A ADDITIONAL DETAILS OF THE DRAG ENCODING

Here, we give a formal definition of $\mathrm{enc}(\cdot, s)$ introduced in Section 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 Qiu et al. (2024) and exclude the 3D models classified as ‘`Poor-Quality`’ as a pre-filtering step prior to our proposed filtering pipelines (Section 4).

When using GPT-4V to filter Objaverse-Animation into Objaverse-Animation-HQ, we design 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 estimated to be $500.

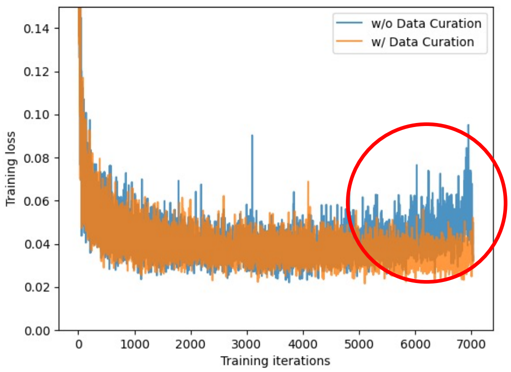

Figure 6: Data curation helps stabilize training.

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

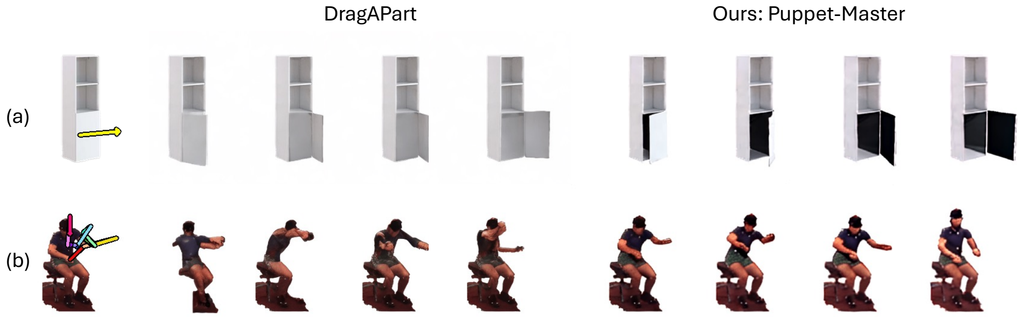

Figure 7: **Qualitative Comparison** with DragAPart. The videos generated by DragAPart lack temporal consistency: (a) the door initially opens to the left, but later it is switched to open to the right, and it partially closes between the second and third frames visualized here; (b) DragAPart fails to generalize to out-of-domain cases, resulting in distorted motion.

B.2 LESS IS MORE: DATA CURATION HELPS AT SCALE

To verify that our data curation strategy from Section 4 is effective, we compare two models trained on Objaverse-Animation and Objaverse-Animation-HQ respectively under the same hyper-parameter setting. The training dynamics are visualized in Fig. 6. The optimization collapses towards 7k iterations when the model is trained on a less curated dataset, resulting in much lower-quality video samples (Table 3). This suggests that the data's quality matters more than quantity at scale.

C ADDITIONAL EXPERIMENT DETAILS

Data. Our final model is fine-tuned on the combined dataset of Drag-a-Move (Li et al., 2024c) and Objaverse-Animation-HQ (Section 4). During training, we balance over various types of part-level dynamics to control the data distribution. We achieve this by leveraging the categorization provided by Qiu et al. (2024) and sampling individual data points with the following hand-crafted distribution: $p(\text{Drag-a-Move}) = 0.3$, 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 is pumped to 1.68B from the original 1.5B SVD.

Training. We fine-tune the base SVD on videos of 256×256 resolution and $N = 14$ frames with batch size 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 Blattmann et al. (2023a). The training takes roughly 10 days on a single Nvidia A6000 GPU where we accumulate gradient for 64 steps. We enable classifier-free guidance (CFG) (Ho & Salimans, 2022) 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.

Inference. Unless stated otherwise, the samples are generated using $S = 50$ diffusion steps. We adopt the linearly increasing CFG (Blattmann et al., 2023a) with maximum guidance weight 5.0. Generating a single video roughly takes 20 seconds on an Nvidia A6000 GPU.

Baselines. For DragNUWA (Yin et al., 2023) and DragAnything (Wu et al., 2024), we use their publicly available checkpoints. They operate on a different aspect ratio (*i.e.*, 576×320). Following previous work (Li et al., 2024c), we first pad the square input image y along the horizontal axis to the correct aspect ratio 1.8 and resize it to 576×320, and then remove the padding of the generated frames and resize them back to 256×256. We train DragAPart (Li et al., 2024c) for 100k iterations using its official implementation on the same combined dataset of Drag-a-Move and Objaverse-Animation-HQ which we used for training PuppetMaster. Since DragAPart is an image-to-image model, we independently generate N frames conditioned on gradually extending drags to obtain the video. All metrics are computed on 256×256 videos.