A. Additional Details of the Drag Encoding

Here, we give a formal definition of $\mathrm{enc}(\cdot, s)$ introduced in Sec. 3.2. Recall that $\mathrm{enc}(\cdot, s)$ encodes each drag $d_k := (u_k, v_k^{1:N})$ into an embedding of shape $N \times s \times s \times 6$. For each frame n, the first, middle, and last two channels (of the $c = 6$ in total) encode the spatial location of u_k, v_k^n, and v_k^N, respectively. Formally, $\mathrm{enc}(d_k, s)[n, :, :, :2]$ is a tensor of all negative ones except for $\mathrm{enc}(d_k, s)[n, \lfloor \frac{s \cdot h}{H} \rfloor, \lfloor \frac{s \cdot w}{W} \rfloor, :2] = \left(\frac{s \cdot h}{H} - \lfloor \frac{s \cdot h}{H} \rfloor, \frac{s \cdot w}{W} - \lfloor \frac{s \cdot w}{W} \rfloor\right)$ where $u_k = (h, w) \in \Omega = \{1, \ldots, H\} \times \{1, \ldots, W\}$. The other 4 channels are defined similarly, with u_k replaced by v_k^n and v_k^N.

B. Additional Details of Data Curation

B.1. Implementation Details

We use the categorization provided by GObjaverse [48] and exclude 3D models classified as '`Poor-Quality`' as a pre-filtering step prior to our proposed filtering pipelines (Sec. 4).

When using GPT-4V to filter Objaverse-Animation into Objaverse-Animation-HQ, we designed the following prompt to cover a wide range of cases to be excluded:

> **System**: You are a 3D artist, and now you are being shown some animation videos depicting an animated 3D asset. You are asked to filter out some animations.
> You should filter out the animations that:
> (1) have trivial or no motion, i.e., the object is simply scaling, rotating, or moving as a whole without part-level dynamics;
> or (2) depict a scene and only a small component in the scene is moving;
> or (3) have motion that is imaginary, i.e., the motion is not the usual way of how the object moves and it's hard for humans to anticipate;
> or (4) have very large global motion so that the object exits the frame partially or fully in one of the frames;
> or (5) have changes in object color that are not due to lighting changes;
> or (6) have motion that causes different parts of the same object to disconnect, overlap in an unnatural way, or disappear;
> or (7) have motion that is very chaotic, for example objects exploding or bursting apart.
> **User**: For the following animation (as frames of a video), `frame1`, `frame2`, `frame3`, `frame4`, tell me, in a single word 'Yes' or 'No', whether the video should be filtered out or not.

The cost of GPT-4V data filtering is about $500.

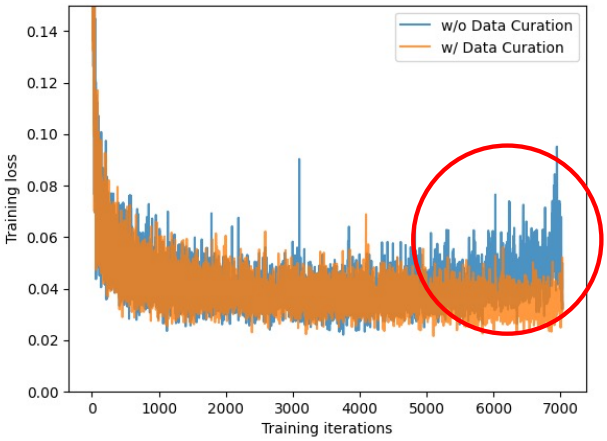

Figure 7. Data curation helps stabilize training.

Setting	PSNR↑	SSIM↑	LPIPS↓	FVD↓
w/o Data Curation	6.04	0.411	0.703	1475.35
w/ Data Curation	**19.87**	**0.884**	**0.181**	**624.47**

Table 3. Training on more abundant but lower-quality data leads to lower generation quality. Here, 'w/o Data Curation' model is trained on Objaverse-Animation while 'w/ Data Curation' model is trained on Objaverse-Animation-HQ. Both models are trained for 7k iterations. Evaluation is performed on the test split of Drag-a-Move.

B.2. Less is More: Data Curation Helps at Scale

To verify that our data curation strategy from Sec. 4 is effective, we compare two models trained on Objaverse-Animation and Objaverse-Animation-HQ, respectively, under the same hyperparameter setting. The training dynamics are visualized in Fig. 7. The optimization collapses towards 7k iterations when the model is trained on a less curated dataset, resulting in much lower-quality video samples (Tab. 3). This suggests that when fine-tuning a pre-trained video diffusion model to generate part-level motion, the quality of the data is more critical than its quantity.

C. Additional Experiment Details

C.1. Training Details

Data. Our final model is fine-tuned on the combined dataset of Drag-a-Move [31] and Objaverse-Animation-HQ (Sec. 4). During training, we balance various types of part-level dynamics to control the data distribution. We achieve this by leveraging the categorization provided by GObjaverse [48] and sampling individual data points with the following hand-crafted distribution: p(Drag-a-Move) $= 0.3$, p(Objaverse-Animation-HQ, category '`Human-Shape`') $= 0.25$, p(Objaverse-Animation-HQ,

category 'Animals') $= 0.25$, p(Objaverse-Animation-HQ, category 'Daily-Used') $= 0.05$, p(Objaverse-Animation-HQ, other categories) $= 0.15$.

Architecture. We zero-initialize the final convolutional layer of each adaptive normalization module before fine-tuning. With our introduced modules, the parameter count increases to 1.68B from the original 1.5B in SVD.

Training. We fine-tune the base SVD on videos of 256×256 resolution and $N = 14$ frames with a batch size of 64 for $12,500$ iterations. We adopt SVD's continuous-time noise scheduler, shifting the noise distribution towards more noise with $\log \sigma \sim \mathcal{N}(0.7, 1.6^2)$, where σ is the continuous noise level following the presentation in [4]. Training takes roughly 10 days on a single Nvidia A6000 GPU, where we accumulate gradients for 64 steps. We enable classifier-free guidance (CFG) [23] by randomly dropping the conditional drags $\mathcal{D}$ with a probability of 0.1 during training. Additionally, we track an exponential moving average of the weights at a decay rate of 0.9999.

C.2. Inference and Evaluation Details

Inference. Unless stated otherwise, samples are generated using $S = 50$ diffusion steps. We adopt linearly increasing CFG [4] with a maximum guidance weight of 5.0. Generating a single video takes roughly 20 seconds on an Nvidia A6000 GPU.

Baselines. For DragNUWA [70], DragAnything [67], and Image Conductor [32], we use their publicly available checkpoints. DragNUWA and DragAnything operate at a resolution of 576×320, and Image Conductor at 384×256. Following previous work [31], we first pad the square input image y along the horizontal axis to the correct aspect ratio and resize it to the corresponding resolution, then remove the padding from the generated frames and resize them back to 256×256. For methods that require text prompts (*i.e.*, DragNUWA and Image Conductor), we use generic prompts to describe the category of the evaluation images (*e.g.*, '`A Furniture`' for Drag-a-Move and '`A person`' for Human3.6M). Note that Image Conductor is trained on 16-frame videos instead of 14-frame ones. We experimented with (1) simply generating 14 frames at inference time; and (2) generating 16 frames and discarding the last two frames. The latter gives slightly better results, which we report. We find that tasking it to generate 14-frame videos produces reasonable results which we report. All metrics are computed on 14-frame videos of resolution 256×256.

We train DragAPart [31] for 100k iterations using its official implementation on the same combined dataset of Drag-a-Move and Objaverse-Animation-HQ used for training Puppet-Master. Since DragAPart is an image-to-image model, we independently generate 14 frames conditioned on gradually extending drags to obtain the video.

For Sora [7], we uploaded the conditioning image in Fig. 4 as the start frame. Since the model does *not* support motion control, we manually crafted the following prompt to convey the motion condition:

> A photorealistic video of a modern, light grey wooden sideboard with a natural wood top. The three drawers at the top remain completely static and closed throughout the entire video, without any movement or displacement. From this initial state, only the bottom cabinet doors begin to slowly and smoothly close, moving in a natural, physically plausible manner. The motion follows proper hinge mechanics, ensuring perfect alignment, symmetry, and realism, with no jerky or unnatural movement. The camera remains fixed in the same frontal view, maintaining the exact perspective of the reference image. The lighting is soft and even, enhancing the wood texture, clean lines, and elegant design without casting harsh shadows or introducing distractions. The video maintains a high-quality, cinematic appearance, with no additional objects or background elements.

D. Video Diffusion Models on Out-of-Domain Resolutions

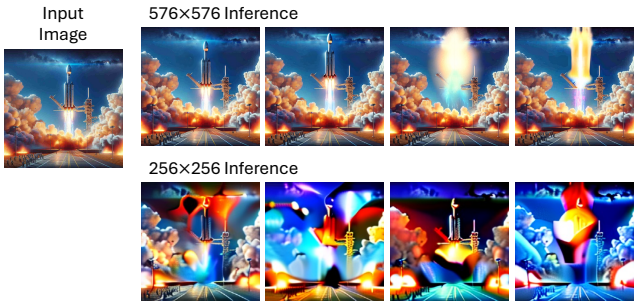

Figure 8. Stable Video Diffusion *fails* to generalize robustly to out-of-domain resolutions at inference time.

The convolution and attention modules in video diffusion models like SVD are *not* invariant to input resolution. As demonstrated in Fig. 8, our base model SVD, which was trained on videos with resolution 1024×576, *cannot* generate high-quality videos at out-of-domain resolutions such as 256×256. We hypothesize that this resolution shift makes fine-tuning susceptible to local optima, resulting in visually cluttered generations (Fig. 6). All-to-first attention (Sec. 3.3) significantly reduces this appearance degradation.

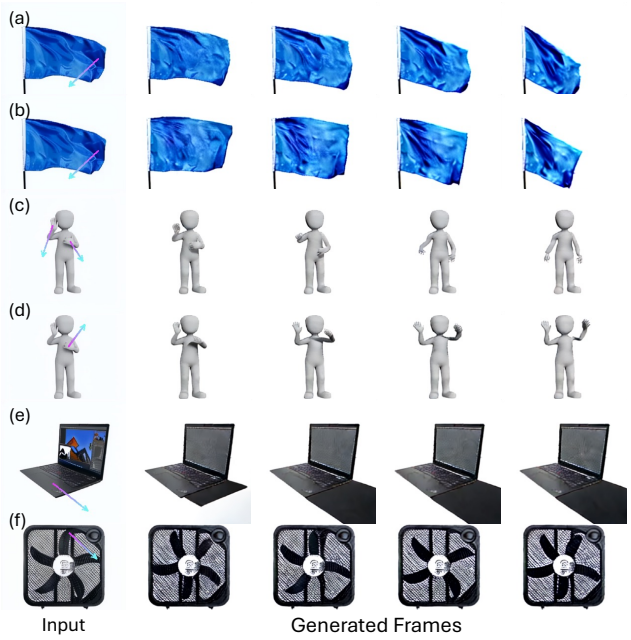

Figure 9. **More examples** generated by Puppet-Master.

Figure 10. **Results** on images with diverse backgrounds.

E. Discussions

Motion diversity. In Fig. 9(a-d), we show that Puppet-Master can generate diverse part-level animations, both across different random seeds when conditioned on the same input image and set of drags (*i.e.*, a and b), and across different sets of drags when conditioned on the same input image (*i.e.*, c and d).

Part-level *vs*. object-level motion. In this work, we focus on synthesizing *internal*, *part-level* motion. To achieve this, we curated Objaverse-Animation-HQ to specifically learn motions involving object parts being manipulated. As a result, Puppet-Master is not designed for *global* object motion and may produce artifacts when the input drag(s) do *not* correspond to meaningful part-level movement (Fig. 9e).

Failure cases. Puppet-Master may fail to maintain the shape of objects, occasionally leading to the disappearance of certain parts. This issue is particularly evident when physically plausible motion necessitates precise coordination among multiple object parts, such as the five fan blades in Fig. 9f.

Results with real-world backgrounds. Although all training frames are rendered with a white background, Puppet-Master retains some ability from the SVD backbone to handle complex backgrounds, as illustrated in Fig. 10. Better results could be obtained by incorporating, *e.g.*, random backgrounds during training.

Limitations. Another limitation of our model is its slight difficulty in preserving the exact color appearance of objects during inference on real-world images. This issue arises due to two primary factors: (1) the synthetic 3D models in Objaverse-Animation-HQ typically feature high-contrast, stylized textures, leading to a train-test discrepancy in color distributions; and (2) when testing at a lower resolution (*e.g.*, 256×256) compared to the native resolution of SVD, noise in the denoiser's output can propagate across a larger region of the image because of the fixed receptive field of convolutional layers, leading to many instances having a slightly flickering appearance.

Future work. While most motion-conditioned video generators prioritize object-level motion over fine-grained part-level motion, we have demonstrated it is feasible to learn a part-level motion prior using a modestly sized, high-quality synthetic dataset that generalizes effectively to real-world data. Future research may develop a dynamic routing mechanism that integrates both part-level and object-level dynamics.