Refaçade: Editing Object with Given Reference Texture

Youze Huang[1,*] Penghui Ruan[2,*] Bojia Zi[3,*] Xianbiao Qi[4,†] Jianan Wang[5] Rong Xiao[4]
[1]University of Electronic Science and Technology of China [2]The Hong Kong Polytechnic University
[3]The Chinese University of Hong Kong [4]IntelliFusion Inc. [5]Astribot Inc.

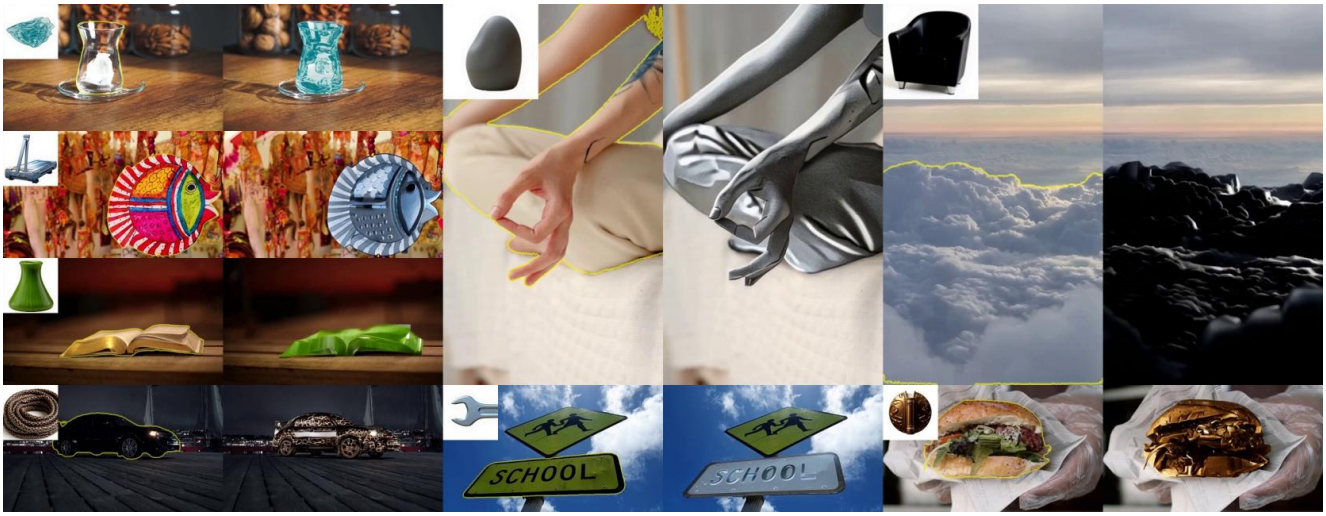

Figure 1. Visual results of Refaçade on videos.

Abstract

Recent advances in diffusion models have brought remarkable progress in image and video editing, yet some tasks remain underexplored. In this paper, we extend Object Retexture *into video domain, which transfers local textures from a reference object to a target object in images or videos. To perform this task, a straightforward solution is to use ControlNet conditioned on the source structure and the reference texture. However, this approach suffers from limited controllability due to two reasons: conditioning on the raw reference image introduces unwanted structural information, and this method fails to disentangle visual texture and structure information of the source. To address this problem, we proposed a method, namely* ***Refaçade****, that consists of two key designs to achieve precise and controllable texture transfer in both images and videos. First, we employ a texture remover trained on paired textured/untextured 3D mesh renderings to remove appearance information while preserving geometry and motion of source videos. Second, we disrupt the reference's global layout using a jigsaw permutation, encouraging the model to focus on local texture statistics rather than global layout of object. Extensive experiments demonstrate superior visual quality, precise editing, and controllability, outperforming strong baselines in both quantitative and human evaluations. Code is available at* https://github.com/fishZe233/Refacade.

1. Introduction

In recent years, diffusion models [7, 8, 10, 12, 13, 24, 30, 35, 47, 49, 50, 56, 58, 63, 64, 67–69, 76, 83, 86] have driven remarkable progress in image and video generation. In early stage, UNet-based architectures, exemplified by models such as StableDiffusionV1.5 [8] and AnimateDiff [24] have demonstrated impressive capabilities in generating high-quality images and videos. More recently, the field has advanced with the introduction of transformer-based architectures, as seen in groundbreaking works such as flux [7], Qwen-Image [69], Sora [50], HunyuanVideo [35] and Wan2.1 [64], which employ DiT-based structures [52] to achieve unprecedented generation quality.

Parallel to these advancements, diffusion-based editing techniques [1, 6–9, 16, 20, 21, 23, 28, 31, 34, 36, 39–41, 45, 46, 54, 59, 61, 65, 66, 70, 73–75, 79, 82, 84, 86–

* Equal contribution, † Corresponding author.

Figure 2. Visual results of Refaçade on images.

88, 88, 89] have also seen significant progress. However, some editing tasks remain insufficiently explored. In this study, we focus on an editing task termed **Object Retexture**, which aims to transfer texture patterns from a reference image onto a target object within a video while preserving the target's geometric structure and leaving surrounding regions unmodified. The fundamental challenge of this task lies in the disentanglement of two key visual components: *texture* (surface patterns, colors, and material properties) and *structure* (shape, geometry, and spatial layout). Specifically, Object Retexture requires: (1) decoupling texture information from the reference image while discarding its structural characteristics; (2) decoupling the target object's structure from the input video while allowing its texture to be modified; and (3) recombining the reference texture with the target structure to generate coherent edited results. This explicit separation ensures that only surface appearance is transferred from the reference, while the target object's geometric details remain intact.

Nevertheless, Object Retexture can be positioned as a specialized subtask within the broader domain of appearance editing. A natural solution is to condition the control conditions (e.g., HED edges or Canny edges) via ControlNet [80] from source videos to preserve structure, while utilizing the reference image to provide texture information. However, we find this approach fundamentally unsuitable for Object Retexture due to two critical limitations in disentanglement: First, *traditional control signals fail to fully decouple texture from structure*. Conventional control conditions such as depth maps, edge maps, or normal maps are designed to capture geometric information, yet they inevitably retain residual texture cues—such as surface patterns, material boundaries, or color gradients—that should be modifiable rather than preserved. This incomplete disentanglement prevents clean separation between what should be retained (target structure) and what should be modified (target texture). Second, *directly conditioning on the raw reference image introduces unwanted structural information*. When the entire reference image is used as a conditioning signal without proper decoupling, the model inadvertently transfers not only the desired texture patterns but also the reference's geometric characteristics, such as object shape, pose, and spatial layout. This structural leakage from the reference contaminates the target object, resulting in unintended deformations that violate the core requirement of preserving the target's original geometry.

To address these limitations, we propose **Refaçade**, a novel framework designed to enhance controllability and suppress unwanted information during texture transfer. Our method comprises two key components. *First, we replace traditional control conditions with texture-free representations rendered from 3D object meshes, which preserve the structural information of the original object while excluding color and texture cues.* To avoid the computational overhead of 3D construction and rendering, we train a texture-remover that directly eliminates texture in the image/video space, eliminating the computational burden associated with 3D lifting and 2D reprojection operations. To achieve fast and accurate texture removal for both images and videos, we train a generator based on Wan2.1 [64] and further distill it using DMD2 [78], reducing the sampling steps from 50 to just 3. *Second, we introduce a jigsaw permutation strategy that shuffles the reference image to disrupt its spatial structure.* This forces the model to concentrate on the texture itself rather than the object's shape, effectively preventing the transfer of undesired structural information to the edited object. By combining these two strategies, our approach completely removes the original

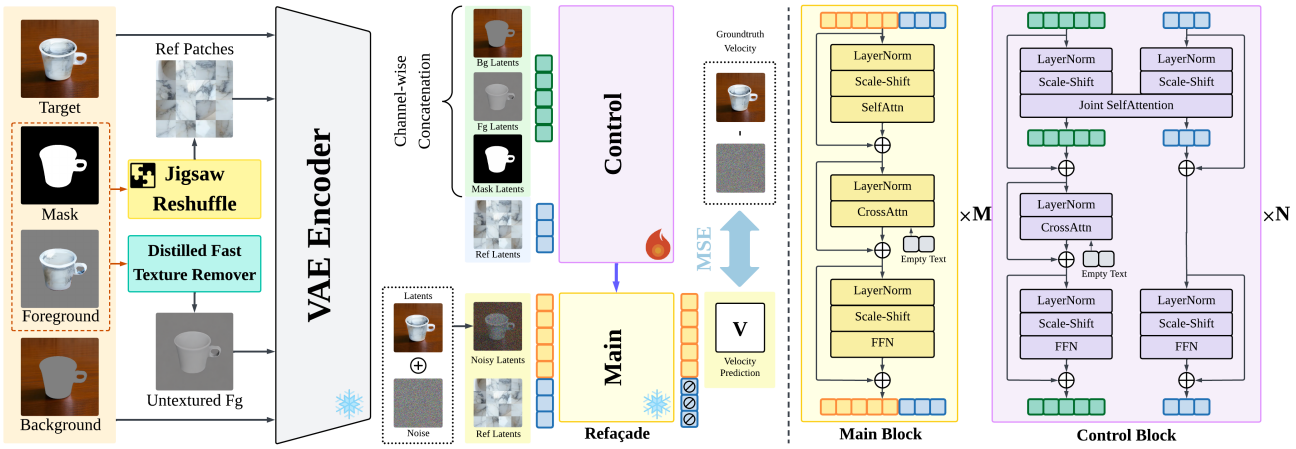

Figure 3. The framework of our **Refaçade**. Left: the training pipeline of **Refaçade**. Right: the model architecture.

texture of the source object and ensures that the retextured results are guided solely by the reference texture. Consequently, **Refaçade** can accurately edit the appearance of the target object according to the reference texture while preserving the surrounding regions unchanged.

Our main contributions are summarized as follows:

- We introduce *object retexture* into video domain, enabling users to edit an object by refering the texture from an image. This task eliminates the need for ambiguous texture prompt when editing an object, allowing users to directly transfer the reference texture onto the source object while preserving the object's original structure.
- We propose **Refaçade**, an unified model for object retexture in image and video. It consists of two strategies to enhance the controllability of texture transfer and reduce the interference of unwanted information. First, we train a generator to convert objects into texture-free representations, replacing the traditional condition extractor. Second, we apply a jigsaw permutation to disrupt the spatial shape of the object in reference image, encouraging the model to focus more on the texture itself.
- We conduct extensive experiments across multiple benchmarks, demonstrating that our method achieves superior performance in object retexturing, producing more precise editing results, higher similarity between the reference and edited textures, and better preservation of the surrounding regions.

2. Methodology

Refaçade employs two key decoupling strategies, as illustrated in Figure 3: **Texture Remover** (Sec.2.1) uses a dedicated diffusion model to remove all texture information from source videos, producing geometry-only representations; and **Jigsaw Permutation** (Sec.2.2) applies an effective permutation strategy to remove structural information from the reference image while preserving its texture.

Given a source video $\mathbf{X}$, its corresponding object mask $\mathbf{M}$, background video $\mathbf{X}^{\text{bg}}$, and reference image $\mathbf{I}^{\text{ref}}$, we first apply the texture remover to obtain an untextured video $\mathbf{X}^{\text{unt}}$, then apply jigsaw permutation to create a structure-agnostic texture guide. Finally, our texture transfer model synthesizes the output by combining geometric structure from the texture-free source with texture patterns from the permuted reference. **Refaçade** is trained with flow matching [43]. Let $\mathbf{z}_0 = \mathcal{E}_{\text{VAE}}(\mathbf{X})$ denote the target latent. The conditioning signal $\mathbf{c}$ comprises multiple components:

$$\mathbf{c} = \left\{ \mathcal{E}_{\text{VAE}}\big(\text{Jigsaw}(\mathbf{I}^{\text{ref}})\big),\ \mathcal{E}_{\text{VAE}}(\mathbf{X}^{\text{unt}}),\ \mathbf{M},\ \mathcal{E}_{\text{VAE}}(\mathbf{X}^{\text{bg}}) \right\},$$

We sample $t \sim \mathcal{U}(0,1)$ and $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with the same shape as $\mathbf{z}_0$, and define the linear interpolation path and target velocity:

$$\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\varepsilon, \qquad \mathbf{v}^\star(\mathbf{z}_t, t) = \varepsilon - \mathbf{z}_0.$$

A velocity network $\mathbf{v}_\theta(\mathbf{z}_t, \mathbf{c}, t)$ is trained with the flow-matching loss [43]:

$$\mathbb{E}_{(\mathbf{z}_0, \mathbf{c})}\ \mathbb{E}_{t\sim\mathcal{U}(0,1),\,\varepsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\Big[\,\|\,\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{z}_t, \mathbf{c}, t)\ -\ \mathbf{v}^\star(\mathbf{z}_t, t)\,\|_2^2\Big].$$

Our framework builds upon VACE [32], with modifications inspired by MM-DiT [18] to better handle distinct conditioning signals. In the control branch, we concatenate background, texture-free video, and mask latents channel-wise and process them through dedicated condition layers, while reference image latents are processed through separate reference layers. This design allows tokens serving different functions (reference vs. source) to use distinct parameters while sharing the same attention mechanism. In the main branch, the reference image is prepended to the first frame of the noisy latent, and hidden states from the control block are added to corresponding layers.

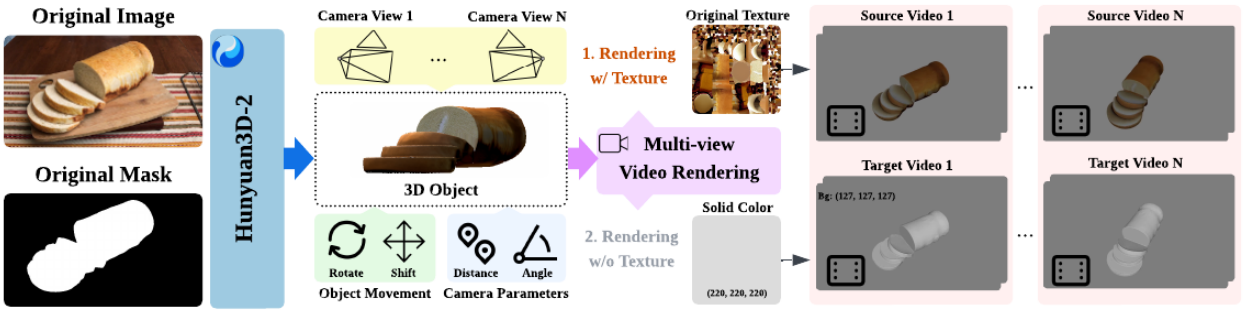

Figure 4. Our data construction pipeline for the texture remover operates as follows: we collect object images, reconstruct 3D meshes, and render paired videos with and without textures under diverse camera trajectories and object motions.

2.1. Texture Remover for Source Image and Video

In 3D mesh representations, object geometry and texture are inherently decoupled: the mesh defines shape through vertices and faces, while appearance is specified separately via texture coordinates and material properties. A naïve solution would be to reconstruct a 3D object mesh from the video and render it in a texture-free manner to obtain geometry-only conditioning signals. However, classical 3D reconstruction from video is computationally expensive, typically requiring several minutes to recover a textured mesh from a single video clip, making it impractical for large-scale training and inference. To obtain geometry supervision efficiently at scale, we train a dedicated diffusion model—the *texture remover*—that learns to map textured video frames directly to texture-free frames of the same object. Specifically, we construct a paired training dataset by rendering 3D objects twice: once with full texture maps applied and once with textures removed using uniform gray materials. We then train a video diffusion model to learn this texture removal mapping directly in 2D space, eliminating the need for explicit 3D reconstruction at inference time. Once trained, this model provides efficient geometry-only control signals for arbitrary video clips while preserving object motion, pose, and shape, ensuring precise temporal and spatial alignment with the source video.

Dataset Construction. The full pipeline is illustrated in Figure 4. Inspired by [57], we begin by collecting a large-scale image dataset containing commonly observed objects from two sources: (1) first frames extracted from real-world videos, and (2) synthetic images generated via text-to-image models using object-centric prompts (e.g., "a chair," "a car"). For each image, we segment the main object using an off-the-shelf segmentation model [55] and reconstruct a textured 3D mesh using Hunyuan3D [85].

For each reconstructed mesh, we generate paired video sequences as follows. First, we render the mesh with full texture maps applied under fixed camera intrinsics and headlight-style point light while varying camera distance and viewing angle over time. This produces a short video clip capturing the object's original textured appearance. Second, we render the same mesh under identical camera and lighting conditions, but with all texture maps and albedo information removed, using a uniform gray Lambertian material. This geometry-only rendering serves as the texture-free target. To increase dataset diversity and improve model robustness, we apply controlled augmentation by varying: (1) camera trajectories (e.g., orbital, arc, zoom-in/out), (2) light intensity, and (3) object poses (random rotations and translations within reasonable bounds).

Training and Distillation. Our texture remover builds on the VACE framework. The input is the source video after background removal, which serves as the control signal. The training objective is to generate the aligned textureless video. We update only the control blocks in VACE while keeping the main branch frozen, thereby restricting learning to the part of the network that translates appearance into geometry. A direct model of this form still requires a large number of denoising steps during sampling, which would significantly increase the total training cost of our full **Refaçade** system. To address this issue, we apply DMD [78] distillation on the trained remover. After distillation, the sampling schedule is reduced from fifty steps to three steps while maintaining the ability to output high-quality texture-free videos.

2.2. Refaçade: Jigsaw Permutation for Structure-Agnostic Texture Transfer

While the texture remover ensures that no source appearance leaks through geometric conditioning, we must also prevent the model from copying the reference image's global layout. A straightforward approach would be to use the first frame of the target video (with background removed) as the reference image during training. However, this strategy introduces a critical problem: the reference image and target video would share identical spatial structure, causing the model to learn spatial alignment rather than texture transfer. During inference, when the reference and

Figure 5. Visualization of Jigsaw Permutation. We extract foreground patches from the reference image on the top-left corner, shuffle and flip them randomly, then rearrange them into a new layout. This destroys global outline while preserving texture patterns of various sizes.

source objects have different shapes or poses, this approach fails catastrophically—the model attempts to transfer structural characteristics rather than appearance patterns.

To bridge this gap between training and inference, similar to [77], we employ a *Jigsaw Permutation* strategy that forces the model to focus on texture rather than object structure. As illustrated in Figure 5, we cut square patches from the foreground area of the reference image. To ensure sufficient reference texture within each patch, we discard any patch containing more than 10% background pixels. We then randomly shuffle and flip these patches, rearranging them into a rectangular area.

Crucially, we resize the crafted reference patches to match the canvas width used during training, but allow the height to vary based on the number of patches. Patch size varies from 16×16 to the maximum inscribed rectangle of the object. This ensures that the reference patches have a different aspect ratio and spatial layout compared to the source object. By training on such spatially-permuted references, the model learns to extract and transfer local texture patterns rather than memorizing global spatial configurations. This facilitates strong generalization: at inference time, the model can successfully transfer textures even when the reference and source objects have vastly different shapes, sizes, or poses.

In the training stage, given a source video or image $\mathbf{X}$, the texture remover module will generate a untextured video or image $\mathbf{X}^{\text{unt}}$. We use jagsaw to permutate the first frame of the source to obtain $\mathbf{I}^{\text{ref}}$. Our final training target is to reconstruct the original video $\mathbf{X}$.

3. Experiments

Training Dataset. We use watermark-free WebVid-10M dataset [5] and the Pexels dataset [53]. Object category names are first extracted using CogVLM2 [27], and the corresponding segmentation masks are generated with Grounded-SAM2 [44, 55]. Only masks with good quality are retained. After filtering, we have approximately 1.8 million videos for WebVid-10M and around 180K for Pexels.

3.1. Implementation Details of Texture Remover

We construct a dataset from 72K distinct object meshes extracted from images with clearly identifiable foreground objects. Each mesh is rendered into short paired video sequences as described in Sec.2.1. Generating approximately eight pairs per object with different augmentation parameters yields 576K paired videos in total—each consisting of a textured source and texture-free target video. Our model is initialized from VACE and trained for two epochs (18K steps, 38 hours) on 32 A800 GPUs with a global batch size of 32, constant learning rate of 1×10^{-5}, gradient checkpointing, and mixed-precision training. We further apply DMD distillation (learning rate 5×10^{-6}, batch size 8, and 300 steps) to produce a fast Texture Remover requiring only three sampling steps at inference.

3.2. Implementation Details of Refaçade

Stage 1: Large-Scale Pretraining. We pretrain the model for two epochs on a mixture of (i) filtered subset of WebVid-10M containing 1.8M videos, (ii) 900k synthetic videos generated by SelfForcing [29], and (iii) 800k synthetic images produced by Stable Diffusion 3.5 Large [60]. The network is initialized from VACE and trained on 96 A800 GPUs with a global batch size of 96 and gradient accumulation of 4, corresponding to 18k training steps over 120 hours. We use a constant learning rate of 1×10^{-5}, enable gradient checkpointing, and train with mixed precision.
Stage 2: High-quality Finetuning. We finetune the model on 180k real videos from Pexels. Finetuning is run for two epochs on 32 A800 GPUs with a global batch size of 32 and gradient accumulation of 4, yielding 2.8k training steps with 28 hours. We keep the same training hyperparameters as in Stage 1, including a constant learning rate of 1×10^{-5}, gradient checkpointing, and mixed-precision training.

3.3. Quantitative Results

We compare **Refaçade** against extensive baselines including specialized inpainting models, general-purpose editing methods, and closed-source commercial APIs. Results are presented in Tables 1 and 2. Baseline implementation details are provided in the supplementary materials.
Benchmark Details Our evaluation benchmark is organized as quadruples, each consisting of a source image/video, a mask, a reference image, and a prompt. For image evaluation, we use the high-resolution image dataset UHRSD [72], which contains 988 images and their corresponding masks. We then employ Flux Kontext to generate reference images with salient objects and randomly pair

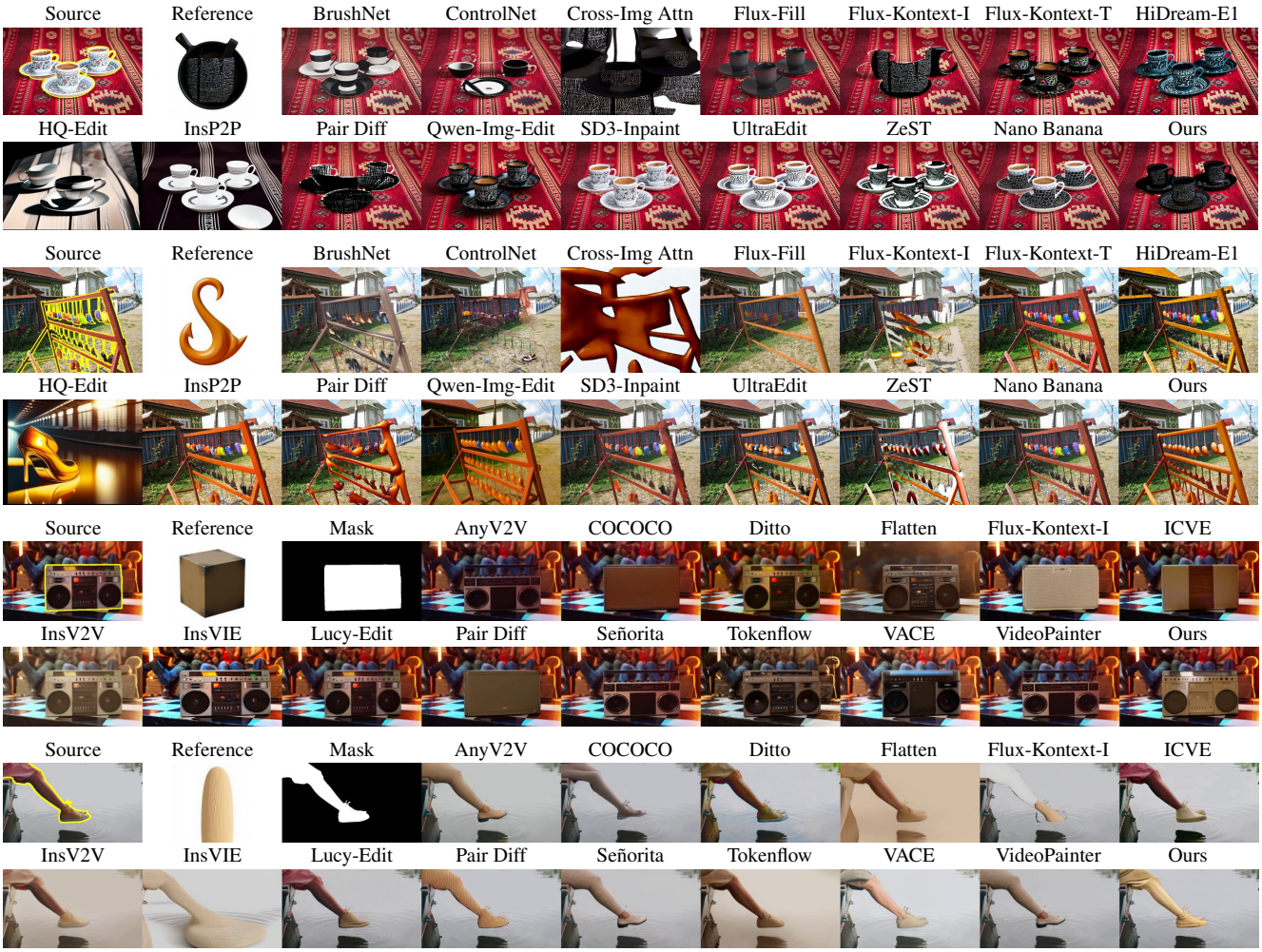

Figure 6. Comparison results of Refaçade and baselines on both images and videos. First 4 rows: images. Bottom 4 rows: videos.

Table 1. Evaluation on image dataset UHRSD. 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. The best results are **boldfaced**, and the second-best results are underlined.

Method	Type	Background				Foreground					LLM Evaluation		User Pref.
		MSE↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	DINO↑	LPIPS↓	Dream↑	GLCM↑	GPT-5↑	Gemini↑	
BrushNet [33]	Inpainting	438.49	23.82	0.7555	0.1758	0.7026	0.2235	0.7162	0.7341	0.8472	2.12	2.12	0.1366
ControlNet-Inp [48]		429.90	27.38	0.7341	0.2386	0.7025	0.1840	0.7281	0.7210	0.8442	1.41	2.12	0.1304
Flux-Fill [7]		67.05	31.92	0.8948	0.0730	0.6900	0.2091	0.7431	0.7134	0.8357	2.71	1.98	0.1615
SD3-Inpaint [18]		65.66	32.35	0.8882	0.0914	0.6617	0.1534	0.7537	0.6821	0.8267	1.29	1.40	0.0311
Pair Diffusion [22]		258.64	27.00	0.7387	0.2431	0.7776	0.4392	0.6369	0.8150	0.8895	2.61	2.66	0.4969
ZeST [15]		198.94	29.55	0.7602	0.2450	0.7620	0.3324	0.7024	0.7971	0.8651	2.64	2.53	0.4658
Cross-img Attn [2]	General	2700.04	15.96	0.6177	0.4870	0.7662	0.3833	0.7223	0.7922	0.8284	2.13	1.96	0.0311
UltraEdit [83]		168.56	30.04	0.7859	0.2216	0.6910	0.1965	0.7373	0.7122	0.8347	2.16	2.16	0.0621
Flux-Kont-I [37]		91.76	31.63	0.8593	0.1321	0.7768	0.4216	0.6607	0.8015	0.8612	1.71	1.65	0.2236
Flux-Kont-T [37]		1038.27	24.16	0.7337	0.2126	0.6770	0.1956	0.7131	0.7025	0.8527	2.37	2.52	0.1553
HiDream-E1 [11]		1187.49	25.55	0.7862	0.2403	0.6866	0.1981	0.7140	0.7170	0.8519	2.41	2.38	0.1491
HQ-Edit [30]		8026.55	9.74	0.4355	0.5654	0.7046	0.2223	0.7267	0.7305	0.8393	1.56	0.90	0.0621
InsP2P [9]		2712.53	16.58	0.6156	0.4087	0.7035	0.2003	0.7166	0.7292	0.8425	1.92	1.73	0.1180
Qwen-I-Edit [69]		1183.89	21.84	0.6868	0.2592	0.6868	0.2196	0.7034	0.7161	0.8576	2.78	2.76	0.1366
NanoBanana [17]		481.66	27.47	0.7547	0.1446	0.6981	0.2582	0.7247	0.7316	0.8391	2.65	2.41	0.1553
Ours(stage1)	Inpainting	49.66	36.19	**0.8994**	**0.0472**	0.7125	0.2665	0.6915	0.7497	0.8830	2.77	**2.81**	0.5714
Ours(stage2)		**49.36**	**36.20**	0.8987	0.0487	**0.7774**	**0.4516**	**0.6181**	**0.8184**	0.9033	**2.89**	2.77	**0.8944**

Table 2. Evaluation results on video dataset Pexels. The LPIPS for background evaluates background perseveration, while LPIPS for foreground evaluates the texture similarity. 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		User Pref.
		MSE↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	DINO↑	LPIPS↓	Dream↑	GLCM↑	EWarp ↓	GPT-5↑	Gemini↑	
COCOCO [89]	Inpainting	164.73	29.09	0.8259	0.1226	0.7125	0.1381	0.7372	0.7286	0.7365	2.1697	1.88	2.18	0.1180
VACE [32]		1596.84	19.73	0.7107	0.2941	0.7159	0.1348	0.8009	0.7240	0.7546	1.7763	1.90	2.40	0.0497
VideoPainter [6]		64.69	32.89	0.9072	0.1052	0.7130	0.1554	0.7377	0.7173	0.7575	1.9965	1.92	2.12	0.0559
AnyV2V [36]	General	498.49	22.77	0.7420	0.1983	0.7178	0.1603	0.7382	0.7253	0.7377	3.5600	2.21	2.18	0.0932
Ditto [3]		2097.47	19.28	0.7144	0.3084	0.6907	0.1229	0.8264	0.6976	0.7409	1.3656	1.20	1.20	0.1366
Flatten [16]		2187.84	15.66	0.6325	0.4308	0.7303	0.1708	0.7731	0.7374	0.6861	1.7492	1.62	1.38	0.0745
TokenFlow [21]		889.93	19.73	0.7107	0.2941	0.7162	0.1502	0.7884	0.7257	0.7931	1.7625	1.69	1.16	0.0683
ICVE [42]		1703.99	19.02	0.7095	0.3098	0.7198	0.1705	0.7766	0.7359	0.7720	1.7486	2.04	1.28	0.1615
InsV2V [14]		3685.70	13.88	0.5556	0.4733	0.7163	0.1389	0.7802	0.7183	0.7418	2.7225	2.00	1.83	0.1429
InsVIE [71]		5450.47	11.94	0.4435	0.5428	0.7172	0.1846	0.8145	0.7448	0.7368	3.3529	2.12	1.70	0.1242
Lucy-Edit [62]		855.43	24.57	0.8204	0.1653	0.6992	0.1463	0.7969	0.7063	0.7463	1.5283	1.84	2.23	0.0683
Señorita [88]		130.53	28.90	0.8634	0.1754	0.6976	0.1503	0.7497	0.7036	0.7444	1.3519	2.10	2.34	0.0621
Pair Diffusion [22]	Image	196.30	27.90	0.7979	0.1318	0.7406	0.2950	0.6176	0.7635	0.8480	3.4220	2.44	2.29	0.0497
Flux-Kont-I [37]		36.54	**35.26**	**0.9460**	**0.0363**	**0.7455**	**0.3209**	**0.7506**	**0.7724**	0.7912	23.5854	**1.69**	1.57	0.0186
Ours(stage1)	Inpainting	30.66	36.44	0.9460	0.0379	0.7331	0.2622	0.6540	0.7473	0.8830	**1.3510**	2.72	**3.27**	0.5155
Ours(stage2)		**30.35**	**36.48**	**0.9485**	**0.0344**	**0.7524**	**0.3241**	**0.6080**	**0.7742**	0.9033	1.4248	**2.82**	3.25	**0.7391**

Table 3. Ablation study for our training pipeline. The LPIPS metric for the background assesses background preservation, whereas the LPIPS metric for the foreground measures the similarity between the reference texture and the generated content. The value of *Ewarp* falls within the range of 1×10^{-3}. The best results are **boldfaced**, and the second-best results are underlined.

Method	Stage	Reference	Structure	Background				Foreground					Motion	LLM Evaluation	
				MSE↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	DINO↑	LPIPS↓	Dream↑	GLCM↑	EWarp ↓	GPT-5↑	Gemini↑
Ab-1	Stage-1	w/o Jigsaw	Canny	68.83	32.62	0.9068	0.0800	0.7022	0.1859	0.7674	0.7046	0.7006	1.5998	2.10	2.36
Ab-2		w/Jigsaw	Canny	30.65	36.47	0.9460	0.0379	0.7149	0.1906	0.7347	0.7205	0.7297	1.4582	2.42	2.76
Ab-3			HED	30.69	36.40	0.9459	0.0379	0.6976	0.1990	0.7484	0.7080	0.7258	**1.2395**	2.44	2.74
Ab-4			Gray	30.70	36.29	0.9458	0.0379	0.7182	0.2115	0.7016	0.7352	0.8049	1.4502	2.66	2.94
Ab-5			Depth	30.73	36.08	0.9458	0.0378	0.6894	0.1790	0.7532	0.7417	0.7608	1.3764	2.21	2.47
Ab-6		w/ Jigsaw	Untextured	30.66	36.44	0.9460	0.0379	0.7331	0.2622	0.6540	0.7473	0.8830	1.3510	2.72	**3.27**
Ab-7	Stage-2	w/ Jigsaw	Untextured	**30.35**	**36.48**	**0.9485**	**0.0344**	**0.7524**	**0.3241**	**0.6080**	**0.7742**	**0.9033**	1.4248	**2.82**	3.25

them with the sources. Qwen2.5-VL 32B [4] takes both the source and the reference image as input to produce captions, from which we derive an instructive prompt and a descriptive prompt that serve as text conditions for some of the methods. For video evaluation, we use 50 videos from Pexels as the test set, which is disjoint from our training data. The reference images are obtained in the same way as for images, using the first frame of each video for captioning.

Automatic Evaluation. We evaluate background preservation using MSE, PSNR, SSIM, and LPIPS, and foreground fidelity using GLCM [25], CLIPScore [26], DINO [51], LPIPS [81], and DreamSim [19]. Video motion consistency is assessed via EWarp [38]. As shown in Table 1, our stage2 model achieves superior background preservation on the image benchmark, substantially outperforming strong baselines such as Flux Fill. Foreground metrics further demonstrate our advantage, with stage2 model attaining the highest CLIPScore (0.7774), DINO (0.4516), and DreamSim (0.8184), alongside the lowest LPIPS (0.6181). On the video benchmark (Table 2), stage2 model again achieves optimal background reconstruction, surpassing VideoPainter. Foreground alignment improves substantially, while temporal stability remains competitive (EWarp: 1.4248 vs. 1.3510 for stage1). Overall, our framework establishes state-of-the-art performance on both image and video texture transfer through high-fidelity background preservation, semantically consistent foreground editing, and strong temporal coherence.

LLM-based Evaluation. To address limitations of automatic metrics in capturing perceptual quality, we employ GPT-5 and Gemini-2.5 for evaluation. LLMs are instructed to evaluate the results along four dimensions: (i) whether the generated texture matches that of the reference image; (ii) whether the generated color is consistent with the reference image; (iii) whether the object structure in the result remains consistent with the source; and (iv) whether the background is preserved as in the source image. Our stage2 model consistently ranks highest: on images, it scores 2.89 with GPT-5 and 2.77 with Gemini-2.5, compared with 2.71 and 1.98 for Flux Fill and 2.65 and 2.41 for NanoBanana; on videos, it achieves 2.82 with GPT-5 and 3.25 with Gemini-2.5, versus 2.21 and 2.18 for AnyV2V.

User Study. To further validate our approach with human judgment, we conduct an extensive user study. We compare the outputs of all competing methods on both images and videos and invite users to evaluate the edited results. Par-

Table 4. Ablation study for patch size in Jigsaw Permutation. The LPIPS metric for the background assesses background preservation, whereas the LPIPS metric for the foreground measures the similarity between the reference texture and the generated content. The value of *Ewarp* falls within the range of 1×10^{-3}. The best results are **boldfaced**, and the second-best results are underlined.

Method	Patch Size	Background				Foreground					Motion	LLM Evaluation	
		MSE↓	PSNR↑	SSIM↑	LPIPS↓	CLIP↑	DINO↑	LPIPS↓	Dream↑	GLCM↑	EWarp ↓	GPT-5↑	Gemini↑
Ab-1	2%	29.87	36.67	0.9485	**0.0326**	0.7184	0.2158	0.7023	0.7210	0.8305	1.5218	2.61	3.08
Ab-2	5%	29.85	36.68	0.9485	**0.0326**	0.7276	0.2495	**0.6387**	**0.7526**	**0.8966**	1.3996	2.76	**3.22**
Ab-3	10%	29.86	36.70	0.9485	**0.0326**	0.7305	**0.2615**	0.6504	0.7410	**0.8996**	1.4495	2.76	3.10
Ab-4	20%	29.84	36.69	0.9485	**0.0326**	**0.7344**	0.2500	0.6554	0.7397	0.8860	1.4603	**2.78**	3.08
Ab-5	50%	**29.83**	36.70	0.9485	**0.0326**	0.7232	0.2345	0.6586	0.7316	0.8708	1.4352	2.60	3.10
Ab-6	100%	**29.83**	**36.71**	**0.9486**	**0.0326**	0.7247	0.2380	0.6503	0.7357	0.8852	**1.3970**	**2.78**	3.10

ticipants are shown the source image/video, the reference image, and the outputs of different methods. They are then asked to evaluate the outputs along three dimensions: (i) whether the reference material is successfully transferred to the selected object; (ii) whether the background is preserved; and (iii) whether the object's structure is maintained. The user preferences on the image and video benchmarks are reported in Tables 1 and 2, respectively. Our method consistently receives the highest number of votes for both images and videos.

3.4. Qualitative Results

Visual comparisons in Figure 6 demonstrate superior background preservation, texture coherence, and foreground fidelity, validating the effectiveness of our framework in terms of perceptual quality. Our method excels in three key aspects: (i) it edits the entire object, unlike HiDream-E1 and NanoBanana; (ii) it precisely preserves the background, outperforming Qwen-Image-Edit and InsVIE; and (iii) it achieves better texture consistency with the reference image during retexturing.

3.5. Ablation Study

To validate our design choices, we conduct ablation studies on structural conditioning, jigsaw augmentation, patch size, and two-stage training, as shown in Tables 3 and 4.

Impact of Structural Conditions. Table 3 compares different structural conditioning signals (Ab-2 to Ab-6). Although all variants exhibit comparable background preservation (similar MSE and SSIM), their foreground quality diverges notably. Conventional signals such as Canny, HED, grayscale, and depth produce weaker texture transfer, whereas our untextured video conditioning (Ab-6) consistently achieves higher semantic alignment and lower perceptual distortion, as reflected by CLIP score, LPIPS, DINO score, and LLM-based scores. This indicates that traditional structural cues are prone to texture leakage, where residual appearance information contaminates the conditioning and ultimately degrades texture transfer.

Impact of Jigsaw Permutation. Table 3 compares Ab-1 (without jigsaw) and Ab-2 (with jigsaw). Without jigsaw augmentation, performance degrades across all metrics: background MSE increases from 30.65 to 68.83, PSNR drops from 36.47 to 32.62, and foreground LPIPS worsens from 0.7347 to 0.7674. LLM scores also decline (GPT-5: 2.10 vs. 2.42). This demonstrates that jigsaw augmentation is essential for preventing *geometry leakage*, where the reference image's structure contaminates the output.

Impact of Patch Size in Jigsaw Permutation. Table 4 examines the effect of patch size, where the value is expressed as a percentage of the reference image side length (i.e., the side length of each square patch divided by the side length of the full reference frame). In particular, the 100% setting degenerates to using the original reference image without any jigsaw permutation. Across all settings, background preservation remains similar, while foreground quality varies. Small patches (2%) yield weaker alignment, medium patches (5–10%) achieve a better texture transfer, whereas large patches (50–100%) provide the best temporal stability at the cost of slightly reduced texture fidelity.

Impact of Two-Stage Training. As shown in Table 3, stage two Ab-7 improves upon stage one Ab-6 across all metrics. Background LPIPS decreases from 0.0379 to 0.0344, foreground DINO score increases from 0.2622 to 0.3241, and LPIPS improves from 0.6540 to 0.6080. LLM scores under GPT-5 also rise from 2.72 to 2.82, confirming that the second stage effectively refines texture transfer quality.

4. Conclusion

In this paper, we introduce **Refaçade** for a new editing task, object retexture. Our method is designed to enhance controllability and suppress unwanted information during texture transfer. It comprises two key components. First, we replace traditional control conditions with texture-free representations rendered from 3D object meshes, which preserve the structural information of the original object while excluding color and texture cues. Second, we introduce a jigsaw permutation strategy that disrupts spatial structure in the reference image, forcing the model to attend to texture statistics rather than object layout. Extensive experiments demonstrate that our approach can accurately transfer the target texture onto source objects while preserving their structure, and produces visually compelling results.

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. 6

[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. 7

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 7

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021. 5

[6] 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, 7

[7] Black Forest Labs. Black forest labs. https://github.com/black-forest-labs/flux/, 2024. 1, 6

[8] 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

[9] 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. 1, 6

[10] 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 preprint arXiv:2505.22705*, 2025. 1

[11] 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. 6

[12] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024. 1

[13] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-$alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 1

[14] Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. *arXiv preprint arXiv:2311.00213*, 2023. 7

[15] 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. 6

[16] 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. 1, 7

[17] Google DeepMind. Nano banana - gemini ai image generator & photo editor. https://gemini.google/overview/image-generation/, 2025. 6

[18] 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. 3, 6

[19] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023. 7

[20] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Houqiang Li, Han Hu, et al. Instructdiffusion: A generalist modeling interface for vision tasks. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 12709–12720, 2024. 1

[21] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 1, 7

[22] 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. 6, 7

[23] 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

[24] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 1

[25] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, SMC-3(6): 610–621, 2007. 7

[26] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 7514–7528, 2021. 7

[27] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024. 5

[28] 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

[29] 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. 5

[30] 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. 1, 6

[31] 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

[32] 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. 3, 7

[33] 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. 6

[34] Xuan Ju, Tianyu Wang, Yuqian Zhou, He Zhang, Qing Liu, Nanxuan Zhao, Zhifei Zhang, Yijun Li, Yuanhao Cai, Shaoteng Liu, et al. Editverse: Unifying image and video editing and generation with in-context learning. *arXiv preprint arXiv:2509.20360*, 2025. 1

[35] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 1

[36] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. Anyv2v: A plug-and-play framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2(3): 5, 2024. 1, 7

[37] 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. 6, 7

[38] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. 7

[39] 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

[40] 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.

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

[42] 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. 7

[43] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3

[44] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 5

[45] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8599–8608, 2024. 1

[46] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. 1

[47] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025. 1

[48] mikonvergence. Controlnetinpaint: Inpaint images with controlnet. https://github.com/mikonvergence/ControlNetInpaint, 2023. 6

[49] Mochi-1. Mochi-1. https://www.genmo.ai/blog, 2024. 1

[50] OpenAI. Sora: Creating video from text. https://openai.com/index/sora/, 2024. 1

[51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7

[52] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 1

[53] Pexels. https://www.pexels.com/, 2024. 5

[54] 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

[55] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4, 5

[56] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[57] Penghui Ruan, Bojia Zi, Xianbiao Qi, Youze Huang, Rong Xiao, Pichao Wang, Jiannong Cao, and Yuhui Shi. Ctrl&shift: High-quality geometry-aware object manipulation in visual generation. *arXiv preprint arXiv:2602.11440*, 2026. 4

[58] GA RunwayML. Introducing gen-3 alpha: a new frontier for video generation, 2024. 1

[59] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 1

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

[61] 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

[62] DecartAI Team. Lucy edit: Open-weight text-guided video editing, 2025. Accessed: 2025-11-13. 7

[63] Kolors Team. Kolors: Effective training of diffusion model for photorealistic text-to-image synthesis. *arXiv preprint*, 2024. 1

[64] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1, 2

[65] 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

[66] 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

[67] Zhouxia Wang, Xintao Wang, Liangbin Xie, Zhongang Qi, Ying Shan, Wenping Wang, and Ping Luo. Styleadapter: A unified stylized image generation model. *arXiv preprint arXiv:2309.01770*, 2023. 1

[68] 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.

[69] 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. 1, 6

[70] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7623–7633, 2023. 1

[71] 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. 7

[72] Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11717–11726, 2022. 5

[73] 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

[74] 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.

[75] 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

[76] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 1

[77] Yuteng Ye, Zheng Zhang, Qinchuan Zhang, Di Wang, Youjia Zhang, Wenxiao Zhang, Wei Yang, and Yuan Liu. Jigsaw3d: Disentangled 3d style transfer via patch shuffling and masking. *arXiv preprint arXiv:2510.10497*, 2025. 5

[78] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image syn-

thesis. *Advances in neural information processing systems*, 37:47455–47487, 2024. 2, 4

[79] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023. 1

[80] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 2

[81] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[82] 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

[83] 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. 1, 6

[84] 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

[85] 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. 4

[86] 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

[87] 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.

[88] 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. 2, 7

[89] 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. 2, 7