Refaçade: Editing Object with Given Reference Texture

Supplementary Material

A. Related Works

A.1. Image Inpainting

Recent progress in image editing is largely driven by deep neural models [1, 5, 6, 13, 14, 25, 36]. Among these methods, SD-Inpainting [6] and ControlNet Inpainting [32] extend Stable Diffusion by fine-tuning on datasets of randomly masked images paired with text prompts. Although these adaptations generate visually plausible results, they often drift from the input text and struggle to place objects accurately according to the described semantics. To mitigate this issue, SmartBrush [47] and Imagen Editor [41] incorporate paired object-description data, yet they implicitly assume that the masked region always contains an object, which restricts their capacity for context-aware completion. PowerPaint [57] instead learns task-specific prompts that adapt to the mask, which strengthens the relationship between textual input and contextual surroundings and leads to state-of-the-art performance in both context-aware inpainting and text-guided editing. BrushNet [25] builds on ControlNet to extract conditioning information and inject it into a frozen diffusion U-Net, whereas Turbo-Fill [46] emphasizes efficiency by combining a few-step text-to-image diffusion process with an inpainting adapter to achieve fast and high-fidelity results. Flux-Fill [5], trained on the Flux base model, likewise produces visually compelling inpainting outcomes. In addition, several methods focus on editing flexibility and object removal. Attentive Eraser [39] proposes a tuning-free strategy that enables pre-trained diffusion models to perform stable and effective object removal. DesignEdit [23] introduces a simple yet powerful approach for spatially flexible editing that first inpaints the background and then applies a two-stage multi-layer latent diffusion framework to modify each element independently. RORem [29] adopts a semi-supervised human-in-the-loop pipeline to curate high-quality paired training data, and ObjectClear [54] integrates an object-effect attention mechanism that guides the model toward target foreground regions through attention masks.

A.2. Video Inpainting

Analogous to image inpainting, existing video inpainting approaches can be broadly grouped into video object removal and text-guided video inpainting. Within video object removal, a line of work focuses on explicit removal of target objects. FFF-VDI [28] propagates future-frame latents to initialize masked regions and then fine-tunes an image-to-video diffusion model to complete the corrupted area. FloED [18] injects both optical-flow and text embeddings to guide removal. DiffuEraser [30] couples flow-guided inpainting with DDIM inversion to attain higher fidelity. Senorita-Remover [59] relies on instruction-driven prompts, using positive prompts to guide removal and negative prompts to suppress unintended content. Minimax-Remover [58] employs a minimax optimization objective that improves removal quality and prevents undesired object regeneration. For text-guided video inpainting, recent work addresses masked-region generation and editing under text prompts. VideoComposer [42] is an early diffusion model for text-guided video inpainting that offers multi-conditional control within a unified framework. AVID [52] scales to sequences of arbitrary length from natural-language prompts. COCOCO [60] improves consistency and controllability using damped global attention and stronger text cross-attention. VIVID [20] provides a 10M-scale image video corpus for localized editing, which enables more capable text-guided inpainters. MTV-Inpaint [49] unifies scene completion and novel object insertion within a single framework. VideoPainter [4] adopts a DiT-based architecture with a context encoder that injects background cues into a pretrained video DiT to achieve plug-and-play consistent inpainting. More recently, VACE [24] introduces a video editing framework that consumes multiple control signals to generate edited videos.

Remark. *Despite notable successes, most inpainting systems remain unable to use a reference image to direct the outcome inside the missing areas.*

B. Dataset Setup Details

The dataset used to train our Stage 1 model consists of two components, a filtered subset of WebVid-10M and our synthetic dataset. The former provides large-scale and inexpensive video resources, while the latter focuses on videos containing objects with rare and long-tailed textures. To construct the synthetic dataset, we use Qwen3-14B [48] to generate 2.2M prompts, which are then used for text-to-image synthesis with Stable Diffusion 3.5 Large [38] and text-to-video synthesis with Self-Forcing [21]. The Stage 2 model is trained on videos from Pexels[35], which leads to improved aesthetic quality.

Dataset for ablation study. For the patch size ablation in Table 4 of the main text, we consider a setting where the reference image shares the same texture as the target object but differs in shape and size. We denote this setting as a patch size of 100%. To obtain such paired images, we em-

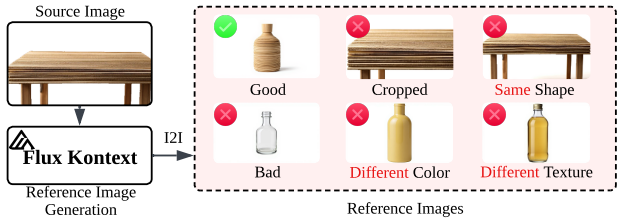


Figure 1. Visual results of reference images generated by Flux-Kontext for our ablation study. Only those reference images that have a different shape from the source but share the same material and dominant color are retained.

ploy Flux-Kontext [27] and leverage its image-to-image capability to generate reference images by reshaping the target objects. We use the following prompt:

Prompt for Reference Image Generation

Reshape the {source object} into a {target object} with same color and texture.

However, relying solely on Flux-Kontext to generate reference images is not sufficient, as we observed that many of the resulting references are suboptimal as shown in Figure 1. Therefore, we further filter the generated images and videos using GPT-5 [33], which yields a final dataset of only 10K video pairs and 8K image pairs for no-Jigsaw training.

C. Implementation Details

C.1. Training Details of Refaçade and Texture Remover

Training Details of Refaçade During training, we randomly resize and downsample frames. In addition, we randomly drop the conditioning information with probability 0.1 by replacing the reference image with an all-white image and its mask with all-black pixel value, so that classifier-free guidance can be applied at inference time. The batch size is 96 in Stage 1 and 32 in Stage 2, with constant learning rate of 1e-5.

Training Details of Texture Remover The training procedure for the Texture Remover is similar to that of Refaçade. We use DMD2 to distill the Texture Remover from 50 sampling steps to 3 steps. Table 1 summarizes the key hyperparameters of Refaçade and the Texture Remover.

C.2. Inference Details of Texture Remover

At inference time, we provide the object mask together with the input video and remove the background so that the input matches the training format. The Texture Remover then produces a sequence of texture free mesh videos that are temporally aligned with the source video. The same pipeline also supports single frame inference, which allows us to obtain texture free conditions from still images. During inference, we disable classifier free guidance and use three sampling steps. For video inputs, inference is performed at a resolution of 480×832 with 81 frames, and for image inputs it is performed at the original resolution.

Table 1. Hyperparameter of **Refaçade** and Texture Remover.

Config	Model			
	Refaçade		**Texture Remover**	
	Stage 1	Stage 2	Stage 1	Distill
Batch Size / GPU	1		1	
Accumulation Step	4		1	
Gradient Checkpointing	True		True	
Optimizer	AdamW		AdamW	
Learning Rate	1×10^{-5}		1×10^{-5}	5×10^{-6}
LR Schedule	Constant		Constant	
Time Sampling	Uniform		Uniform	
Num GPUs	96	32	32	8
Training Steps	18000	2800	18000	300
Num Main Layers	24		24	
Token Dimension	1536		1536	
Parameters	2.0258B		1.7143B	
Control Layer Indices	0,5,10,15,20,24,28		0,5,10,15,20,25	
Pre-trained Model	Wan2.1-VACE-1.3B		Wan2.1-VACE-1.3B	
Sample Steps	20		50	3
Sampler	Flow UniPC [55]		Flow Euler	
Input Resolution(s)	Multi-resolution		Multi-resolution	
Frame Count(s)	Multiple frame lengths		Multiple frame lengths	

C.3. Inference Details of Refaçade

Inference is conducted on a single A800.

Image editing. Editing a single image with resolution 480×832 peaks at about 12 GB of GPU memory and takes approximately 6.5 s. The Texture Remover accounts for about 0.35 s, and the remaining overhead arises from VAE encoding and decoding as well as the diffusion process. The CFG scale is set to 1.5.

Video editing. Editing an 81 frame video at 480×832 peaks at about 20 GB of GPU memory and takes approximately 150 s. The Texture Remover contributes 5.5 s of this runtime. The CFG scale is set to 1.5.

C.4. Inference Details of Baseline Methods

Most image and video editors rely on text prompts rather than reference images. To accommodate such editors, we use Qwen-VL-2.5 32B to generate captions for the reference image and then convert these captions into text prompts. For editors that accept a reference image as input, we directly feed the reference image into the model for inference. The templates for generating prompts are detailed below.

Template for Instructive Prompt Generation

You are given an image and a target object name: {object_name}.
1) Identify the dominant color tone(s) and the surface material/texture of the main object in the image (choose the largest/central salient object).
2) Write ONE imperative instruction to restyle {object_name} with that exact color and material.
Rules: - 20–40 words, one sentence, NO ENTER.
- Mention color shade (e.g., dark brown, icy blue) and material (e.g., chocolate texture, brushed metal, glossy ceramic).
- No extra commentary.
For example, "Turn the dog into dark brown, covered with chocolate texture". If multiple objects, "Turn the bike and man into dark brown, covered with chocolate texture".
Please ONLY return the instructive prompt sentence.

Template for Descriptive Prompt Generation

You are given an image and a target object name: {object_name}.
1) Identify the dominant color tone(s) and the surface material/texture of the main object in the image (choose the largest/central salient object).
2) Write ONE descriptive prompt to describe {object_name} with that exact color and material. Rules:
- 20–40 words, one sentence, NO ENTER.
- Mention color shade (e.g., dark brown, icy blue) and material (e.g., chocolate texture, brushed metal, glossy ceramic).
- No extra commentary.
For example, "A dog in dark brown, covered with chocolate texture". If multiple objects, "Bike and man in dark brown, covered with chocolate texture."
Please ONLY return the descriptive prompt sentence.

C.4.1. Image Baseline

Implementation Details of BrushNet. We adopt BrushNet with its released pretrained checkpoint together with the Stable Diffusion XL base model to generate images at resolution 1024×1024. The pipeline takes the original image, its corresponding mask and a descriptive prompt as input. We perform 50 denoising steps with a CFG scale equal to 5.0 and set `brushnet_conditioning_scale` to 1.0. The generated images are then resized to the original resolution for comparison with other methods.

Implementation Details of Controlnet-Inpainting. We use the pretrained control block checkpoint together with the Stable Diffusion 1.5 base model for inference. Input images are first resized to resolution 512×512. We then provide the source image and its corresponding mask together with a descriptive prompt to the inference pipeline. We employ the default settings with 20 denoising steps and a CFG scale is set to 7.5.

Implementation Details of Flux-Fill. We use the FLUX.1-Fill-dev model and perform inference at the original image resolution. The pipeline takes the source image, its corresponding mask and a descriptive prompt as input. We set the CFG scale to 30.0 and use 50 steps for inference.

Implementation Details of Flux-Kontext-Text. We use the FLUX.1-Kontext-dev model conditioned on the instructive prompt. Inference is performed at the original image resolution with 28 denoising steps and CFG is set to 3.0.

Implementation Details of Flux-Kontext-Image. We use the FLUX.1-Kontext-dev model conditioned on both the reference image and the instructive prompt. Inference is performed at the original resolution of the source image and mask using 42 denoising steps with a CFG scale equal to 2.5 and `strength` set to 1.0.

Implementation Details of HiDream-E1. We use the HiDream-E1-1 model. The source image and mask are first resized to resolution 768×768. We set the CFG to 3, `image_guidance_scale` to 1.0 and `refine_strength` to 0.3. Both the instructive prompt and the descriptive prompt are used as textual conditions as shown below

Prompt for HiDream-E1

Editing Instruction {instructive_prompt} Target Image Description {descriptive_prompt}

Implementation Details of HQ-Edit. We use the released pretrained checkpoint of HQ-Edit. Input images are resized to resolution 512×512 before inference. We set the CFG to 7.0, perform 30 denoising steps and set `image_guidance_scale` to 1.5 while conditioning on the instructive prompt. Finally, the generated images are resized back to the original resolution for comparison.

Implementation Details of InsP2P. We use the released pretrained InsP2P checkpoint for inference. Input images are resized to resolution 512×512 in advance. We set the text guidance scale `text_cfg_scale` to 7.5 and the image guidance scale `image_cfg_scale` to 1.5 while conditioning the model on the descriptive prompt. We perform 100 denoising steps.

Implementation Details of NanoBanana. We call the official NanoBanana API to generate edited images. Due to the aspect ratio constraint in this API, we first resize input images to resolution 1024×1024. The output image is then resized back to the original resolution. The model is conditioned on the source image, the reference image, and the following textual prompt:

Prompt for NanoBanana

Keep the background unchanged. Replace the texture of the {object} in the first image using the material from the second image (the reference). Output only the edited image.

Implementation Details of Qwen-Image-Edit. For Qwen-Image-Edit, we perform inference at the original resolution of each input image. We run 50 denoising steps with `true_cfg_scale` set to 4.0, conditioning the model on the instruction prompt.

Implementation Details of Stable Diffusion3-Inpainting. For Stable Diffusion3-Inpainting, we use the Stable Diffusion3-medium base model. Inference is carried out at the original resolution of the source image and its mask, using 50 denoising steps with the CFG scale set to 7.0, conditioned on the descriptive prompt.

Implementation Details of UltraEdit. For UltraEdit, we use the pretrained UltraEdit checkpoint for inference. The source images and masks are uniformly resized to a resolution of 512×512 before sampling. We run 50 denoising steps with the CFG scale set to 7.5 and `image_guidance_scale` set to 1.5, conditioning the model on the descriptive prompt.

Implementation Details of Pair Diffusion. We use the finetuned SDv1.5 for inference. Source images, source masks, reference images and reference masks are all resized to a resolution of 512×512. We run 30 ddim steps for inference, conditioned on the descriptive prompts.

Implementation Details of Cross-Image Attention. We use the pretrained SDv1.5 as base model. We use source image as *struct image*, and reference image as *appearance image*. Images are all resized to 512×512 in advance. We set `contrast_strength` to 1.67 and `swap_guidance_scale` to 3.5. Inference is performed using 68 denoising steps.

Implementation Details of ZeST. We use the pretrained SD-XL and IP-Adapter [50] for inference. Images are all resized to 512×512 in advance. Structural information is provide by Depth. We set `control_scale` to 0.9 and `brightness` to 1.0. Inference is performed using 30 denoising steps.

C.4.2. Video Baseline

Implementation Details of AnyV2V. We adopt a two-stage pipeline built upon I2VGen-XL [51]. In the first stage, we apply DDIM inversion with 500 steps to obtain noisy latents from the input video. In the second stage, we use Flux-Fill to edit the first frame, conditioning the generation on both the inverted latents from the first stage and the descriptive prompt. We set `pnp_f_t` = 1, `pnp_spatial_attn_t` = 1, and `pnp_temp_attn_t` = 1. The input videos are resized to a spatial resolution of 512×512, and the number of frames is truncated to 36. We use a CFG scale of 9.0 and perform 50 denoising steps. Finally, the generated videos are resized back to the original resolution for comparison.

Implementation Details of COCOCO. We use the pretrained COCOCO checkpoint together with the Stable Diffusion Inpainting model for inference. The input videos and their masks are resized to a spatial resolution of 512×512 and truncated to 33 frames. We set CFG to 10.0 and perform 50 denoising steps, conditioning the model on the descriptive prompt and using a negative prompt of "worst quality, low quality". Finally, the generated videos are resized back to the original resolution for comparison.

Implementation Details of Ditto. We use a pretrained LoRA with Wan2.1-VACE-14B. Inference is performed at a resolution of 480×832 with 33 frames, conditioned on the instructive prompt, while keeping all other settings at their default configuration. The generated videos are resized back to the original resolution.

Implementation Details of Flatten. We use Stable Diffusion 2.1 as the base model. The input videos are resized to a spatial resolution of 512×512 and truncated to 33 frames. We perform 50 denoising steps with CFG set to 15.0 and set `inject_step` to 40, conditioning the model on the descriptive prompt. All other settings follow the default configuration.

Implementation Details of ICVE. We use the pretrained ICVE checkpoint together with the HunyuanVideo base model. We follow the default parameter configuration, resizing input videos to a resolution of 240×384 and truncating them to 33 frames. Inference is performed with 50 denoising steps and CFG set to 6.0, with `embedded_cfg_scale` set to 1.0, conditioning the model on the instructive prompt.

Implementation Details of InsV2V. We use the pretrained InsV2V checkpoint for evaluation. Input videos are resized to a resolution of 384×384 and truncated to 33 frames. We set `text_cfg` to 7.5 and `img_cfg` to 1.2, while keeping all other parameters at their default settings. The generated videos are resized back to the original resolution.

Implementation Details of InsVIE. We use the pretrained InsVIE checkpoint together with the CogVideoX-2B base model. Input videos are resized to a resolution of 480×720 and truncated to 49 frames. The model is conditioned on the instructive prompt, with a negative prompt of "bad quality", while all other parameters follow the default configuration.

Implementation Details of LucyEdit. We use the Lucy-Edit-1.1-Dev model for evaluation. Input videos are resized to a resolution of 480×832 and truncated to 33 frames. We set CFG to 5.0 and condition the model on the instructive prompt, using empty prompt as the negative prompt. All other settings follow the default configuration. Finally, the generated videos are resized back to the original resolution.

Implementation Details of Señorita. We use the pre-

trained Señorita checkpoint with the CogVideoX-5b-I2V base model for evaluation. Input videos are resized to 448×768 and truncated to 33 frames. The first frame is edited by Flux-Fill and then used as the starting frame for generation. We set CFG to 4.0 and perform 50 denoising steps, conditioning the model on the instructive prompt, while keeping all other parameters at their default settings.

Implementation Details of TokenFlow. We use Stable Diffusion 2.1 as the base model. In the first stage, we apply DDIM inversion with 50 steps to obtain noisy latents from the input video. In the second stage, we set `pnp_f_t` = 0.8 and `pnp_attn_t` = 0.5. Inference is performed at the original video resolution, with the number of frames truncated to 40. We set CFG to 7.5 and condition the model on the descriptive prompt, using the latents from the first stage to guide 50 denoising steps.

Implementation Details of VACE. We use the Wan2.1-VACE-1.3B model for inference. Input videos are resized to a resolution of 480×832 and truncated to 33 frames. We set CFG to 3.0, `context_scale` to 1.0, and `shift_scale` to 1.0, and perform 20 denoising steps, conditioning the model on the descriptive prompt and the reference image. Empirically, we observe that directly feeding the original video causes the model to copy the foreground and ignore the control signals. To mitigate this, we convert the foreground into a scribble-style representation while preserving the original background as input. Finally, the generated videos are resized back to the original resolution.

Implementation Details of VideoPainter. We use the pre-trained VideoPainter checkpoint with the CogVideoX-5b-I2V base model. Input is resized to a resolution of 480×720 and truncated to 49 frames. The first frame is edited using Flux-Fill and serves as the starting frame for generation. During inference, the foreground mask is dilated by 10 pixels. We perform 50 denoising steps with CFG set to 6.0, conditioning the model on the descriptive prompt.

Implementation Details of Pair Diffusion and Flux-Kontext-Image. We evaluate strong image-based editors on video inference tasks, by applying them frame-by-frame. All parameters follow the default configuration.

C.5. Evaluation Metrics Implementation

Implementation Details of Background Evaluation. We first dilate original mask by 16 pixels to accommodate the settings of some editing models. We then compute the average MSE, PSNR, SSIM, and LPIPS over the remaining background region. For videos, these metrics are computed on a per-frame basis and then averaged over all frames of all videos.

Implementation Details of Foreground Evaluation. As discussed in Sec. 3 of the main text, we use CLIPScore, DINO, LPIPS, and DreamSim for foreground evaluation. Specifically, we first crop the foreground regions from the images or videos and resize them to match the spatial resolution of the reference image. For CLIPScore, DINO, and LPIPS, we use their corresponding base models to extract features from both the generated images or videos and the reference image, and then compute the cosine similarity between the two feature vectors, where a larger value indicates higher similarity in material and color. For GLCM, we set `patch_size=21`.

Implementation Details of LLM Evaluation. Given the source image or videos, the reference image, and the output of one method, we ask GPT-5 [33] and Gemini-2.5-Pro [11] to assign a score. The instruction is as follows:

Template for LLM Evaluation

You will receive three images:
A: the original image with a visible outline over the foreground region (for localization only);
B: the reference image that shows the desired material/-texture and color;
C: the candidate (edited) image to be evaluated.
Check ONLY the outlined foreground and return one integer 0..4 (number of satisfied criteria):
1) Material application is reasonable and complete.
2) Color is similar to reference.
3) Structure preserved.
4) Background stays the same as the original.
Return ONLY the integer.

Implementation Details of User Preference. To evaluate human preferences over different editing methods, we design a questionnaire that presents the results of various image and video editing approaches. Participants are asked to assess the outputs from multiple aspects and select all options they find satisfactory. The questionnaire instructions are as in Figure 2.

D. Inference cost

We report wall-clock inference time and peak GPU memory usage under the same settings, both image and video resolution is **512×512** and video has **33** frames, using default inference steps, on an A800 GPU. Results are reported in Table 2 and 3.

E. LLM Consistency with Human Annotation

Following [43, 58], we compare the discrepancy between LLM-based scores (GPT-5 and Gemini-2.5 Pro) and human preferences on 90 samples. As shown in Table 4, Gemini-2.5 Pro [11] exhibits preferences that are highly consistent with human annotations, indicating that its scoring criteria are closer to human judgments. GPT-5 [33], on the other hand, shows larger discrepancies, suggesting that its scores deviate more from human preferences and are generally stricter. Nevertheless, the relative ranking induced by

Table 2. Inference cost on videos.

Method	COCOCO	VACE	VideoPainter	AnyV2V	Ditto	Flatten	TokenFlow	ICVE	InsV2V
Time (s)↓	67.52	82.99	164.25	311.06	389.83	188.34	679.55	338.45	149.36
GPU (GB)↓	32.42	32.48	39.54	16.64	42.97	16.08	33.84	34.41	16.14
Steps	50	20	50	50	50	50	50	50	20
Method	InsVIE	Lucy-Edit	Señorita	Pair Diff	Flux-Ktx-I	Ours (S1)	Ours (S2)	Texture Remover	
Time (s)↓	93.59	22.17	118.42	314.28	940.57	**21.99**	**21.99**	1.70	
GPU (GB)↓	31.05	27.66	30.04	**12.13**	35.85	14.88	14.88	13.53	
Steps	50	50	50	30	28	20	20	3	

Table 3. Inference cost on images.

Method	BrushNet	CtrlN-Inp	Flux-Fill	SD3-Inp	Ultra-Edit	Flux-Ktx-I	Flux-Ktx-T	HiDream	HQEdit
Time (s)↓	4.31	3.37	8.54	**1.38**	2.95	29.62	29.62	44.93	1.47
GPU (GB)↓	9.24	4.77	34.41	18.44	17.94	35.85	35.85	63.47	**3.48**
Steps	50	20	50	50	50	28	28	28	30
Method	InsP2P	Qwen-I-E	Pair Diff	ZeST	Cross-I-Attn	Ours (S1)	Ours (S2)	Texture Remover	
Time (s)↓	7.80	117.17	9.89	3.93	55.72	4.70	4.70	0.40	
GPU (GB)↓	19.16	59.67	12.13	14.35	15.13	10.63	10.63	9.85	
Steps	50	50	30	30	68	20	20	3	

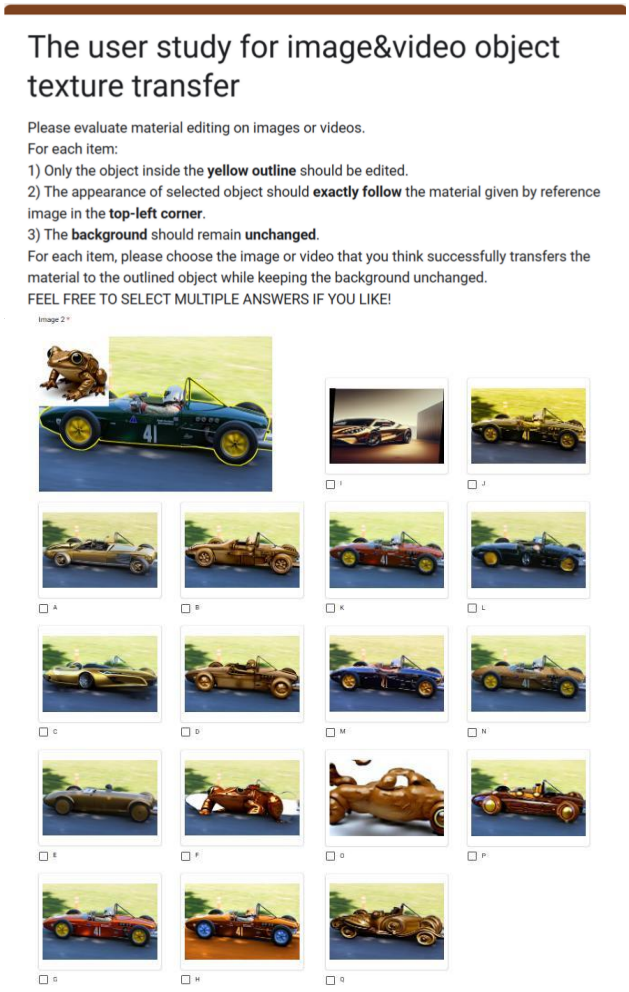


Figure 2. Questionnaire for user study.

GPT-5 still aligns well with the comparative quality of the different methods.

Table 4. LLM Consistency with human annotations.

Metric	Human	GPT-5	Gemini-2.5
Score	3.15	2.76	3.01

F. The Impact of CFG for Refaçade

To evaluate the impact of the CFG [19] scale, we conduct experiments on the Pexels validation set. In particular, a scale of 1 corresponds to using only the conditional information without any unconditional guidance. As shown in Table 5, increasing the CFG scale leads to a degradation in background quality. When the CFG scale lies between 1.0 and 2.0, the background remains relatively stable, but it deteriorates rapidly once the scale exceeds 2.5. For the foreground region, we observe that the best scores are mostly concentrated around scales of 1.5 and 2.0, where the model achieves strong material and color similarity to the reference. Figure 3 illustrates editing results on the same image under different CFG scales. When the scale is set to 1.0, the influence of the reference image is relatively weak: the marble streaks on the cup are sparse, and large regions remain white. When the scale reaches 1.5 or higher, the marble texture becomes much more pronounced. However, when CFG $\geq$ 2.5, background distortions begin to appear; for example, the wooden texture of the table becomes noticeably darker. At a scale of 4.0, clear artifacts can be observed.

G. Performance of the Texture Remover

To evaluate the performance of the Texture Remover, we render 50 pairs of textured and texture-free videos as our evaluation dataset, each at a resolution of 480×832 with

Table 5. Ablation study for CFG scales. The LPIPS for background evaluates background perseveration, while LPIPS for foreground evaluates the similarity between reference texture and generated content. CLIP, DINO and Dream are the abbreviations of CLIPScore, DINOScore and DreamSim, respectively. The best results are **boldfaced**, and the second-best results are underlined.

Scale	Background				Foreground					LLM Evaluation	
	MSE↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↓	DINO↑	LPIPS↓	Dream↑	GLCM↑	GPT-5↑	Gemini-2.5↑
1.0	**28.68**	**36.82**	**0.9500**	**0.0322**	0.7224	0.2386	0.6627	0.7303	0.8147	2.640	3.080
1.5	29.87	36.69	0.9485	0.0326	0.7331	0.2622	0.6540	0.7473	0.8967	**2.763**	**3.280**
2.0	30.82	36.61	0.9470	0.0333	0.7296	**0.2653**	**0.6526**	0.7403	0.8897	2.680	3.260
2.5	37.41	35.46	0.9402	0.0429	**0.7364**	0.2535	0.6680	**0.7488**	**0.9010**	2.760	3.020
3.0	101.04	30.85	0.8970	0.0916	0.7323	0.2559	0.6707	0.7387	0.8903	2.580	2.980

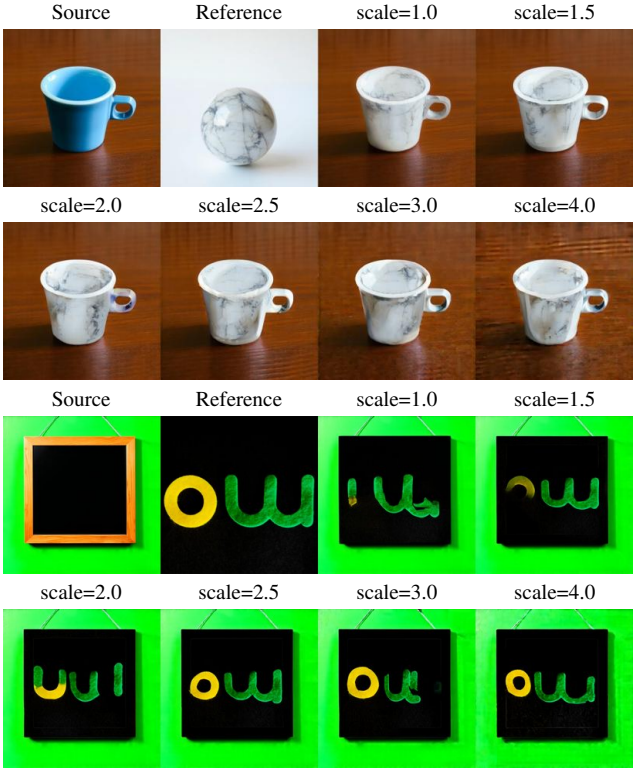


Figure 3. Qualitative visualization of Refaçade with CFG scales. First 2 rows: patch size=10%; bottom 2 rows: patch size=100%.

33 frames. We use exactly the same camera parameters and object motion for each pair to ensure strict correspondence.

Performance of the original Texture Remover. As shown in Table 6, for the original Texture Remover, increasing the number of inference steps reduces the reconstruction error, but at the cost of substantially higher computation time. In practice, using 50 inference steps is impractical, which highlights the importance of distilling the model to operate reliably with fewer steps.

The performance of distilled Texture Remover. We find that CFG is unnecessary for the Texture Remover (as shown in Table 6). Moreover, to further accelerate inference, we reduce the original 50 denoising steps to 3 via distillation,

Table 6. Ablation study for inference steps and CFG scales of primitive texture remover. The value of *Ewarp* falls within the range of 1×10^{-3}. The best results are **boldfaced**, and the second-best results are underlined.

Infer Steps	Scale	MSE↓	PSNR↑	SSIM↑	LPIPS↓	EWarp↓	Time(s)↓
3	1.0	22.91	35.52	0.9719	0.0279	0.4872	**5.7787**
10	1.0	21.68	35.66	0.9729	0.0263	0.4523	9.2262
20	1.0	20.18	36.01	0.9741	0.0239	**0.4383**	18.5865
50	1.0	**17.33**	**36.56**	**0.9767**	**0.0250**	0.4407	46.7409
50	1.5	57.54	31.75	0.9674	0.0427	1.6259	93.7643
50	2.0	191.51	26.47	0.9510	0.0916	4.7785	93.7643
50	2.5	299.97	24.38	0.9408	0.1103	6.4976	93.7643

Table 7. Ablation study for distillation steps of texture remover, inference step $= 3$. The value of *Ewarp* falls within the range of 1×10^{-3}. The best results are **boldfaced**, and the second-best results are underlined.

Distill Steps	MSE↓	PSNR↑	SSIM↓	LPIPS↑	EWarp↓
100	19.41	36.04	0.9720	**0.0300**	**0.4423**
200	21.37	35.72	0.9729	0.0341	0.4533
300	**19.16**	**36.55**	0.9730	0.0371	0.4669
400	19.35	36.47	0.9724	0.0381	0.4739
500	19.51	36.40	0.9726	0.0373	0.4678
600	20.32	36.01	**0.9739**	0.0396	0.4983
700	25.57	31.11	0.9718	0.0331	0.6631
800	26.76	34.94	0.9693	0.0359	0.7460

making the Texture Remover fast enough to be integrated into training. Table 7 reports the results of applying DMD2 distillation with different training steps and evaluating the distilled 3-step models. Compared with the original (undistilled) 3-step Texture Remover, the distilled variants consistently achieve better performance. We ultimately select the checkpoint distilled for 300 steps, as it attains the lowest MSE and highest PSNR. Notably, when distillation continues beyond 600 steps, all metrics deteriorate rapidly, indicating overfitting. Figure 4 compares the original Texture Remover and the distilled variant under the same 3-step denoising setting. The original Texture Remover produces noticeably blurred regions, which may interfere with subsequent Refaçade training, whereas the distilled Texture Remover yields cleaner and more reliable results.

Table 8. Evaluation of different texture extraction strategies.

Method	Foreground					Motion	LLM Evaluation	
	CLIP↑	DINO↑	LPIPS↓	Dream↑	GLCM↑	EWarp↓	GPT-5↑	Gemini↑
Org Ref	0.7406	0.3050	0.6249	0.7575	0.8286	1.4300	2.04	2.25
Features	0.7515	**0.3241**	0.6107	0.7684	0.8437	1.4410	2.78	3.11
Ours (S2)	**0.7524**	**0.3241**	**0.6080**	**0.7742**	**0.8516**	**1.4248**	**2.82**	**3.25**

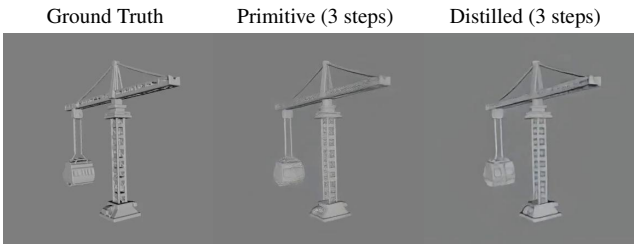


Figure 4. Qualitative comparison of the primitive Texture Remover and its distilled variant at 3 denoising steps.

H. Evaluation on Challenging Dataset

H.1. Evaluation on Small-resolution Images

To further investigate the performance of different methods on images, we conduct experiments on the ECSSD [37] dataset. The image resolution in this dataset is relatively low, typically between 200 and 500 pixels on the longer side. We discard samples whose foreground mask area is smaller than 5% or larger than 90%.

Table 9 reports the background preservation and foreground texture similarity of all methods. All methods perform inference at the original image resolution. Our approach (Stage 1 and Stage 2) achieves the lowest MSE and LPIPS, together with the highest PSNR and SSIM, indicating the strongest background preservation. Moreover, Stage 2 further improves the retexturing ability over Stage 1, showing higher texture consistency.

In Figure 5, although some methods can roughly turn the train into a beige color, such as Qwen-Image-Edit and Flux-Kontext-Text, they still fail to match the overall color and texture of the reference image. This reflects a fundamental limitation of using text as the sole conditioning signal: even if *beige color* and *fabric-like* are explicitly mentioned, it is difficult to specify the exact RGB values in natural language, and even harder to describe them within a short prompt. Current models also struggle to accurately interpret such RGB-level textual descriptions.

H.2. Evaluation on Fast-Motion Videos

Editing fast moving objects in videos is particularly challenging. To evaluate this setting we conduct experiments on the DAVIS [34] dataset. For a fair comparison we compute all metrics on the first 33 frames of the output videos.

Table 10 shows that our method achieves state of the art performance on the foreground background and LLM based evaluation metrics, but performs worse in terms of EWarp, indicating slightly reduced temporal consistency. We attribute this limitation to the Texture Remover. Although we synthesize its training set by rendering many pairs of videos with fast moving objects, these data lack nonrigid deformations and only simulate motion through rotations of rigid objects, which introduces a domain gap compared with real world videos. Figure 11 provides some visual results.

I. Exploration for extracting texture

We retrain our model using: (i) original reference without jigsaw; (ii) features extracted by image encoder combined with Dinov2 and VGG as in Pair Diffusion. Raw pixels without jigsaw performs worst on LLM evaluation, where model only learns paste and crop as shown in Figure 6a. Table 8 and Figure 6b show that extracting features via an image encoder mitigates this issue, but still performs a little weaker on color transfer.

J. Qualitative Results

J.1. Jigsaw doesn't Break Global Texture

Jigsaw with small patch size breaks the global texture, but increasing the patch scales will preserve more global information. As shown in Table 4 (main text), the 100% patch size (i.e., no jigsaw) achieves comparable performance. Results are shown in Figure 7.

J.2. Impact of Lightning on Reference Images

We handle lighting implicitly via supervision on real-world videos with consistent lighting, preventing reference lighting copy. Results are consistent across variations in Figure 8.

J.3. Comparison with Different Conditions

Canny and Grayscale leak original texture, while depth doesn't contain fine-grained structure information. HED sometimes leads to flattened results. Results are shown in Figure 9.

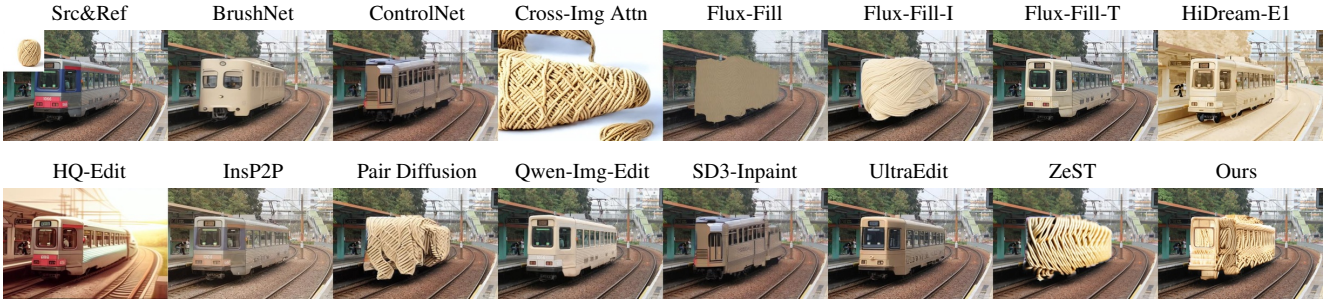
K. Limitations and Failure Cases

K.1. Limitations

We believe that the main limitations of our method stem from the Texture Remover. First, the capability of the Texture Remover is entirely inherited from Hunyuan3D [56]. Hunyuan3D is relatively insensitive to textual details, and small characters in particular tend to be treated as texture noise and removed during image-to-mesh reconstruction. This behavior is then learned by the Texture Remover, which causes Refaçade to miss certain fine-grained details. Second, when training the Texture Remover we rely on 3D

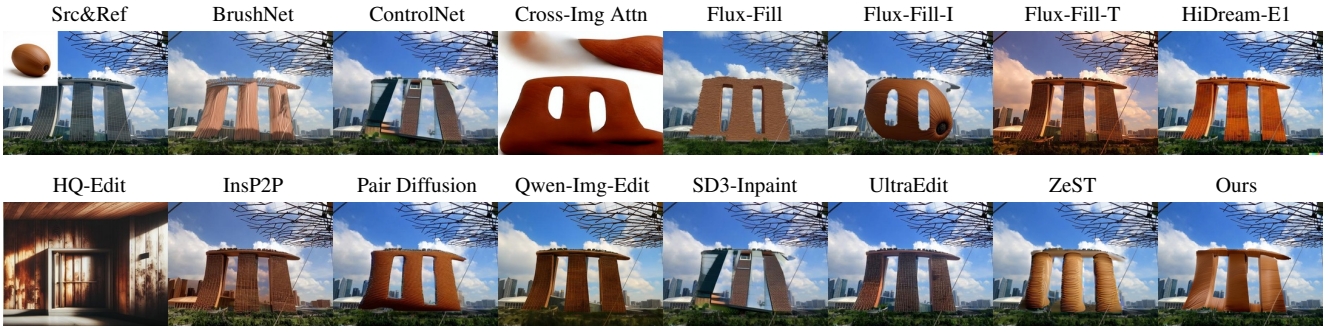
Table 9. Evaluation on ECSSD image dataset. The LPIPS for background evaluates background perseveration, while LPIPS for foreground evaluates the similarity between reference texture and generated content. CLIP, DINO and Dream are the abbreviations of CLIPScore, DINOScore and DreamSim, respectively. The best results are **boldfaced**, and the second-best results are underlined.

Method	Type	Background				Foreground					LLM Evaluation	
		MSE↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	DINO↑	LPIPS↓	Dream↑	GLCM↑	GPT-5↑	Gemini↑
BrushNet [25]	Inpainting	361.22	24.63	0.8471	0.0592	0.7086	0.2069	0.7426	0.7358	0.8483	2.254	1.271
ControlNet-Inp [32]		113.14	29.80	0.8487	0.2386	0.6901	0.1808	0.7701	0.7077	0.8219	1.983	0.864
Flux-Fill [5]		1148.02	19.89	0.5431	0.1699	0.7200	0.2071	0.7190	0.7599	0.8475	2.034	0.983
SD3-Inpaint [15]		113.14	29.80	0.8487	0.0416	0.6901	0.1774	0.7701	0.7077	0.8219	1.797	0.932
Pair Diff [17]		170.62	27.78	0.7983	0.0494	0.7858	0.4680	0.6377	0.8268	0.8146	2.764	2.586
ZeST [10]		109.13	30.75	0.8459	0.0392	0.7620	0.3474	0.7220	0.8025	0.8564	2.702	2.586
Cross-Img Attn [2]	General	6344.89	13.46	0.3568	0.3191	0.7653	0.4732	0.6857	0.8189	0.8560	1.856	1.020
UltraEdit [53]		62.34	32.07	0.9049	0.0255	0.6837	0.1708	0.7679	0.7006	0.8257	2.644	1.763
Flux-Kont-I [27]		59.08	31.88	0.9133	0.0367	0.7918	0.4902	0.6418	0.8253	0.8754	2.407	0.847
Flux-Kont-T [27]		1651.74	20.02	0.5558	0.1038	0.6789	0.1719	0.7267	0.7029	0.8454	2.322	2.102
HiDream-E1 [8]		2402.24	22.39	0.7692	0.1282	0.7008	0.2073	0.7326	0.7223	0.8500	2.542	1.746
HQ-Edit [22]		7733.84	10.39	0.2732	0.3621	0.7017	0.2172	0.7461	0.7223	0.8314	1.288	0.983
InsP2P [7]		2779.51	15.93	0.4687	0.2177	0.6933	0.1760	0.7340	0.7155	0.8442	1.881	1.661
Qwen-I-Edit [44]		1596.50	20.19	0.5489	0.1369	0.6864	0.2156	0.7228	0.7135	0.8455	2.797	**2.764**
Ours(stage1)	Inpainting	**23.45**	**37.66**	**0.9653**	**0.0095**	0.7365	0.3177	0.6726	0.7809	0.8836	**2.864**	2.740
Ours(stage2)		24.73	37.99	0.9630	0.0101	**0.7934**	**0.5050**	**0.6395**	**0.8407**	0.8937	2.831	**2.764**



Instructive prompt: *Paint the train in a soft beige color, giving it a smooth, slightly textured fabric-like appearance reminiscent of tightly wound yarn.*

Descriptive prompt: *A train in creamy beige, crafted from matte fabric-like material, exudes a cozy aesthetic reminiscent of artisanal craftsmanship.*



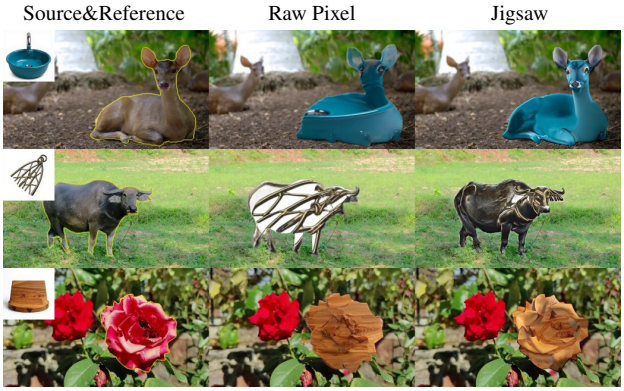
Instructive prompt: *Cover the building in a rich, warm brown tone resembling wood grain texture for a natural and rustic appearance.*

Descriptive prompt: *A building in warm terracotta, crafted from smooth, polished clay with subtle wood-like textures.*

Figure 5. Qualitative visualization on ECSSD. Each pair of rows uses the instructive and descriptive prompts shown below the images.

Table 10. Evaluation results on DAVIS dataset. The LPIPS for background evaluates background perseveration, while LPIPS for foreground evaluates the similarity between reference texture and generated content. CLIP, DINO and Dream are the abbreviation of CLIPScore, DINOScore and DreamSim, respectively. Ewarp is at the range of 1×10^{-3}. The best results are **boldfaced**, the second-best are underlined.

Method	Type	Background				Foreground					Motion	LLM Evaluation	
		MSE↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	DINO↑	LPIPS↓	Dream↑	GLCM↑	EWarp ↓	GPT-5↑	Gemini↑
COCOCO [60]	Inpainting	3353.94	14.10	0.5452	0.4981	0.6979	0.1055	0.8076	0.6967	0.7239	11.7707	1.644	1.644
VACE [24]		2175.58	15.24	0.6199	0.3409	0.7182	0.1699	0.7586	0.7141	0.7172	11.8412	2.033	2.433
VideoPainter [4]		262.14	24.91	0.8132	0.2425	0.7098	0.1573	0.7577	0.7105	0.7183	13.6478	1.811	1.700
AnyV2V [26]	General	1189.29	19.00	0.6139	0.3440	0.7182	0.1538	0.7533	0.7317	0.7231	12.4481	2.000	2.167
Ditto [3]		1882.43	17.51	0.6784	0.4238	0.6720	0.1149	0.8118	0.6880	0.7313	9.2721	1.300	1.333
Flatten [12]		3662.32	13.23	0.5597	0.5924	0.7131	0.1499	0.7524	0.7280	0.6837	11.2783	1.686	1.070
TokenFlow [16]		1165.58	18.18	0.6563	0.3972	0.7088	0.1311	0.7523	0.7241	0.7369	9.2764	1.778	1.133
ICVE [31]		1513.94	18.55	0.6501	0.4014	0.7018	0.1517	0.7856	0.7158	0.7154	10.8412	1.622	1.056
InsV2V [9]		3173.99	13.77	0.5379	0.6097	0.6900	0.1075	0.7602	0.7113	0.7158	12.4250	1.822	1.422
InsVIE [45]		4972.99	11.87	0.4316	0.5970	0.7091	0.1447	0.8144	0.7069	0.7253	31.3162	1.583	1.144
Lucy-Edit [40]		430.00	23.73	0.7680	0.2708	0.6966	0.1563	0.7576	0.6899	0.7232	13.2379	2.231	2.489
Señorita [59]		290.22	24.56	0.7739	0.3534	0.6987	0.1819	0.7456	0.6992	0.7094	**8.6078**	2.139	2.178
Pair Diffusion [17]	Image	241.90	25.73	0.7414	0.2123	0.7102	0.2337	0.6530	0.7263	0.7667	17.7545	1.866	1.908
Flux-Kont-I [27]		62.71	31.57	0.9005	0.0868	0.7058	0.2099	0.7025	0.7283	0.7523	30.0722	1.584	1.608
Ours(stage1)	Inpainting	51.33	32.20	0.9160	0.0805	0.7183	0.2108	0.6529	0.7269	0.7635	11.1025	2.622	3.150
Ours(stage2)		**48.42**	**32.33**	**0.9163**	**0.0795**	**0.7221**	**0.2426**	**0.6373**	**0.7338**	**0.7681**	10.8550	**2.654**	**3.200**



(a) Pixel-space injection comparison.

(b) Feature-space injection comparison.

Figure 6. Visual comparisons of texture injection strategies. (a) Raw pixel-level injection. (b) Feature-level injection.

Figure 7. Visualization results of global texture.

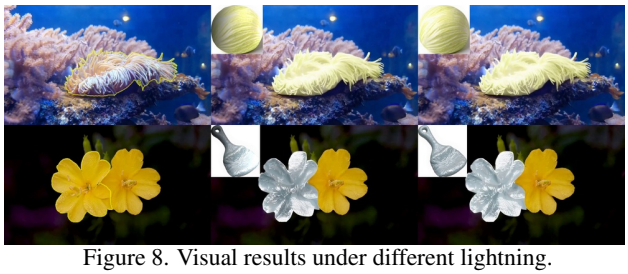


Figure 8. Visual results under different lightning.

meshes that are rendered into dynamic videos by translating or rotating the mesh. The reconstructed 3D object is static and cannot deform, which leaves a gap with respect to real-world videos where objects often undergo nonrigid motion. Third, for videos with large motion, some frames may contain motion blur. Such cases are absent from the Texture Remover training data, so the model cannot handle them well, which can lead to structural collapse and chaotic geometry in some Refaçade outputs.

K.2. Failure Cases

As shown in Figure 10, our model tends to strictly follow the texture of the reference image while overlooking the aesthetic quality of the generated visual content. We attribute this behavior to two factors. First, the CFG scale

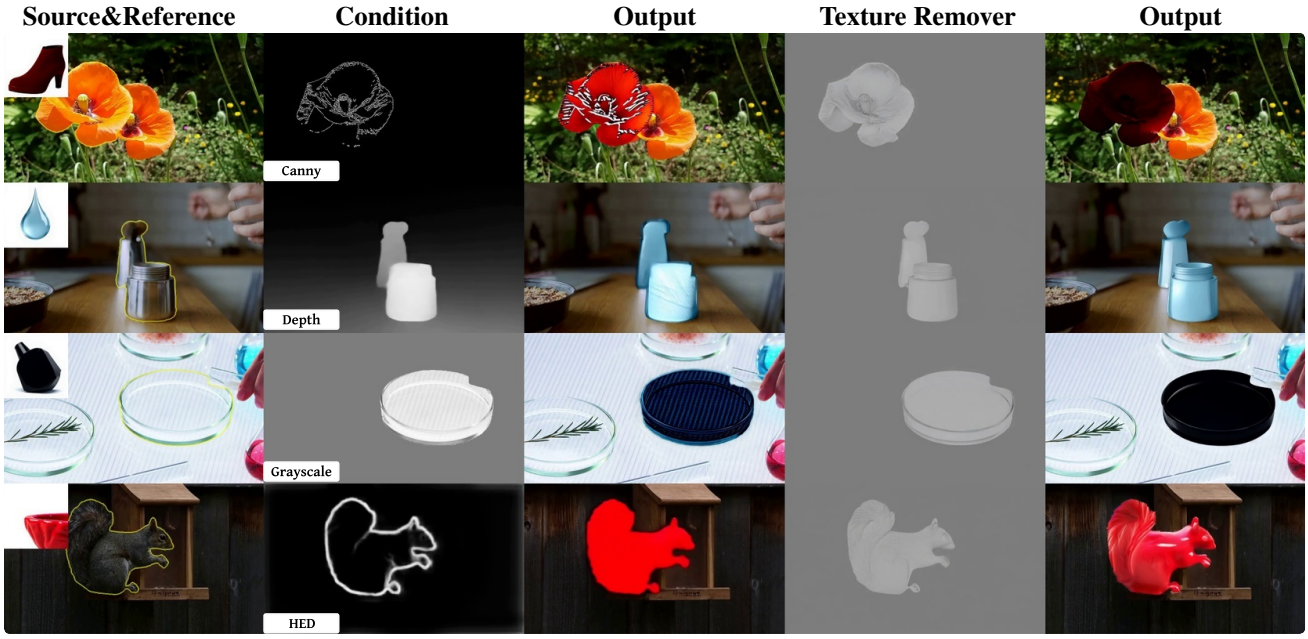
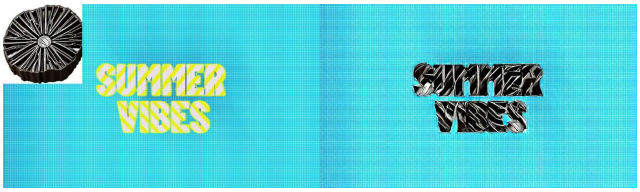


Figure 9. Comparison results of different conditions.

(a) Object merging introduced by mask dilation.



(b) Texture remover struggles with extreme high-frequency details.

(c) Texture remover exhibits low sensitivity to textual information.

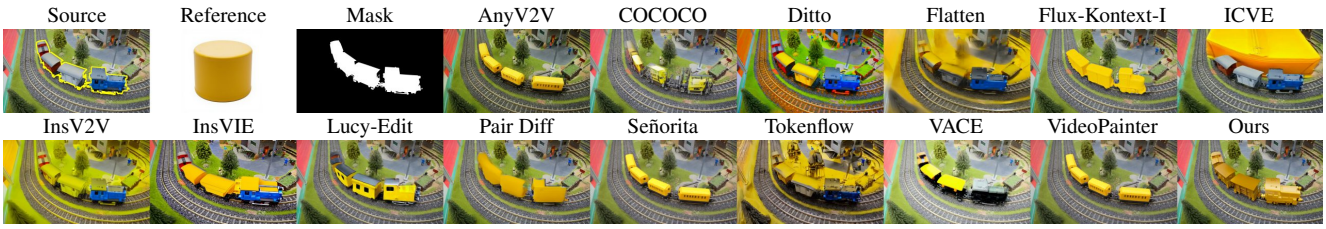
(d) **Refaçade** sometimes fails to reshuffle patches properly.

Figure 10. Visual results of failure cases.

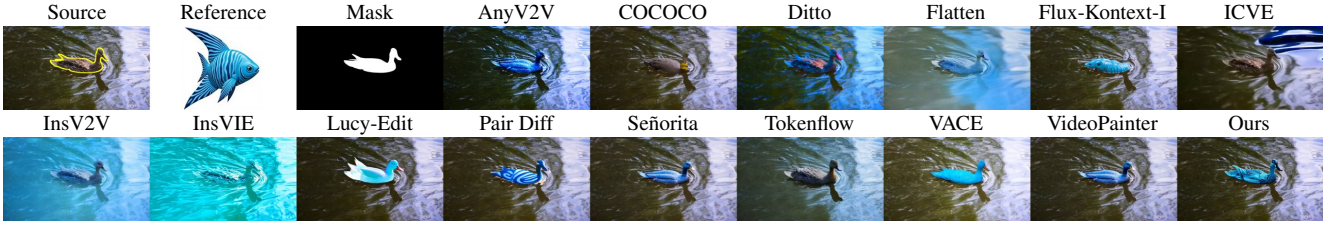
used in these examples is suboptimal. Better visual quality can often be obtained by tuning this scale. Second, scenes that contain untextured objects are more susceptible to reconstruction failures, which in turn lowers the overall success rate of editing.

L. Future Works

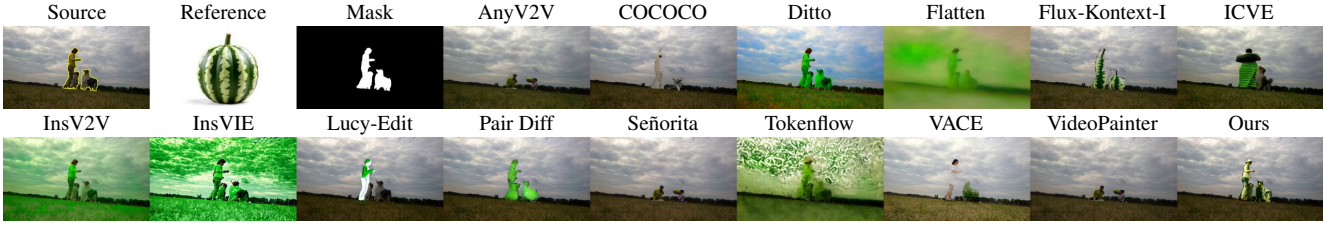
In future work, we plan to expand the dataset used to train the Texture Remover. Since the current meshes are all rigid and relatively coarse, we aim to incorporate more detailed 4D meshes to enhance the remover's capability and thereby improve overall robustness. In addition, to further boost the aesthetic quality of **Refaçade**, we plan to explore reinforcement learning with reward models.



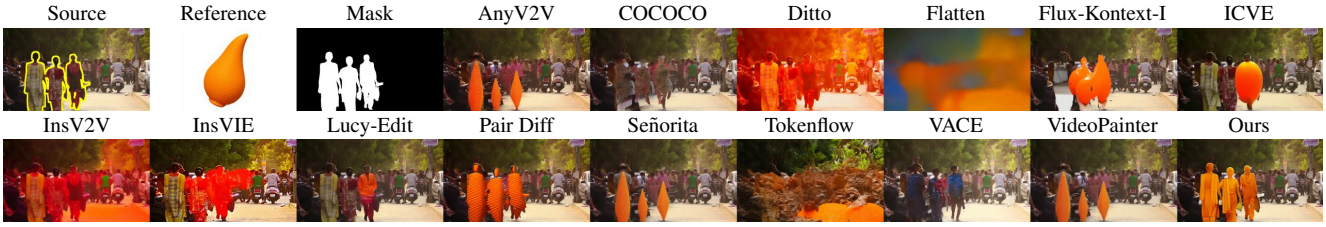
Instructive prompt: *Paint the train car and train engine in vibrant yellow, giving them a smooth, glossy plastic-like finish.*
Descriptive prompt: *Train car and train engine in vibrant yellow, featuring a smooth, glossy plastic-like finish.*



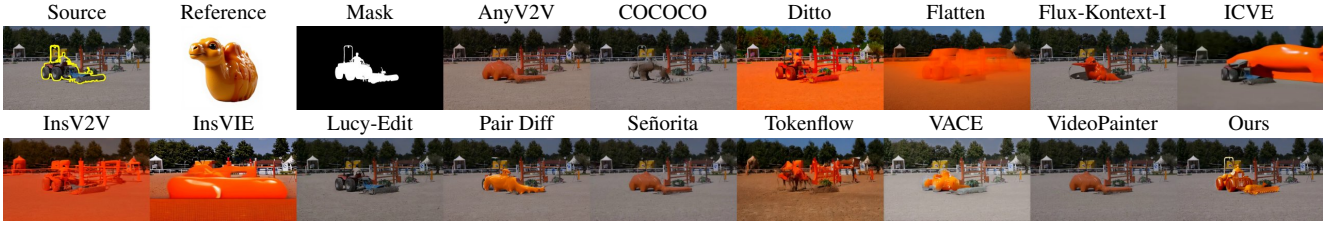
Instructive prompt: *Transform the duck into alternating light and dark icy blue, adorned with a glossy ceramic texture.*
Descriptive prompt: *A duck in alternating light and dark icy blue, adorned with a glossy ceramic finish.*



Instructive prompt: *Dress the person and dogs in vibrant green and creamy white striped attire, mimicking the glossy, smooth texture of a ripe watermelon.*
Descriptive prompt: *A person and dogs in vibrant green and creamy white, adorned with a smooth melon-like texture.*



Instructive prompt: *Turn the people into vibrant orange, covered with glossy silicone texture.*
Descriptive prompt: *Orange, glossy silicone teardrop-shaped people with textured ridges.*



Instructive prompt: *Restyle the tractor into a glossy orange ceramic material.*
Descriptive prompt: *A tractor in warm orange, crafted from glossy plastic, stands as a barrier and obstacle, its smooth surface reflecting light while its hump and features suggest resilience and strength.*

Figure 11. Comparison results of **Refaçade** and baselines on DAVIS. *Best viewed with Adobe Acrobat Reader; click to play.*

Figure 12. Visual results of Texture Remover.

Figure 13. Visual results of **Refaçade** on images.

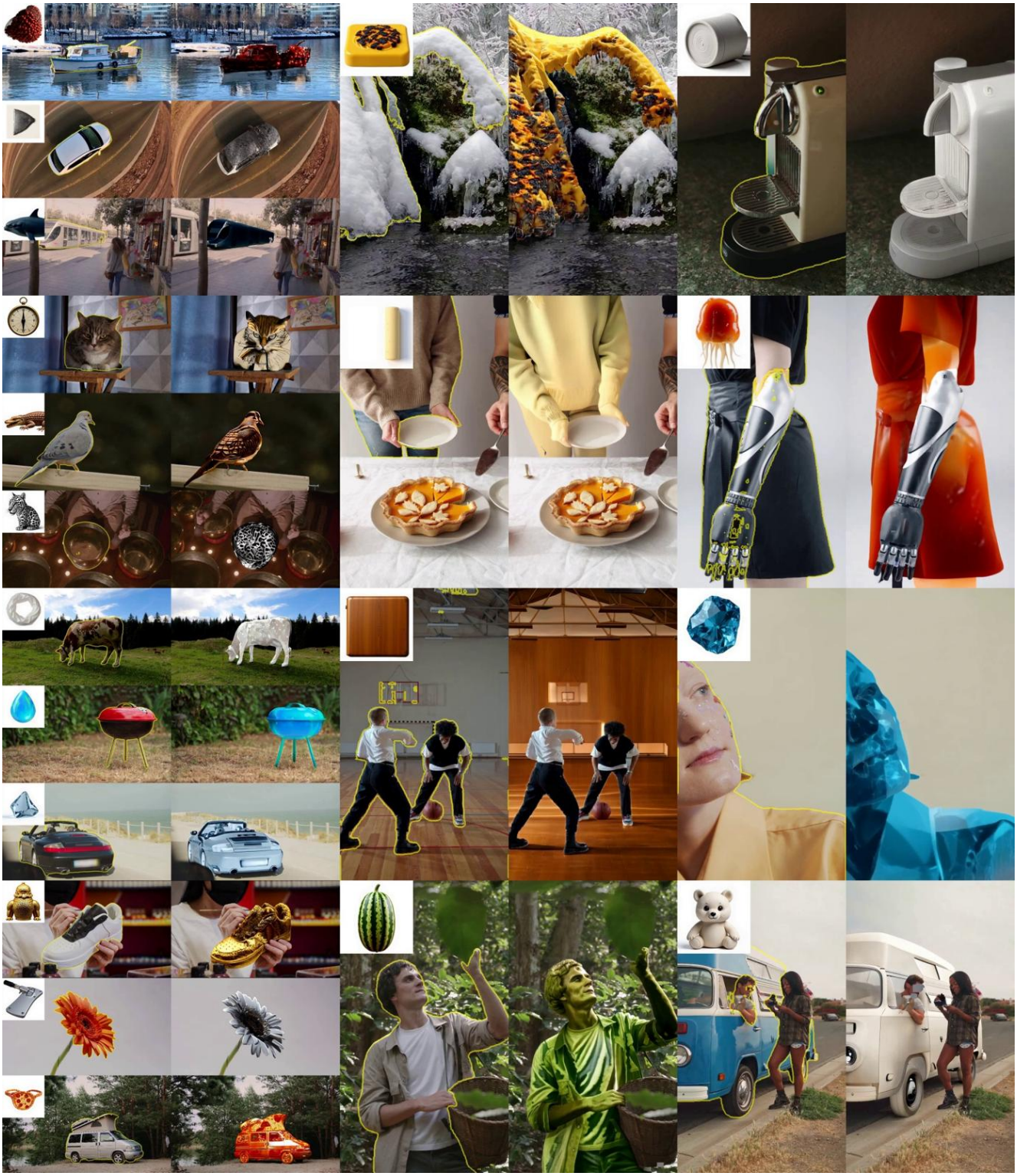


Figure 14. Visual results of **Refaçade** on videos. *Best viewed with Adobe Acrobat Reader; click to play.*

References

[1] Sakshi Agarwal, Gabe Hoope, and Erik B. Sudderth. Vipaint: Image inpainting with pre-trained diffusion models via variational inference, 2024. 1

[2] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH 2024 conference papers*, pages 1–12, 2024. 9

[3] Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, et al. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv preprint arXiv:2510.15742*, 2025. 10

[4] Yuxuan Bian, Zhaoyang Zhang, Xuan Ju, Mingdeng Cao, Liangbin Xie, Ying Shan, and Qiang Xu. Videopainter: Any-length video inpainting and editing with plug-and-play context control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 1, 10

[5] Black Forest Labs. Black forest labs. https://github.com/black-forest-labs/flux/, 2024. 1, 9

[6] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023. 1

[7] 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. 9

[8] Qi Cai, Jingwen Chen, Yang Chen, Yehao Li, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Yiheng Zhang, Fengbin Gao, Peihan Xu, et al. Hidream-i1: A high-efficient image generative foundation model with sparse diffusion transformer. *arXiv:2505.22705*, 2025. 9

[9] Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. *arXiv preprint arXiv:2311.00213*, 2023. 10

[10] Ta-Ying Cheng, Prafull Sharma, Andrew Markham, Niki Trigoni, and Varun Jampani. Zest: Zero-shot material transfer from a single image. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024. 9

[11] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 5

[12] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922*, 2023. 10

[13] Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and Min Zhang. Parameter competition balancing for model merging. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 1

[14] Guodong Du, Xuanning Zhou, Junlin Li, Zhuo Li, Zesheng Shi, Wanyu Lin, Ho-Kin Tang, Xiucheng Li, Fangming Liu, Wenya Wang, Min Zhang, and Jing Li. Knowledge fusion of large language models via modular skillpacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. 1

[15] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 9

[16] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 10

[17] Vidit Goel, Elia Peruzzo, Yifan Jiang, Dejia Xu, Xingqian Xu, Nicu Sebe, Trevor Darrell, Zhangyang Wang, and Humphrey Shi. Pair diffusion: A comprehensive multimodal object-level image editor. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8609–8618, 2024. 9, 10

[18] Bohai Gu, Hao Luo, Song Guo, and Peiran Dong. Advanced video inpainting using optical flow-guided efficient diffusion. *arXiv e-prints*, pages arXiv–2412, 2024. 1

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 6

[20] Jiahao Hu, Tianxiong Zhong, Xuebo Wang, Boyuan Jiang, Xingye Tian, Fei Yang, Pengfei Wan, and Di Zhang. Vivid-10m: A dataset and baseline for versatile and interactive video local editing. *arXiv preprint arXiv:2411.15260*, 2024. 1

[21] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025. 1

[22] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024. 9

[23] Yueru Jia, Aosong Cheng, Yuhui Yuan, Chuke Wang, Ji Li, Huizhu Jia, and Shanghang Zhang. Designedit: Unify spatial-aware image editing via training-free inpainting with a multi-layered latent diffusion framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3958–3966, 2025. 1

[24] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025. 1, 10

[25] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, pages 150–168. Springer, 2024. 1, 9

[26] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. Anyv2v: A plug-and-play framework for any videoto-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2(3): 5, 2024. 10

[27] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. 2, 9, 10

[28] Minhyeok Lee, Suhwan Cho, Chajin Shin, Jungho Lee, Sunghun Yang, and Sangyoun Lee. Video diffusion models are strong video inpainter. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4526–4533, 2025. 1

[29] Ruibin Li, Tao Yang, Song Guo, and Lei Zhang. Rorem: Training a robust object remover with human-in-the-loop. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14024–14035, 2025. 1

[30] Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. Diffueraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025. 1

[31] Xinyao Liao, Xianfang Zeng, Ziye Song, Zhoujie Fu, Gang Yu, and Guosheng Lin. In-context learning with unpaired clips for instruction-based video editing. *arXiv preprint arXiv:2510.14648*, 2025. 10

[32] mikonvergence. Controlnetinpaint: Inpaint images with controlnet. https://github.com/mikonvergence/ControlNetInpaint, 2023. 1, 9

[33] OpenAI. GPT-5 is here, 2025. 2, 5

[34] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 8

[35] Pexels. https://www.pexels.com/, 2024. 1

[36] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[37] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4): 717–729, 2015. 8

[38] Stability AI Team. Introducing stable diffusion 3.5. https://stability.ai/news/introducing-stable-diffusion-3-5, 2024. Accessed 2025-10-28. 1

[39] Wenhao Sun, Xue-Mei Dong, Benlei Cui, and Jingqun Tang. Attentive eraser: Unleashing diffusion model's object removal potential via self-attention redirection guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 20734–20742, 2025. 1

[40] DecartAI Team. Lucy edit: Open-weight text-guided video editing, 2025. Accessed: 2025-11-13. 10

[41] Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18359–18369, 2023. 1

[42] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems*, 36:7594–7611, 2023. 1

[43] 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. 5

[44] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 9

[45] Yuhui Wu, Liyi Chen, Ruibin Li, Shihao Wang, Chenxi Xie, and Lei Zhang. Insvie-1m: Effective instruction-based video editing with elaborate dataset construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16692–16701, 2025. 10

[46] Liangbin Xie, Daniil Pakhomov, Zhonghao Wang, Zongze Wu, Ziyan Chen, Yuqian Zhou, Haitian Zheng, Zhifei Zhang, Zhe Lin, Jiantao Zhou, et al. Turbofill: Adapting few-step text-to-image model for fast image inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7613–7622, 2025. 1

[47] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22428–22437, 2023. 1

[48] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 1

[49] Shiyuan Yang, Zheng Gu, Liang Hou, Xin Tao, Pengfei Wan, Xiaodong Chen, and Jing Liao. Mtv-inpaint: Multi-task long video inpainting. *arXiv preprint arXiv:2503.11412*, 2025. 1

[50] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 4

[51] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. 4

[52] Zhixing Zhang, Bichen Wu, Xiaoyan Wang, Yaqiao Luo, Luxin Zhang, Yinan Zhao, Peter Vajda, Dimitris Metaxas, and Licheng Yu. Avid: Any-length video inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7162–7172, 2024. 1

[53] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024. 9

[54] Jixin Zhao, Shangchen Zhou, Zhouxia Wang, Peiqing Yang, and Chen Change Loy. Objectclear: Complete object removal via object-effect attention. *arXiv preprint arXiv:2505.22636*, 2025. 1

[55] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023. 2

[56] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 8

[57] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *European Conference on Computer Vision*, pages 195–211. Springer, 2024. 1

[58] Bojia Zi, Weixuan Peng, Xianbiao Qi, Jianan Wang, Shihao Zhao, Rong Xiao, and Kam-Fai Wong. Minimax-remover: Taming bad noise helps video object removal. *arXiv preprint arXiv:2505.24873*, 2025. 1, 5

[59] Bojia Zi, Penghui Ruan, Marco Chen, Xianbiao Qi, Shaozhe Hao, Shihao Zhao, Youze Huang, Bin Liang, Rong Xiao, and Kam-Fai Wong. Senorita-2m: A high-quality instruction-based dataset for general video editing by video specialists. *arXiv preprint arXiv:2502.06734*, 2025. 1, 10

[60] Bojia Zi, Shihao Zhao, Xianbiao Qi, Jianan Wang, Yukai Shi, Qianyu Chen, Bin Liang, Rong Xiao, Kam-Fai Wong, and Lei Zhang. Cococo: Improving text-guided video inpainting for better consistency, controllability and compatibility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11067–11076, 2025. 1, 10