Beyond the Visible: Disocclusion-Aware Editing via Proxy Dynamic Graphs

Anran Qi[1] Changjian Li[2] Adrien Bousseau[1] Niloy J.Mitra[3,4]

[1]Inria - Université Côte d'Azur [2]University of Edinburgh [3]Adobe Research [4]UCL

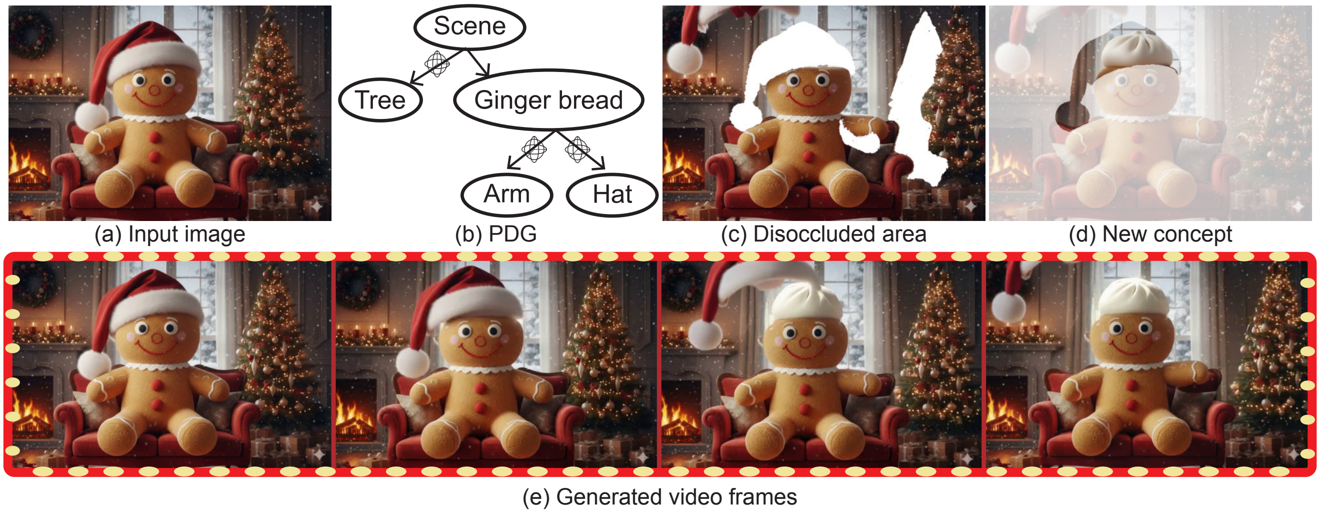


Figure 1. Given a single input image (a), the user (i) creates a lightweight *PDG* (b) to indicate *how* parts should move and (ii) specifies *what* should appear in the final frame's disoccluded areas. The result (e) is a plausible image→video with explicit articulation control (c) and user-chosen reveals (d), surpassing the latest text and/or drag/flow-based alternatives.

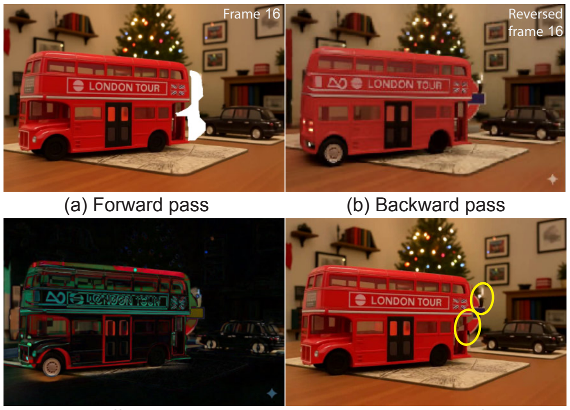
Abstract

We address image-to-video generation with explicit user control over the final frame's disoccluded regions. Current image-to-video pipelines produce plausible motion but struggle to generate predictable, articulated motions while enforcing user-specified content in newly revealed areas. Our key idea is to separate motion specification from appearance synthesis: we introduce a lightweight, user-editable Proxy Dynamic Graph (PDG) that deterministically yet approximately drives part motion, while a frozen diffusion prior is used to synthesize plausible appearance that follows that motion. In our training-free pipeline, the user loosely annotates and reposes a PDG, from which we compute a dense motion flow to leverage diffusion as a motion-guided shader. We then let the user edit appearance in the disoccluded areas of the image, and exploit the visibility information encoded by the PDG to perform a latent-space composite that reconciles motion with user intent in these areas. This design yields controllable articulation and user control over disocclusions without fine-tuning. We demonstrate clear advantages against state-of-the-art alternatives towards images turned into short videos of articulated objects, furniture, vehicles, and deformables. Our method mixes generative control, in the form of loose pose and structure, with predictable controls, in the form of appearance specification in the final frame in the disoccluded regions, unlocking a new image-to-video workflow. Code will be released on acceptance.

1. Introduction

Imagine turning a single image into a short video where *you* decide both *how* things move and *what* appears when parts move aside. Today's generative models [16, 25, 59] get you plausibility, but not precision: beyond coarse text prompts, they either ask for tedious arrows/flows [28, 43] or demand frustrating trial-and-error to coax a specific motion; controlling the *appearance in disoccluded regions* is even harder, and rarely supported. We aim to achieve *both* – predictable articulation from coarse specifications and user-chosen reveals – while preserving the realism we now expect from modern video generators [12, 13, 24, 59].

Contrary to current practice, we argue that high-quality image-to-video generation should be both generative as well as predictable. Users should be able to specify *what*



(a) Forward pass (b) Backward pass

(c) Difference map (d) Pixel-level copy&paste

Figure 2. **Motivation: forward vs. backward inconsistency in disocclusions.** (a) Forward pass with DaS [16] exposes a large disoccluded region (white mask); (b) Backward pass from the last frame produces a plausible but *different* reveal resulting in a (c) difference map highlights misalignments concentrated on newly visible areas (rear of the bus, background), showing that forward/backward visibility disagree. Hence, (d) Naïve pixel copy–paste between forward/backward passes creates seams/ghosting (yellow circles) due to parallax, shading, and occlusion-order mismatches. This motivates us to solve the disocclusion problem.

moves and *how*, while the generator is invoked *only where necessary*, primarily in disoccluded regions, preserving the identity and geometry of the input image elsewhere. Existing approaches fall short because they (i) do not expose an explicit, user-editable, part-level scene representation, and/or (ii) entangle motion specification with appearance synthesis, leaving regions revealed under disocclusions to chance. Even with auxiliary signals (text, depth, or image-space flow), users lack a principled mechanism to drive coherent part motion or to enforce a desired final appearance in newly revealed (disoccluded) areas. Providing multiple images as input, i.e., keyframes, would be an option but it raises the challenge of aligning and fusing multiple sources of visual content (see Fig. 2).

We introduce a training-free pipeline built around a lightweight *Proxy Dynamic Graph (PDG)*, an abstracted scene representation that users annotate directly over the input image (see Fig. 1(b)). Nodes represent rigid or semi-rigid primitives (e.g., articulated object parts), while edges encode relative motions with limited degrees of freedom (e.g., translation, rotation). We utilize off-the-shelf image segmentation and monocular depth estimation to help users quickly create this representation. Once annotated, users can easily repose the graph to prescribe a target pose, effectively addressing the challenge of precise motion control.

From the user-provided PDGs, we internally derive a target motion flow that encodes dense correspondence across frames. We then feed this proxy-guided flow into a pretrained image-to-video diffusion model, which acts as a motion-conditioned shader [16]: it harmonizes warped content and inpaints within the disocclusion areas. However, the generated inpainting may not still align with user intents, especially in the newly revealed parts. We let users overwrite the disoccluded content by editing the final frame of the generated video with their tool of choice. We then perform a novel *training-free update* by suitably mixing features from the first and the target last frame and rerun the forward pass to obtain the final video – this reconciles the proxy-guided motion with the final-frame correction in disoccluded areas. We perform this update in the latent space of the image-to-video model rather than in pixel space, as this strategy brings robustness to misalignment and benefits from diffusion priors to synthesize realistic secondary effects like shadows and moving highlights. See Figure 1.

Thus, we address the two key weaknesses (discussed above) in current generative image editing setups. (i) Users can loosely specify motion by our PDG, providing direct, interpretable control: moving a part predictably moves attached parts; modifying limits or hierarchy deterministically changes the global flow. (ii) Users have control over appearance in disoccluded regions as synthesis is mainly restricted by video priors, preserving identity and preventing global drift while still ‘reaching’ the target end frame. Note that as our method is training-free, it can only synthesize video clips that are reachable in its internal latent manifold.

We validate the proposed method on images to produce short clips of articulated objects, furniture, vehicles, and deformables. Compared to text/point/box-guided diffusion, flow-only warping, and training-based edit models, ours achieves higher pose and structure fidelity to the user target, lower run-to-run variance, and better identity preservation on unedited regions. A user perceptual study demonstrates the superior performance of our method over baselines, and additional ablation studies and discussions further validate the effectiveness of our technical design choices.

In summary, our contributions are (i) a training-free, controllable edit pipeline driven by a user-annotated PDG; (ii) motion-guided diffusion that is limited to disocclusions and closely aligned to a user-specified final frame, and (iii) a user-guided tool that seamlessly marries generative results with predictable workflows.

2. Related Works

Image/video generator–based editing. For *image* editing, state-of-the-art practice is dominated by large diffusion generators, augmented with lightweight controllers and personalization. Commercial services (*e.g.* Photoshop Firefly, Midjourney, DALL·E 3) and open-source backbones (*e.g.* Stable Diffusion/SDXL, widely deployed via ComfyUI/InvokeAI/FluxKontent) deliver high-fidelity edits through prompt conditioning and localized constraints [1, 3, 11, 14, 22, 33, 36, 48, 49]. Controllability is typically provided by conditioning modules and adapters (ControlNet, T2I-Adapter, IP-Adapter, InstantID, BrushNet [23, 35, 53, 60, 62]), identity- and concept-preserving personalization (DreamBooth, TextualInversion, LoRA/Custom Diffusion [15, 20, 26, 42], and localized guidance in the denoising process (Prompt-to-Prompt, Null-text Inversion, SDEdit, Blended Diffusion, MasaCtrl [4, 9, 17, 31, 34]), with point/drag interfaces (DragGAN [38], DragDiffusion [45], EasyDrag [19], DiffusionHandles [39]) for spatial manipulation. Recent compact or task-tailored variants (*e.g*. NanoBanana) target faster inference or more robust localized control [13]. In *video*, pretrained priors are increasingly used to maintain temporal coherence during edits (*e.g*. lift-edit-project with a 3D proxy; sequence-to-sequence movement with a video diffusion backbone), highlighting a trade-off between zero-shot generality and task-specific training [10, 25, 61].

Control modalities: from prompts to parts. The usability of editing systems rests on the *type of control* exposed to users. *Text/instruction* editors modulate internal attention or features for zero-shot edits but generally lack precise composition/pose control [7, 8]. *Drag-/point-based* interfaces enable local geometric manipulation with identity-aware constraints but offer limited understanding of global articulation; recent variants, such as DragAPart, push toward part-level motion [5, 19, 27, 44–46]. *Geometry-driven* controls condition on depth/flow/feature warps improving correspondences while still lacking an editable articulation structure for coherent part motion [6]. Finally, *3D-/pose-aware* controls steer perspective and articulation but often require reconstruction, optimization, or training [32, 39, 43]. Video-centric works like *VideoHandles* expose object transforms via a reconstructed proxy while using a video prior to propagate edits [25] or using particle-based physics priors [51]. Across these families, two gaps persist: first, the absence of an explicit, user-editable *part–joint* structure for specifying motion; and second, limited user control over *disoccluded content*.

Training-free editing. Training-free pipelines compose off-the-shelf segmentation, optical flow, monocular depth, and frozen diffusion/video priors with inference-time constraints. Traditional composition workflows (remove first, then insert) are brittle: identity drift is common, and secondary effects around shadows and reflections outside the inpaint mask are poorly synchronized (see pixel baseline in Section 4); enlarging masks harms identity [21, 56]. Zero-shot image/video editors perform latent lift-edit-project operations to obtain temporally consistent edits without fine-tuning [25, 39]. Drag-style methods reduce setup cost and provide intuitive local controls, but motion remains local and newly revealed regions are left to stochastic inpainting, limiting user authority over disocclusions [5]. Our approach also falls into this training-free family, while introducing both a user-annotated *proxy articulation graph* to deterministically, yet loosely, dictate part movement and explicitly enabling predictive control in the disocculded regions via the last frame. Note that the disoccluded region in our editing workflow is dependent on the user-authored movements as indicated by the PDG articulations.

Training-based editing. Task-specific fine-tuning yields strong realism and synchronization but demands sizable data pipelines and reduces generality. *ObjectMover* [61] builds a synthetic corpus and adds auxiliary real-video tasks to fine-tune a video diffusion transformer for single-stage movement/removal/insertion with synchronized lighting/shadows. *3D-Fixup* [2, 10] trains a feed-forward editor from video-mined supervision and image-to-3D priors, handling large viewpoint changes yet inheriting reconstruction/mask limitations. Still, none offers appearance control in disoccluded regions. *Boximator* constrains motion via trainable box controls but lacks an explicit part–joint structure [52]; *MagicStick* transforms internal handles (T2I+ControlNet with LoRA/inversion) for convincing edits but needs adaptation and no final-frame disocclusion control [30]; *Puppet-Master* uses sparse drags with a learned motion prior for point-handle articulation, while relying on stochastic synthesis for newly revealed content [29]; see Section 4 for comparison.

3. Method

Fig. 3 displays an overview of our approach. Given an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, our goal is to generate an editing video $\mathbf{V} \in \mathbb{R}^{(1+T)\times H\times W\times 3}$ (the first dimension is the number of frames), guided by a text prompt $\mathbf{T}$ and minimal user interaction. The user interaction is grounded in several advanced vision models and translated into an abstract scene graph (PDG, Sec. 3.1), which provides explicit and predictable control over objects and their parts (Sec. 3.2). Beyond dynamic object control, our approach also enables users to edit the disoccluded regions revealed by motion. The editing intention – including object manipulation and disocclusion region modification – is ultimately transformed into a coherent editing video using a pre-trained image-to-video diffusion model (Sec. 3.3).

Before detailing the technical details, we first introduce the concept of Proxy Dynamic Graph (PDG) and provide background on the image-to-video diffusion model DaS [16] that serves as our backbone.

Proxy Dynamic Graph. The PDG is a directed acyclic graph that encodes the geometry and motion of moving objects in the image. Nodes in the graph can represent entire objects or individual object parts, modeled as 3D point clouds. A directed edge between a parent node and a child

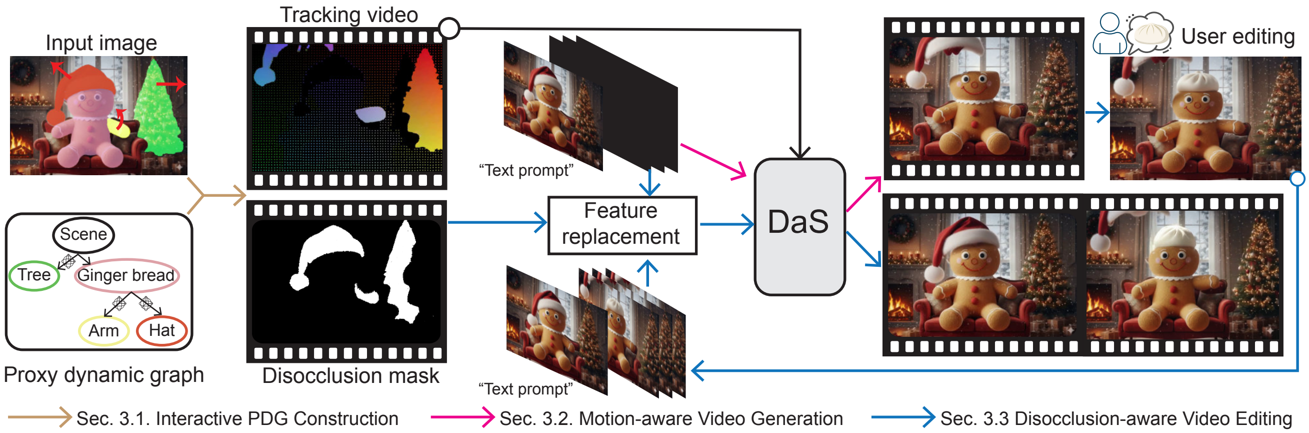


Figure 3. **Overview.** From an input image, we build a *Proxy Dynamic Graph (PDG)* and obtain coarse part tracks (marked with red arrows) and a disocclusion mask. **Pass I, top:** we run DaS (Diffusion-as-shader) to generate a motion-aware video driven by the PDG. The user then edits the *final frame* to prescribe the desired reveal in disoccluded regions (top-right). **Pass II, bottom:** without any retraining, we surgically replace the corresponding feature channels with the edited final-frame features and rerun DaS, yielding a video that preserves PDG-driven motion while matching the user-specified reveal (bottom-right). See supplemental additional detail and videos.

node encodes how the child moves with respect to the parent. Our implementation supports translational and rotational motions. A motion is parameterized by its center, axis, and range of movement. When a node undergoes a spatial transformation, all its descendant nodes are updated following forward kinematics. Figure 1 illustrates a typical PDG, where the Christmas tree can translate with respect to the static scene, while the arm of the gingerbread can rotate and the hat can translate with respect to its body.

DaS video generation. DaS [16] is an image-to-video latent diffusion model. It takes as input an image, a text prompt, and a 3D tracking video, and denoises a random noise into a realistic video expressing the desired motion. Since it is a latent diffusion model, both the image and the tracking video ($\mathbf{V}_{tr} \in \mathbb{R}^{(1+T)\times H\times W\times 3}$) are first converted into latent features using a frozen VAE encoder $\mathcal{E}$: $\mathcal{F}_{tr} = \mathcal{E}(\mathbf{V}_{tr}) \in \mathbb{R}^{(1+\frac{T}{4})\times\frac{H}{8}\times\frac{W}{8}\times 16}$ (see their Fig. 2). Specifically, the input image is first preprocessed with zero padding to obtain the pseudo video $\mathbf{V}_s = \mathbb{R}^{(1+T)\times H\times W\times 3}$, which is further converted into a latent feature:

$$\mathcal{F}_s = \mathcal{E}(\mathbf{V}_s) \in \mathbb{R}^{(1+\frac{T}{4})\times\frac{H}{8}\times\frac{W}{8}\times 16}. \quad (1)$$

The tracking video encodes the precise motion of the object of interest, while the input image provides the appearance reference for the diffusion model.

In the denoising process, a denoising Diffusion Transformer (DiT) ϵ_θ iteratively denoises an initial random noise z_N into a clean latent z_0, conditioned on the text prompt and the encoded image and tracking video. The denoised latent is finally decoded into a resulting video $\mathbf{V} = \mathcal{D}(z_0)$. The denoising process at step n $(0 < n \leq N)$ is defined as:

$$\epsilon_n = \epsilon_\theta(z_n, n, \mathbf{T}, \mathcal{F}_s, \mathcal{F}_{tr}), \quad z_{n-1} = z_n - \epsilon_n. \quad (2)$$

3.1. Interactive PDG Construction

To enable precise control over objects/parts in the image, a structural and dynamic analysis is essential. For example, part segmentation and reconstruction of a lamp, as well as its part articulation, reveal the full degree of freedom to move its parts. To this end, existing research either relies on professional software to manually model the 3D scene and move its objects or object parts [16], or on drag-based interaction to achieve in-plane and limited 3D transformations [29] (*e.g.*, dragging a mug towards the right side of a table, or dragging an animal nose to rotate its head). The former approach requires expertise and is labor-intensive, while the latter is easy to perform but inaccurate without accessing the 3D information. Our PDG offers a sweet spot between these two extremes. On the one hand, the PDG models the coarse 3D geometry and motion parameters for accurate object/part movement. On the other hand, we leverage computer vision algorithms to greatly ease the construction of the PDG compared to full 3D modeling.

Given an input image, we first employ MoGe [54] to simultaneously estimate the depth map and the camera parameters (extrinsic and intrinsic). Next, we exploit SAM2 [41] to segment the objects or object parts. This step involves little interaction in the form of 2D bounding boxes that users place around each part they wish to control. The resulting segments, along with the camera parameters, allow us to lift the depth map into a point cloud for each part, which form the nodes of the PDG. Users can then assign motion to nodes by specifying parent-child relationships and motion parameters (*i.e.*, the motion type, center, axis, and the range of the motion), making the whole graph ready to be manipulated.

3.2. Motion-aware Video Generation

Having the graph, users are free to choose any child node and re-pose the associated 3D point cloud based on their editing intention. The manipulation is propagated to subsequent child nodes in a forward kinematics manner. Note that inverse kinematics could also be exploited to let users only transform the end nodes (we leave it for future work). We keep track of the user manipulation by recording the change of the moved objects/parts, which gives us transformed point clouds as well as a disocclusion mask $\mathbf{M} \in \mathbb{R}^{(1+T)\times H\times W\times 1}$. The disocclusion mask is an evolving binary mask along time, where the region revealed by the dynamics is set to one, while the other regions are set to zero, depending on the instantaneous position of the movable objects/parts. The transformed point cloud is further translated into a tracking video serving as the input for DaS.

Given the input image $\mathbf{I}$, the tracking video $\mathbf{V}_{tr}$, and the text prompt $\mathbf{T}$, we execute the forward video generation process with DaS to produce the resulting motion video $\mathbf{V}_m \in \mathbb{R}^{(1+T)\times H\times W\times 3}$, adhering to the user interaction with movable objects/parts. We next explain how we use the disocclusion mask to provide additional control on the revealed regions.

3.3. Disocclusion-aware Video Editing

The motion of the objects/parts inevitably reveals disoccluded regions, which DaS inpaints based on its diffusion priors. While the result is often realistic, users have no explicit control on it. We reinstate control over these regions in the following way.

Firstly, the last frame of the motion video is extracted. Based on the disocclusion mask in the last frame, users can inpaint the disoccluded area with the desired appearance (*e.g.*, a bun or the fire in Figs. 1 and 5), forming a new target last frame $\mathbf{I}_{edit}$. Users can rely on any advanced image editing tool, such as Adobe Photoshop Generative Fill, to perform this correction. We additionally describe the user newly inpainted concept or object in the last frame with a text prompt $\mathbf{T}_{new}$.

Next, we fuse the input image and the target last frame in the latent space during DaS's denoising process to produce the final motion- and disocclusion-aware video. Specifically, we first concatenate T copies of the edited last frame $\mathbf{I}_{edit}$ to obtain an initial edited video, which we encode as a latent feature $\mathcal{F}_{edit} \in \mathbb{R}^{(1+\frac{T}{4})\times\frac{H}{8}\times\frac{W}{8}\times 16}$ using the VAE encoder $\mathcal{E}$. We then downsample the disocclusion mask to $\mathbf{M}' \in \mathbb{R}^{(1+\frac{T}{4})\times\frac{H}{8}\times\frac{W}{8}\times 1}$ and use it to composite the input image encoding $\mathcal{F}_s$ with the edited last frame encoding $\mathcal{F}_{edit}$ to produce $\mathcal{F}_{compose} = \mathbf{M}'\mathcal{F}_{edit} + (1-\mathbf{M}')\mathcal{F}_s$. Finally we replace $\mathcal{F}_s$ with $\mathcal{F}_{compose}$ in the first M steps of the iterative denoising process, which effectively injects the appearance cues of the edited disoccluded regions:

$$\epsilon_n = \begin{cases} \epsilon_\theta(z_n, n, \mathbf{T}, \mathcal{F}_{compose}, \mathcal{F}_{tr}) & n > (N-M) \\ \epsilon_\theta(z_n, n, \mathbf{T}, \mathcal{F}_s, \mathcal{F}_{tr}) & n \leq (N-M) \end{cases}.$$

We empirically set $M = 35 < N = 50$, which we found to be a good trade-off between adhering to the user-provided content while leaving the diffusion model room for merging and harmonizing the edit throughout the video. The final editing video $\mathbf{V}$ is obtained by decoding z_0. Note that in this generation pass, the text prompt is a concatenation of $\mathbf{T}$ and $\mathbf{T}_{new}$.

4. Experiments and Results

Dataset. Since we introduce the new task of controllable, disocclusion-aware image-to-video generation and found no public benchmark, we created a benchmark test set.

We curated ten indoor images from the web or AI tools (*e.g.*, NanoBanana) and paired each with a short descriptive prompt. Two images (a desk lamp and a toaster) have richer articulation on simple backgrounds; we use them *only* to evaluate part manipulation. The other eight images have simpler motions but harder disocclusions; we use them to evaluate both manipulation and revealed-content synthesis. For every image, we constructed an interactive PDG with our tool. For the lamp and toaster, we defined five distinct, plausible manipulations each, yielding $2 \times 5 = 10$ samples. We automatically extracted tracking videos from these manipulations and fed them to DaS for video generation; because disocclusions are minimal, we did not edit the last frames. For each of the other eight images, we defined one manipulation, ran DaS to generate the video, and then inpainted the final-frame disoccluded region(s) with five user-specified variations, producing $8 \times 5 = 40$ samples.

In total, we created 50 samples: 10 manipulation-only samples (lamp/toaster) with the input image, prompt, and tracking video, and 40 reveal-focused samples that also include an edited last frame. We do not provide ground-truth videos; this benchmark is designed to test controllability, final-frame authority over disocclusions, and overall video quality under realistic authoring conditions.

Metrics. We employ several computational metrics to evaluate our approach on three aspects. (a) *Motion accuracy.* We extract the optical flow from every pair of consecutive frames in both the generated video and our tracking video, and compute the mean cosine similarity between the two sets of flow vectors (denoted as OptFlow). (b) *Last-frame similarity.* We propose two new metrics to assess the consistency between the user-edited last frame and the corresponding last frame in the generated video. Specifically, we measure their pixel-wise Euclidean distance in the RGB space (denoted as Idiff). To focus on the edited areas, we further apply the disocclusion mask to restrict the

Table 1. **Quantitative results for image→video.** Two settings: *Manipulation* (top) and *Video Editing* (bottom). Higher is better for OptFlow/SSIM/PSNR/CLIP-S; lower is better for Idiff/Idiff$_m$/FID/FVDS/FVDC/LPIPS. Best and second-best are **bold** and underlined. Ours attains the strongest motion accuracy and overall video quality.

	Motion Accuracy	Last-frame Similarity			Video Quality					
	OptFlow (↑)	Idiff (↓)	Idiff$_m$ (↓)	FID (↓)	FVDS (↓)	FVDC (↓)	SSIM (↑)	PSNR (↑)	LPIPS (↓)	CLIP-S (↑)
Manipulation										
Veo3+$\mathbf{I}$+$\mathbf{T}_m$	0.02	-	-	-	1597.32	1600.05	0.63	9.89	0.52	**0.24**
Ours	**0.72**	-	-	-	**1107.99**	**1108.91**	**0.81**	**17.58**	**0.29**	0.22
Video Editing										
Pixel-CP	**0.65**	**8.97**	**0.05**	**12.15**	1719.87	1724.77	**0.72**	16.80	**0.31**	0.21
Pixel-CP++	0.64	9.51	1.48	17.86	1730.73	1735.34	0.71	16.70	**0.31**	**0.22**
DaS+$\mathbf{T}_{new}$	0.64	24.38	14.4	72.8	1660.42	1664.84	**0.72**	**16.88**	**0.31**	0.21
Veo3+$\mathbf{I}$+$\mathbf{T}_m$+$\mathbf{I}_{edit}$	0.10	52.55	12.22	75.61	1645.04	1650.12	0.43	12.28	0.55	**0.22**
Ours	**0.65**	23.82	6.91	57.14	**1639.47**	**1643.88**	0.71	16.63	0.32	**0.22**

computation to relevant regions, resulting in a masked variant (denoted as Idiff$_m$). Additionally, we use the FID metric [18] as a complementary measure of perceptual realism. (c) *Video quality.* For overall quality, we employ FVDS [47, 50], FVDC [50, 58], SSIM, PSNR[55], LPIPS [63], and CLIP-space Similarity [40] (denoted as CLIP-S). Intuitively, SSIM and PSNR measure pixel-level fidelity, LPIPS captures perceptual similarity, and FVDS and FVDC evaluate temporal and distributional realism. For metrics that need a ground truth video, we exploit CogVideoX [59] to produce pseudo ground truth videos by taking as input the text prompt concat($\mathbf{T}$, $\mathbf{T}_{new}$, $\mathbf{T}_m$) and the input image $\mathbf{I}$. Here, $\mathbf{T}_m$ describes the manipulation intention (Sec. 3.2). For example, in Figure 3, $\mathbf{T}$ is "The hat is flying...", $\mathbf{T}_{new}$ is "There is a bun on the head of gingerbread", while $\mathbf{T}_m$ is "The hat is moving upward and diagonally towards the upper left corner ...". See supplementary for more examples.

4.1. PDG-based Manipulation

Competitors. To evaluate the effectiveness of our PDG-based manipulation, we compare against two baselines:

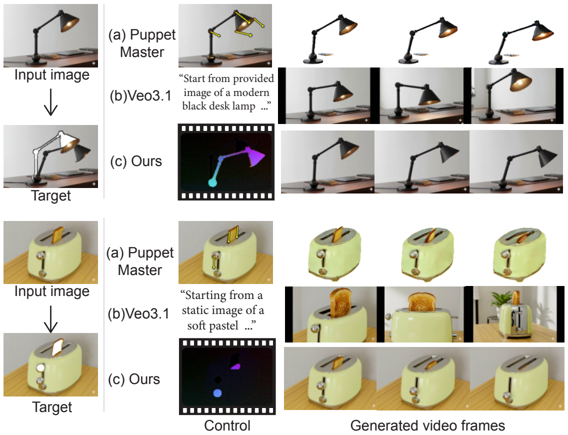


Figure 4. **Object/part manipulation.** Given an input image and a target manipulation (overlay), each method generates a short video. Baselines [12, 29] often drift from the target pose or distort unedited regions; our results track the specified articulation while preserving identity and handling any small disocclusions.

- Puppet-Master [29] takes as input an image along with up to five user-annotated straight arrows indicating manipulation intent, and produces a corresponding video demonstrating the specified object or part manipulation.
- Veo3.1 [12] is a multimodal language model (mLLM) capable of processing multi-modal inputs, and generating videos accordingly. In our setting, we prompt Veo3.1 to generate a video that begins with the given input image and follows the text prompt concat($\mathbf{T}$, $\mathbf{T}_m$) (Sec. 3.2, denoted as Veo3+$\mathbf{I}$+$\mathbf{T}_m$).

Visual and statistical comparisons are presented in Fig. 4 and Tab. 1, respectively. We use the 10 samples (*i.e.*, lamp and toaster) in our dataset. Qualitatively, despite extensive trial-and-error, Puppet-Master fails to rotate the lampshade towards the wall, and its generated videos appear blurry. Veo3.1 produces visually plausible results. However, specifying precise motion through text is challenging, and unintended camera movements frequently occur even when the prompt describes a fixed viewpoint. In contrast, our PDG-based approach enables accurate and controllable 3D transformations. Quantitatively, we only report the comparison with Veo3+$\mathbf{I}$+$\mathbf{T}_m$ (see supplementary for the comparison with Puppet-Master). Our method outperforms Veo3.1 by an order of magnitude on OptFlow, and also achieves superior performance on most video quality metrics.

4.2. Disocclusion-aware Video Editing

Competitors. Since no prior work addresses our novel motion- and disocclusion-aware video generation task, we created four baseline methods, as:

- DaS+$\mathbf{T}_{new}$: DaS is run with the input image, the tracking video, and the concatenated text prompt (*i.e.*, concat($\mathbf{T}$, $\mathbf{T}_{new}$), Sec. 3.3) to produce the resulting video.
- Pixel-CP: using the disocclusion mask and the user-edited image, we directly copy the disoccluded region from $\mathbf{I}_{edit}$ and paste it into the corresponding frames in $\mathbf{V}_m$.
- Pixel-CP++: instead of direct copy-paste, we re-run DaS conditioned on $\mathbf{I}_{edit}$, the reversed tracking video, and the reversed text (*e.g.*, 'open the drawer' → 'close the

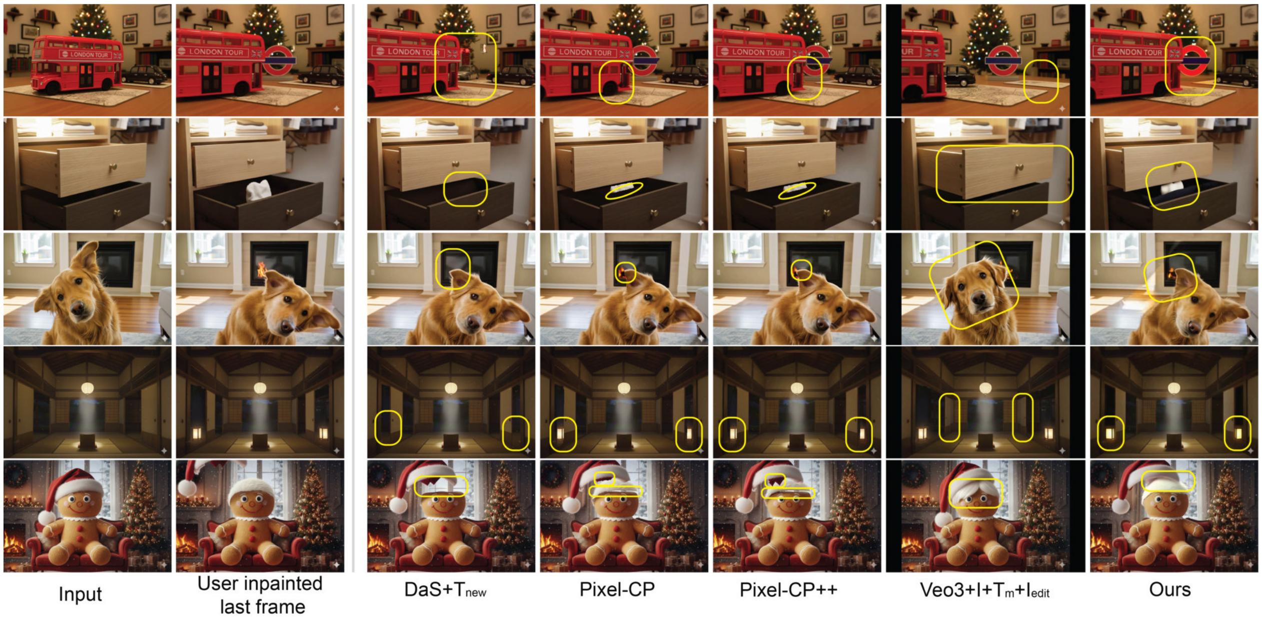


Figure 5. **Qualitative comparison.** Given an input image and a user-edited last frame specifying the desired reveal, each method generates image-to-video. The four baselines often miss the target reveal, drift in pose, or introduce seams/identity loss (yellow circles). **Ours** respects the user's final-frame content while preserving global appearance and motion, producing coherent videos across diverse scenes.

drawer'). The resulting backward video is reversed again, and its disoccluded regions are composited onto $\mathbf{V}_m$.

- Veo3+$\mathbf{I}$+$\mathbf{T}_m$+$\mathbf{I}_{edit}$: we prompt Veo3.1 to take as input $\mathbf{I}$, $\mathbf{I}_{edit}$, concat($\mathbf{T}$, $\mathbf{T}_m$, $\mathbf{T}_{new}$) to generate the corresponding video.

Results analysis. The comparison results are presented in Fig. 5 and Tab. 1. All methods are evaluated on the 40 samples in our dataset. For motion accuracy, all methods except Veo3+$\mathbf{I}$+$\mathbf{T}_m$+$\mathbf{I}_{edit}$ achieve comparable and superior accuracy, as they share the same tracking video obtained from our PDG. This confirms that text prompts alone are insufficient to specify precise motions, particularly for complex objects with multiple parts. Pixel-CP performs best in terms of last-frame similarity, as it directly replaces the edited last-frame pixels. Pixel-CP++ achieves the second-best scores with a slight degradation caused by inconsistencies between the forward and backward videos, leading to boundary misalignments that can be observed in Figs. 2 and 5 (highlighted boxes). In terms of video quality, both Pixel-CP and Pixel-CP++ yield higher scores on pixel fidelity (SSIM and PSNR) and perceptual similarity (LPIPS), but lower scores in video realism (FVDS and FVDC). This is because Pixel-CP lacks global visual effects (*e.g.*, shadows), while Pixel-CP++ introduces misalignment artifacts that disrupt temporal consistency.

DaS+$\mathbf{T}_{new}$ leverages text prompts to synthesize the new content or object in the disoccluded regions. However, textual descriptions cannot precisely capture object geometry and appearance, often producing objects that are visually similar but not identical, leading to lower last-frame similarity scores. Thanks to the diffusion priors in DaS, its video quality surpasses two pixel-based baselines.

Table 2. Ablation study results. We evaluate several choices of the replacement step M and report their effects on last-frame similarity and the overall video quality. The full statistics on all the evaluation metrics can be found in the supplmentary.

	Last-frame Similarity			Video Quality	
	Idiff (↓)	Idiff$_m$ (↓)	FID (↓)	FVDS (↓)	FVDC (↓)
M=25	23.86	7.81	60.69	**1629.01**	**1633.33**
M=30	23.85	7.45	59.48	1633.73	1637.90
(default) M=35	23.82	6.91	57.14	1639.47	1643.88
M=40	23.36	6.71	**56.95**	1641.26	1645.58
M=50	**23.28**	**6.46**	59.13	1640.33	1644.83

Given the images and text prompt, Veo3+$\mathbf{I}$+$\mathbf{T}_m$+$\mathbf{I}_{edit}$ generates videos with high realism but low pixel fidelity, as its frames are often contain uncontrolled elements (*e.g.*, randomly appearing hands) and view changes. Although the last frame is included in the input, it frequently appears in the wrong temporal position, leading to the lowest last-frame similarity scores. For the CLIP space text-to-video similarity, all methods achieve comparable scores. Visual results (in Fig. 5) are consistent with these findings.

4.3. User Evaluation

We have conducted a perceptual study to compare the quality of ours to those using alternative approaches.

Task. Each participant of the study was shown 15 pairs of videos along with the corresponding input images, inpainted regions, and tracking videos. Each of the 15 pairs of videos contained one video produced by our method and one video produced by one of the 5 alternative approaches (*i.e.*, a - DaS+$\mathbf{T}_{new}$, b - Pixel-CP, c - Pixel-CP++, d - Veo3+$\mathbf{I}$+$\mathbf{T}_m$, e - Veo3+$\mathbf{I}$+$\mathbf{T}_m$+$\mathbf{I}_{edit}$), in random order. We

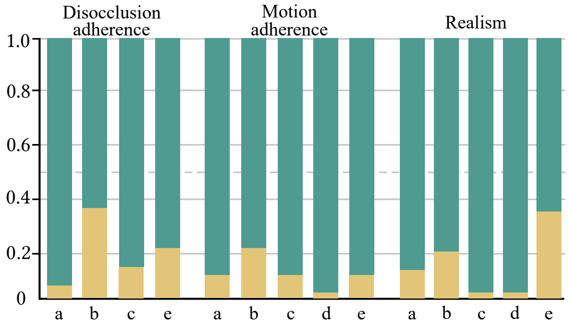


Figure 6. User evaluation results on three aspects (Disocclusion adherence, Motion adherence, and Realism), comparing our method with competitors (a-e). A higher percentage reflects a stronger preference for our method over the competitors.

created the pairs by randomly selecting 3 samples out of the 10 in the lamp and toaster subset, and 12 samples from the remaining 40 examples. At least one sample from each collected image was included to ensure coverage.

In the study, the user were asked, for each pair, to indicate the best result according to (i) *Disocclusion adherence* (how well the video matches the target specification within the revealed areas), (ii) *Motion adherence* (how closely the video follows the user-specified object motion shown in the tracking video), (iii) *Realism* (overall visual quality, including natural-looking shadows, minimal seams/artifacts, and smooth, consistent motion over time). Note, when the paired videos come from the lamp or toaster, only the realism and motion adherence questions were asked. Each video lasts for 3 seconds, and the entire study takes between 5 and 10 minutes to complete.

Answers. Figure 6 details the distribution of answers collected over 32 participants. Our method is largely favored over all competitors on all three criteria. For video realism, Veo3+$\mathbf{I}$+$\mathbf{T}_m$+$\mathbf{I}_{edit}$ outperforms other baselines, demonstrating its inherent ability to produce realistic videos without explicitly enforcing motion or disocclusion adherence. Pixel-CP ranks second in motion and disocclusion adherence, primarily due to its brute-force replacement strategy.

4.4. Ablation Study and Discussions

Feature replacement step M. To validate our choice of setting the replacement step to $M = 35$, we conduct an ablation study by varying M to 25, 30, 40, and 50. As shown in Tab. A1, when $M \geq 35$, the resulting video quality and last-frame similarity remain comparable. We therefore adopt $M = 35$ as our default, as using fewer replacement steps allows the diffusion model to harmonize better.

Feature vs. noise replacement. When adapting DaS, we choose to replace the features of $\mathcal{F}_s$ and $\mathcal{F}_{edit}$ based on the disocclusion mask. An alternative approach is to replace the per-step noise according to the same mask during the denoising process. In this case, the backward video (see Pixel-CP++ in Sec. 4.2) becomes essential, and its appearance cues are injected through the mask-based noise replacement. We implemented this variant and observed noticeable artifacts along the boundaries of the newly inpainted regions, even after testing different noise-injection time steps. Another possible reason for this failure could be the misalignment between the forward/backward videos.

4.5. Limitations and Future Work

Scope of deformation. Our PDG is part-based and best suited to articulated or piecewise-rigid motion; highly free-form dynamics (e.g., water splashes, fluttering flags, hair in wind) cannot be handled by our primitives. Extending PDG with deformable elements (e.g., ARAP or linear blend skinning (LBS) wrapping rig-based frames, or learned controllable blendshapes) is a promising direction.

Training-free approximation. Being training-free, our final-frame enforcement is only approximate: outputs converge *near* the target rather than matching it exactly under large disocclusions and/or depth ambiguity. Lightweight adaptation (e.g., constraint-aware adapters/LoRA, differentiable PDG-to-latent solvers) could tighten alignment while preserving generality.

Reachability and satisfiability. Not all user targets are attainable from a single input image. Formalizing *edit reachability* – the set of final frames consistent with PDG kinematics, visibility, and available evidence – is an open problem. Future work can develop a SAT-like criterion for controllable generative editing that *certifies* when a motion specification and final-frame content are jointly satisfiable.

Toward training-based PDG. Finally, a training-based variant that *learns* to follow PDG motion while *exactly* enforcing final-frame constraints could combine our controllability with the robustness of specialized backbones. This, however, requires a dataset of motion specifications plus final-frame targets, and objectives that jointly optimize pose fidelity, identity preservation, and disocclusion correctness.

5. Conclusion

We have presented a training-free pipeline for *image-to-video generation* that grants users explicit control over *disoccluded content* in the final frame. The core idea is to decouple *motion specification* from *appearance synthesis*: a lightweight, user-editable PDG deterministically, yet approximately, drives part motion, while a frozen diffusion prior acts as a motion-guided shader to synthesize appearance of both moving objects as well as disoccluded parts. Our latent space mixing towards final-frame enforcement reconciles PDG-driven motion with user-specified content, yielding controllable articulation and user authority over newly revealed regions, without requiring fine-tuning or paired edit data. We thus unlock a practical workflow that mixes *generative power* (for reveals) with *predictable control* (for motion and end frame).

References

[1] Adobe Inc. Adobe firefly: Generative ai by adobe. https://www.adobe.com/sensei/generative-ai/firefly.html, 2024. Accessed: 2025-10-30. 2

[2] Hadi Alzayer, Zhihao Xia, Xuaner Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. Magic fixup: Streamlining photo editing by watching dynamic videos. *ACM Transactions on Graphics*, 44(5):1–25, 2025. 3

[3] AUTOMATIC1111. Automatic1111 stable diffusion webui. https://github.com/AUTOMATIC1111/stable-diffusion-webui, 2024. Accessed: 2025-10-30. 2

[4] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18208–18218, 2022. 3

[5] Omri Avrahami, Rinon Gal, Gal Chechik, Ohad Fried, Dani Lischinski, Arash Vahdat, and Weili Nie. Diffuhaul: A training-free method for object dragging in images. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–12, 2024. 3

[6] Shariq Farooq Bhat, Niloy Mitra, and Peter Wonka. Loosecontrol: Lifting controlnet for generalized depth conditioning. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[7] Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8861–8870, 2024. 3

[8] 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. 3

[9] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22560–22570, 2023. 3

[10] Yen-Chi Cheng, Krishna Kumar Singh, Jae Shin Yoon, Alexander Schwing, Liang-Yan Gui, Matheus Gadelha, Paul Guerrero, and Nanxuan Zhao. 3d-fixup: Advancing photo editing with 3d priors. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–10, 2025. 3

[11] Comfy-Org. Comfyui: A node-based ui for stable diffusion. https://github.com/comfyanonymous/ComfyUI, 2024. Accessed: 2025-10-30. 2

[12] Google DeepMind. Veo: Video generation with moe-driven diffusion and audio generation. https://deepmind.google/models/veo/, 2024. Accessed: 2025-11-04. 1, 6

[13] Google DeepMind. Nano banana (gemini 2.5 flash image) model. Google Blog, 2025. Accessed: 2025-10-30. 1, 3

[14] FLUX Context. Flux context: Ai image and video generation platform. https://flux-context.org/, 2025. Accessed: 2025-10-30. 2

[15] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ICLR*, 2023. 3

[16] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *ACM SIGGRAPH 2025 Conference Papers*, pages 1–12, 2025. 1, 2, 3, 4

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *ICLR*, 2023. 3

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[19] Xingzhong Hou, Boxiao Liu, Yi Zhang, Jihao Liu, Yu Liu, and Haihang You. Easydrag: Efficient point-based manipulation on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8404–8413, 2024. 3

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[21] Fengling Hu, Andrew Chen, Hannah Horng, Vishnu Bashyam, Christos Davatzikos, Aaron Alexander-Bloch, Mingyao Li, Haochang Shou, Theodore Satterthwaite, Meichen Yu, and Russell Shinohara. Image harmonization: A review of statistical and deep learning methods for removing batch effects and evaluation metrics for effective harmonization. *NeuroImage*, 274:120125, 2023. 3

[22] InvokeAI Community. Invokeai: Stable diffusion toolkit. https://github.com/invoke-ai/InvokeAI, 2024. Accessed: 2025-10-30. 2

[23] 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. 3

[24] 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

[25] Juil Koo, Paul Guerrero, Chun-Hao P Huang, Duygu Ceylan, and Minhyuk Sung. Videohandles: Editing 3d object compositions in videos using video generative priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17692–17701, 2025. 1, 3

[26] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion, 2023. 3

[27] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dragapart: Learning a part-level motion prior for articulated objects. In *European Conference on Computer Vision*, pages 165–183. Springer, 2024. 3

[28] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Puppet-master: Scaling interactive video generation as a motion prior for part-level dynamics. *arXiv preprint arXiv:2408.04631*, 2024. 1

[29] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Puppet-master: Scaling interactive video generation as a motion prior for part-level dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13405–13415, 2025. 3, 4, 6, 1

[30] Yue Ma, Xiaodong Cun, Sen Liang, Jinbo Xing, Yingqing He, Chenyang Qi, Siran Chen, and Qifeng Chen. Magicstick: Controllable video editing via control handle transformations. *arXiv preprint arXiv:2312.03047*, 2023. 3

[31] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *ICLR*, 2022. 3

[32] Oscar Michel, Anand Bhattad, Eli VanderBilt, Ranjay Krishna, Aniruddha Kembhavi, and Tanmay Gupta. Object 3dit: Language-guided 3d-aware image editing, 2023. 3

[33] Midjourney, Inc. Midjourney: Generative ai art platform. https://www.midjourney.com/, 2024. Accessed: 2025-10-30. 2

[34] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023. 3

[35] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4296–4304, 2024. 3

[36] OpenAI. Dall·e 3 by openai. https://openai.com/dall-e-3, 2024. Accessed: 2025-10-30. 2

[37] OpenAI. Chatgpt-5.1. https://www.openai.com/, 2025. Large language model. 2

[38] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023. 3

[39] Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J Mitra. Diffusion handles enabling 3d edits for diffusion models by lifting activations to 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7695–7704, 2024. 3

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 6

[41] 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

[42] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 3

[43] Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling, 2024. 1, 3

[44] Yujun Shi, Jun Hao Liew, Hanshu Yan, Vincent YF Tan, and Jiashi Feng. Instadrag: Lightning fast and accurate drag-based image editing emerging from videos. *CoRR*, 2024. 3

[45] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024. 3

[46] Joonghyuk Shin, Daehyeon Choi, and Jaesik Park. Instantdrag: Improving interactivity in drag-based image editing. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–10, 2024. 3

[47] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3626–3636, 2022. 6

[48] Stability AI. Stable diffusion xl (sdxl). https://stability.ai/news/stable-diffusion-xl-release, 2023. Accessed: 2025-10-30. 2

[49] Stability AI. Stable diffusion: Open source image synthesis model. https://stability.ai/stable-diffusion, 2024. Accessed: 2025-10-30. 2

[50] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 6

[51] Chen Wang*, Chuhao Chen*, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, and Lingjie Liu. Physctrl: Generative physics for controllable and physics-grounded video generation. In *NeurIPS*, 2025. 3

[52] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024. 3

[53] Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, Anthony Chen, Huaxia Li, Xu Tang, and Yao Hu. Instantid: Zero-shot identity-preserving generation in seconds. *arXiv preprint arXiv:2401.07519*, 2024. 3

[54] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings*

of the Computer Vision and Pattern Recognition Conference, pages 5261–5271, 2025. 4

[55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[56] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision*, pages 112–129. Springer, 2024. 3

[57] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024. 1

[58] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 6

[59] 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, 6

[60] 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. 3

[61] Xin Yu, Tianyu Wang, Soo Ye Kim, Paul Guerrero, Xi Chen, Qing Liu, Zhe Lin, and Xiaojuan Qi. Objectmover: Generative object movement with video prior. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17682–17691, 2025. 3

[62] 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. 3

[63] 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. 6