
Pexels Pairs: Multiview Masked Entities from Video Data

Ryan Murdock* Jizhizi Li Jacob Schneider-Martin Jiayang Ao Danny Wicks

Sami Ede

Rebecca Li[†]

Abstract

Image editing models have shown impressive improvements in image understanding and manipulation over the last several years. Although many of these models are released or open-sourced, the data necessary to train them is less readily available. Here we present *Pexels Pairs*, a permissively-licensed dataset of 24K samples containing masked entities in near-counterfactual image pairs. We carefully curate these pairs from videos sourced from Pexels in a multi-stage pipeline and create high-resolution data featuring entities acting or being acted on with instruction-based captions. In addition to our paired data, we release our 330K videos with their exhaustive masklets.

1 Introduction

Approaches to creating or capturing image editing data are varied, with early text-conditioned datasets typically focusing on curated synthetic data, low-level computer vision tasks, and images of celebrities [2, 7, 14, 17]. Meanwhile, many approaches for object manipulation [1, 16] and newer text-conditioned methods integrate additional curated video data [13, 15].

Counterfactual images [12] show the same scene with and without a specific object. This class of image pairs is of particular interest for the modeling of object manipulation tasks like object insertion, manipulation, and erasing. However, their manual collection can be costly and time-consuming. Previous works like OmniEraser [11] and RORD [6] collect near-counterfactual images using video data with fixed cameras where entities may move in and out of frame. We take a similar approach, curating and cropping high-resolution videos where an object or subject moves outside an area of a scene. See Figure 1 for examples from our data, available at *Pexels-Pairs-Masklets-330K*, *Pexels-Pairs-Near-Counterfactuals*, and *Pexels-Pairs-Text-Near-Counterfactuals*.

In addition to our near-counterfactuals curation, *Pexels Pairs* is differentiated from previous work by its clear and permissive licensing (available at <https://www.pexels.com/terms-of-service/>) and carefully curated, high-quality source videos. As a result, it is well suited for application to object manipulation tasks (e.g., erasing, inserting, and dragging). With our additional instruction-oriented captioning, it can also serve as focused data for instruction-based image editing.

2 Related Work

2.1 Object Editing Models

Models like MagicFixup and ObjectMover [1, 16] target intuitive, natively-visual interfaces for manipulating existing elements within scenes. MagicFixup learns from paired video frames, whereas

*Send correspondence to rynmurdock@gmail.com

[†]All authors are with Canva Research



Figure 1: Thumbnails showing near-counterfactuals from Pexels Pairs. Our data additionally includes the full scenes, entity masks, VLM edit captions and entity names, and patch-wise similarity information.

Table 1: Comparison of video-derived datasets for object editing. Pexels Pairs is the only one combining automatic mining, non-static viewpoints, exhaustive entity masks, and a permissive public release.

Dataset	Collection	Camera	Masks	Primary tasks	Public
RORD [6]	Manual	Fixed	Single object	Removal	Yes
Video4Removal [11]	Auto	Fixed	Single object	Removal (+effects)	No
X2I2 [13]	Auto	—	Not released	Instruction edits	Pairs only
ImgEdit [15]	Auto	—	Not released	Instruction edits	Pairs only
Pexels Pairs (ours)	Auto	Dynamic	All entities	Remove / insert / drag / instruction edits	Yes (perm.)

ObjectMover relies primarily on a synthetic game-engine pipeline, using real video only as an auxiliary signal. In either case the training data is rarely released as a reusable resource, and where a public substitute exists, it can be limiting. For instance, the open-source MagicFixup model is trained on Moments in Time [4], whose standard distributed frames are only 340×256 pixels, well below what is typical for training current models. Pexels Pairs helps fill this gap for newer object-manipulation and text-conditioned editing models.

2.2 Datasets for Object Editing using Video Data

Counterfactual data derived from real video has driven clear gains on object editing and removal, yet much of it is hard to build upon. Some datasets were never distributed as a standalone resource (e.g., ObjectDrop) [12]. Among released datasets, RORD [6] provides 516,705 HD (1920×1080) image pairs across 3,447 scenes, but each pair is produced by physically staging and then removing an object in front of a fixed camera. As a result, it is annotated with a *single* target object and confined to static viewpoints and labor-intensive manual collection. Closest in spirit to our approach, Video4Removal [11] instead reduces labor by automatically mining object and background frame pairs from video with off-the-shelf vision models, comprising over 100,000 pairs. It is, however, purpose-built for object (and object-effect) removal, draws pairs under largely fixed viewpoints, annotates a *single* target object whose mask deliberately excludes shadow and reflection regions, and, like ObjectDrop, is not publicly released. Pexels Pairs shares this automatic, low-labor philosophy but differs in three ways. First, we exhaustively mask *every* entity in both frames rather than one removal target, so the same pairs also support insertion, dragging, and text-conditioned editing. Second, our crop-and-erode procedure (Sec. 3.2) recovers near-counterfactuals from clips with moving objects and non-stationary cameras rather than only fixed viewpoints. Third, all data derives from permissively licensed Pexels stock and is released in full.



Figure 2: Columns here show two frames from the same video. Each video has masklets tracking each discovered entity.

2.3 Datasets for Instruction-Guided Edits using Video Data

Datasets like OmniGen2’s X2I2 [13] and ImgEdit [15] use video as one source among several for constructing text-conditioned edit examples, and their pipelines likewise rely on entity detection and segmentation. They differ from Pexels Pairs in two respects that make them complementary rather than overlapping. First, much of their supervision is produced by inpainting or generative repainting rather than observed directly, a construction that OmniGen2 itself notes can degrade image quality and cause the edit instruction to mismatch the resulting pair. Our counterfactual subset, by contrast, is drawn solely from real, unedited footage. Second, both release only the artifacts needed for instruction tuning (input image, output image, and instruction) rather than the underlying per-entity masks. Pexels Pairs instead releases exhaustive entity masks over natural frame pairs, offering a reusable substrate that can be processed into removal, insertion, dragging, or text-conditioned edits, rather than just the set of pre-baked instruction triplets.

3 The Pexels Pairs Dataset

3.1 Data Statistics

Within the data, the average cropped natural-counterfactual is 1477×1348 pixels and covers 28% of the full scene. 57% of pairs contain two different views of the entity; the other 43% show only one view, meaning the entity entirely left the shot between the two frames. Pairs in which we have both views of the entity may be especially well-suited for object insertion and object movement tasks. We tend to take only one or two entities from each video (mean=1.57), meaning that we avoid mining any same small set of scenes many times over.

3.2 Video Data Processing Pipeline

We source our data from Canva’s Pexels stock library, which provides us high resolution video content with clear licensing and extensive manual curation, making it ideal for model training.

Before processing our data to identify near-counterfactuals, we segment the full extent of each frame using SAM [3], creating masks for each object. After this, we use those masks to prompt SAM2 for masklets [5]. We adapt the AutoSeg-SAM2 library available at <https://github.com/zrporz/AutoSeg-SAM2> [19] for this portion of the pipeline and subsample videos at one frame per second. See Figure 2 for examples of this data.

After obtaining masks, our processing is ordered from least to most computationally expensive and from the levels of videos, entities, and then frames. This allows us to disqualify data early on, lowering cost and latency.

Table 2: Our data filtering criteria ordered sequentially. By excluding many videos and entities early on, processing our 338,705 videos becomes feasible within days on only a handful of H200 GPUs.

Videos
> 33 or < 1 discovered masks
Includes alpha transparency
Entities
SAM predicted IoU < 0.7
Bounding box non-maximum suppression < 0.7
The entity’s bounding box does not move across the entire video
The entity’s mask overlaps itself in every frame pair
The entity is present in < 4 frames
The entity moves over > 0.9 of the video diagonal in one frame
$\sqrt{\text{total pixels in mask}} < 256$
Frames
Entity absence from both frames
LPIPS similarity < 0.75
Mean SSIM over crop < 0.6
There is no crop that excludes the object in one frame but not the other and passes our other criteria
Either side of the crop < 512 pixels long

3.3 Mining Counterfactuals

Following our initial filtering and segmentation in Section 3.2, we utilize the intersection over union (IoU) of masks alongside perceptual similarity metrics to discover potential “natural counterfactual images.” We choose SSIM [10] and LPIPS [18] to target structural information while remaining invariant to changes in shadow, reflection, and other side effects of the entity on its surroundings.

We observe that in our video data near-counterfactual images exist when one entity moves across a scene and the rest of the scene and its viewing angle remain largely unchanged. We also notice that while the entire frame of a video may not fit these criteria, a large portion of the frame may. When they contain a sufficiently large area, these cropped areas of frames can serve as counterfactual images themselves.

We visualize our data processing pipeline in Figure 3. To collect scenes that meet or closely approximate our criteria of an entity moved from its original location with an otherwise static scene, we select pairs in which the IoU of an entity’s mask is zero while surrounding pixels retain high perceptual similarity between frames. LPIPS and SSIM are calculated patchwise to facilitate eroding the scenes until we obtain an area that meets our criteria. For our near-counterfactual, we crop our scenes such that they contain only one view of the entity; in cases where two views exist, we take the larger possible crop containing one view and not the other.

In addition to pixel-level similarity, we filter and erode masks by measuring the IoU of masks within the candidate crop, excluding the mask of the candidate entity. Mask metrics are notably insufficient alone: for example, the mask of an entity could remain entirely unchanged while its contents are rotated. However, we find their combination with perceptual similarity metrics provides another useful guardrail for capturing near-counterfactuals with the additional benefit that they are ideally invariant to side effects like shadow and reflection by design. Table 1 lays out our exclusion criteria in detail.

We pack each pair into tars in WebDataset format. See Table 3 for the structure that each sample dictionary follows.

3.4 VLM Captioning and Filtering

In addition to the filtering in Section 3.3, we run VLM captioning and collect yes/no log-probability scores for further refinement of our mined counterfactuals. Using Qwen3.5-VL-27B [8] and InternVL3_5-30B-A3B [9], we collect action-oriented captions, specifying the change between

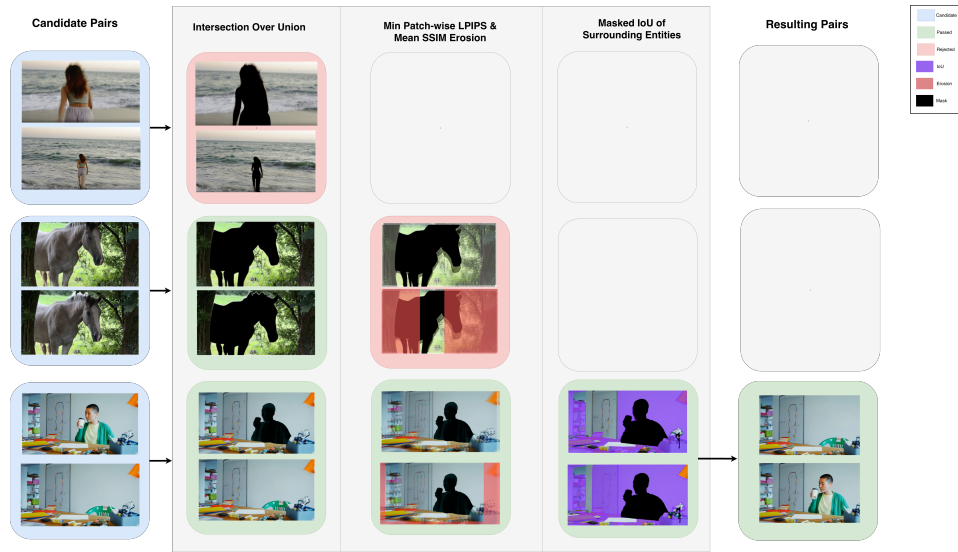


Figure 3: After initial video-level and entity-level filtering (described in Section 3.2), we compare frames each-to-each, collecting the most temporally distant pair for each entity that passes our criteria after eroding and cropping. These criteria include having no intersecting area in their masks, which excludes the sample showing a woman at the beach, and surpassing minimum similarity thresholds outside of their masks while retaining cropped side lengths greater than 512 pixels, which excludes the sample showing a horse. This process allows us to obtain near-counterfactuals within videos that have multiple moving objects and cameras that are not still across the entire video.

Table 3: Our dataset schema. The consistent crop covers the same area in both frames but contains the entity in only one of them. Bounding boxes and cutouts are absent for a frame without the entity present, and cutout similarity exists only when the entity appears in both frames.

Key	Description
<code>__key__</code>	Unique sample id
<code>video_id.txt</code>	Source video id
<code>scenes_width_height.json</code>	Frame size in pixels
<code>first_scene.jpg</code>	Full 1 st frame
<code>second_scene.jpg</code>	Full 2 nd frame
<code>consistent_crop_bounding_box.json</code>	Consistent crop $[x, y, w, h]$
<code>first_scene_consistent_crop.jpg</code>	Consistent crop of 1 st frame
<code>second_scene_consistent_crop.jpg</code>	Consistent crop of 2 nd frame
<code>first_cutout_view.png</code>	Entity cutout with alpha mask, 1 st frame
<code>second_cutout_view.png</code>	Entity cutout with alpha mask, 2 nd frame
<code>first_bounding_box.json</code>	Entity bounding box, 1 st frame
<code>second_bounding_box.json</code>	Entity bounding box, 2 nd frame
<code>lpips_patchwise_similarity_cutouts.json</code>	Mean LPIPS between cutouts
<code>similarity_of_frames.json</code>	Patchwise LPIPS between frames
<code>ssim_of_frames.json</code>	Patchwise SSIM between frames
<code>vlm_prompt_logprob.<model>.json</code>	VLM likelihood of “Yes” to a whole visible entity
<code>vlm_entity_name.<model>.txt</code>	VLM short name for the changed entity
<code>vlm_change_caption.<model>.txt</code>	VLM one-sentence caption of the change



Figure 4: Examples of image edit instructions and entity names generated for our image pairs. Zoom in for best clarity. We prompt for generic names in order to produce the most useful categories, facilitating finetuning on edits for specific use cases like removing subjects from photos.

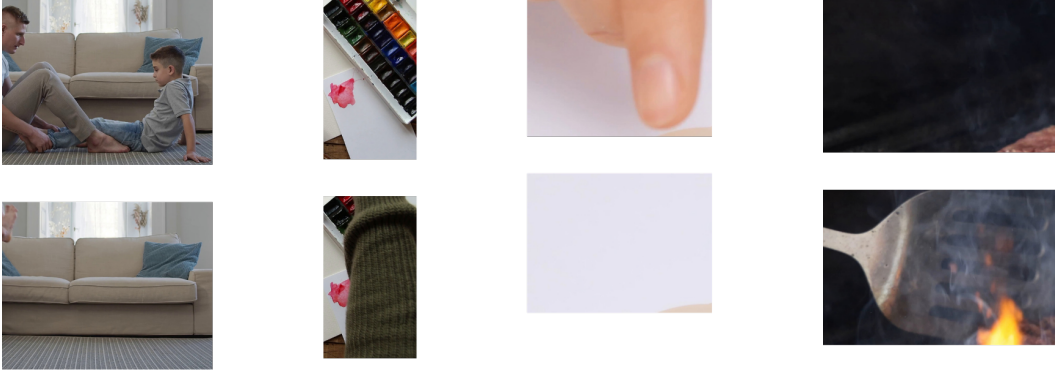


Figure 5: Examples of typical image pairs following mining without applying VLM filtering and annotation and using moderate filtering thresholds.

the first and second full scenes as an instruction, allowing for our data to be used in instruction-based image editing. The design of our data curation here is well-suited for enabling action-oriented edits to a single entity. In addition, we collect short entity names for the changed entity in our counterfactuals. We explicitly specify that simple graphics should be tagged as such, allowing us to filter non-photorealistic samples when useful. Finally, we include the predicted likelihood that the counterfactual pair features a whole person or object, allowing us to curate a “cream of the crop” of near-counterfactuals with whole entities in photorealistic style. Figure 4 shows such examples, in this case specifically tagged as showing people in order to illustrate entity names’ usefulness for targeted finetuning for specific use cases.

3.5 Limitations

While we succeed in collecting a large quantity of natural near-counterfactuals, our pairs are not perfect true counterfactuals and are not collected by hand. The first pair of our samples in Figure 5 shows an example of two entities, specifically the two subjects, being grouped into a single masklet with a small portion of the masklet retained on the edge. The third pair exemplifies a part being shown instead of a whole within our crop, in this case two fingers instead of a hand or whole person. Entity grouping should not interfere with many of the object manipulation tasks of interest, like erasing, inserting, etc., but the distinction is notable. Additionally, we have set our thresholds at reasonable levels to allow for high-quality pairs per extensive visual inspection – and more importantly provide the necessary similarity features to facilitate staged or weighted training with our data.

4 Conclusion

We release three datasets: the first contains 330K videos with exhaustive masklets over entities; the second features 24K curated near-counterfactuals; and the third enriches our near-counterfactuals with instruction-focused VLM captions and the name and part/whole prediction for the pairs’ entity of interest. Counterfactual image pairs are rare, and due to the costs of manual collection, difficult to obtain. High quality real-data image edit pairs are similarly scarce. Given this, Pexels Pairs offers a valuable resource for advancing image editing with text instructions and intuitive visual interfaces. While the usefulness of open models for the research community is difficult to overstate, we believe

contributions of open data are another vital ingredient. In publishing this data, we hope to provide an additional high-quality resource for researchers to advance novel UXs and approaches to image editing.

References

- [1] Hadi Alzayer, Zhihao Xia, Xuaner (Cecilia) Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. Magic fixup: Streamlining photo editing by watching dynamic videos. *ACM Transactions on Graphics*, 44(5):1–25, 2025.
- [2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023.
- [3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.
- [4] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfrud, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding, 2019.
- [5] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024.
- [6] Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. Rord: A real-world object removal dataset. In *British Machine Vision Conference*, 2022.
- [7] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks, 2023.
- [8] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026.
- [9] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingdong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhao Wang, and Gen Luo. Internv13.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025.
- [10] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [11] Runpu Wei, Zijin Yin, Shuo Zhang, Lanxiang Zhou, Xueyi Wang, Chao Ban, Tianwei Cao, Hao Sun, Zhongjiang He, Kongming Liang, and Zhanyu Ma. Omnieraser: Remove objects and their effects in images with paired video-frame data, 2025.
- [12] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion, 2024.
- [13] Chenyuan Wu, Pengfei Zheng, Ruirui Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Towards instruction-aligned multimodal generation, 2026.
- [14] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruirui Yan, Chaofan Li, Shutong Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation, 2024.

- [15] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark, 2025.
- [16] Xin Yu, Tianyu Wang, Soo Ye Kim, Paul Guerrero, Xi Chen, Qing Liu, Zhe Lin, and Xiaojuan Qi. Objectmover: Generative object movement with video prior, 2025.
- [17] Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing, 2024.
- [18] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.
- [19] Zrporz. Autoseg-sam2, 2024. Automated image segmentation tool based on Segment Anything Model (SAM).