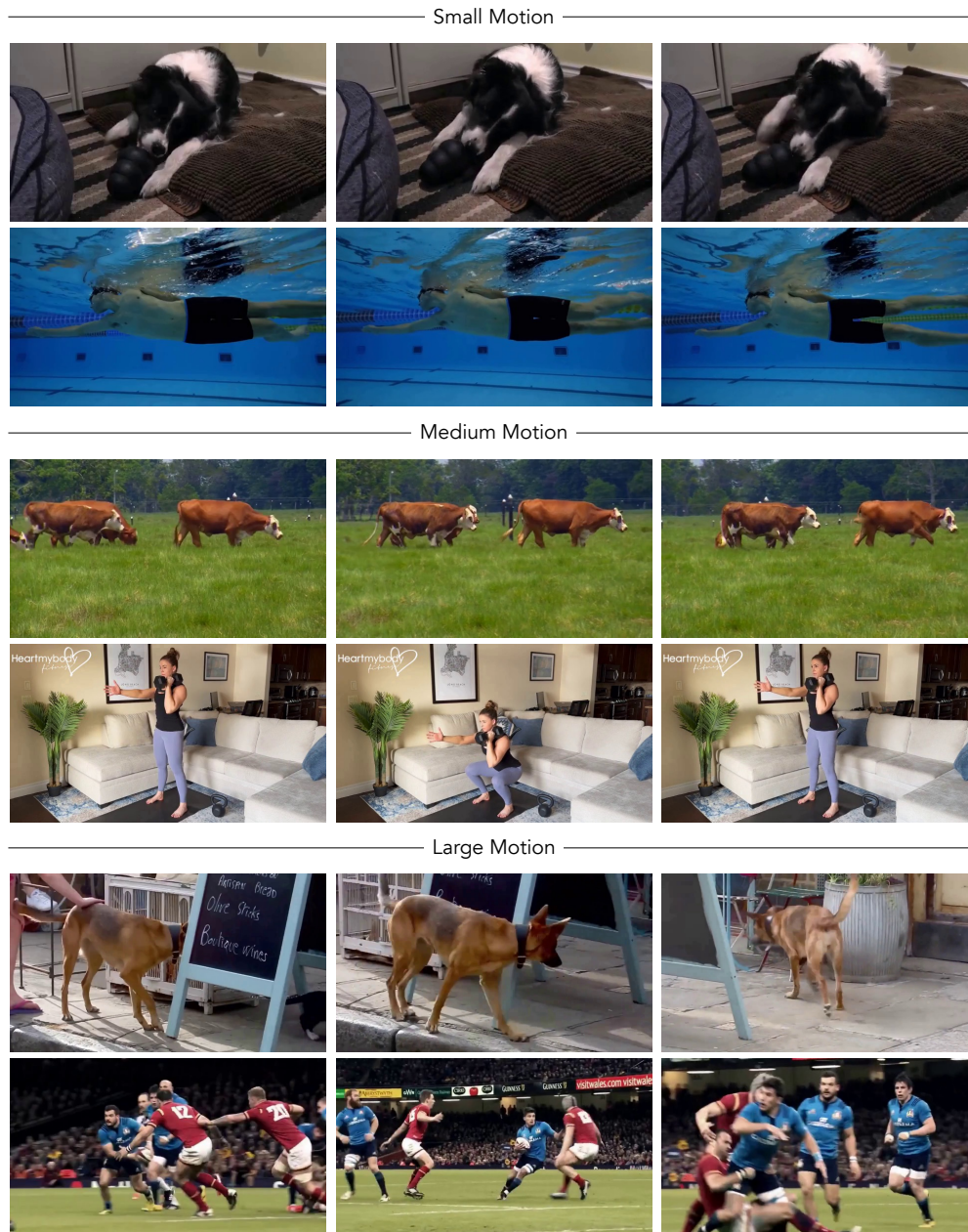


Figure 11: Examples of small, medium and large motion in VEditBench.



Figure 12: **Exploring automatic segmentation tools for evaluation metrics.** SAM 2 (Ravi et al., 2024) struggles to segment the soccer ball accurately and consistently over time (row 1). It also encounters difficulties with edited videos (rows 2 and 3), especially when the videos are blurry or when temporal consistency is disrupted. Please zoom in for best view.

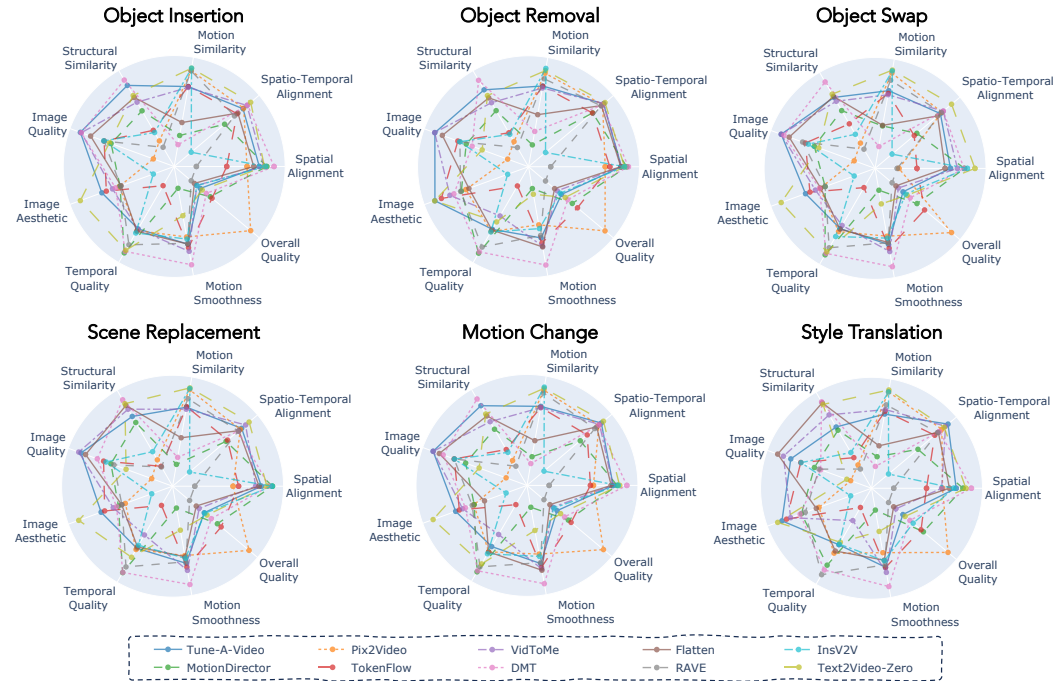


Figure 13: **Results per editing task on VEditBench-Short.**