

Generating Fit Check Videos with a Handheld Camera

Bowei Chen Brian Curless Ira Kemelmacher-Shlizerman Steven M. Seitz
University of Washington

{boweiche, curless, kemelmi, seitz}@cs.washington.edu

1. Motion and Background Retrieval

Motion Acquisition. For motion retrieval, the key idea is to retrieve the top-k closest matches by computing dynamic time warping (DTW) distances between the recorded and candidate motions’ orientation and translation. For orientation, we focus on the horizontal direction (yaw) since a mobile phone’s IMU provides a more robust estimate in this axis compared to roll and pitch. Moreover, yaw captures spinning, which, along with translation, effectively represents most of the motions. For simplicity, we define orientation as yaw throughout the rest of the paper. Below we introduce the details of motion retrieval.

First, the user performs a motion while holding a phone, recording IMU data to obtain orientation $O_{1:\tilde{N}}$ and global translation $\mathcal{T}_{1:\tilde{N}}$, where \tilde{N} is the motion length. Here, the orientation is directly derived from the device’s motion sensors, while the global translation is computed through a multi-step process. First, we filter the acceleration data to reduce noise. Then, we integrate the filtered acceleration to obtain velocity, followed by a second integration to derive translation over time.

For each candidate motion i in the database (stored as RGB video), we extract the SMPL [13] sequence $S_{1:\tilde{N}}^i$ and global translation $\mathcal{T}_{1:\tilde{N}}^i$ using a pretrained human pose estimator [10], where \tilde{N} is the motion length. The candidate’s orientation $O_{1:\tilde{N}}^i$ is obtained from the yaw component of the global rotation of the root joint in $S_{1:\tilde{N}}^i$. We compute the distance between the recorded motion and each candidate as:

$$D^i = D_{dtw}(O_{1:\tilde{N}}, O_{1:\tilde{N}}^i) + \alpha D_{dtw}(\mathcal{T}_{1:\tilde{N}}, \mathcal{T}_{1:\tilde{N}}^i), \quad (1)$$

where D_{dtw} is dynamic time warping (DTW) to handle sequences of different lengths. $\alpha = 0.1$ is a constant that balances orientation and translation importance. All candidates are ranked based on D^i , and the top-k motions are retrieved for user selection. After selection, we extract the DWPose sequence $P_{1:N}$ as target pose sequence.

In practice, we set our motion database to the same as our training set, as the training videos consist of fit check



Figure 1. **Examples of Images from Background Database.** The dataset contains a diverse collection of images spanning both indoor and outdoor environments.

videos, which provide well-suited motions.

Background Acquisition. The retrieved motion must be compatible with the new background, ensuring natural alignment with the ground plane. One approach is to rotate the SMPL sequence of the retrieved motion to align with the ground plane of any given background, and then extract DWPose from the rendered motion. However, this fails due to foot sliding in the estimated SMPL sequence. Instead, we opt for background retrieval, selecting backgrounds where the ground plane is closely aligned with the retrieved motion. Specifically, we begin by utilizing a pretrained depth estimation model [19] and a pretrained image segmentation model [8] to estimate the ground plane normal for both the original background – where the motion was captured – and all candidate backgrounds. Then, we retrieve the top-k backgrounds from our database with the closest ground plane normals. Once a background is selected, we automatically scale and translate $P_{1:N}$ to keep foot keypoints on the ground.

For our background database, we curate a diverse database of indoor and outdoor environments, consisting of 800 images, including both AI-generated and real images. Fig. 1 shows some examples of images from the background database.



Figure 2. **Training Frames from Fit Check Video.** Samples of training frames from our fit check videos, showing that our video capture diverse backgrounds, front and back perspectives, and a wide range of fit check motions. Four non-consecutive frames are shown for each video.

2. Implementation Details

We will introduce the implementation details below, and all the images and videos are operated in the resolution of height 1024 and width 576, same as the MimicMotion. *We will release the code upon acceptance.*

2.1. Our Method

Reference Image Augmentation. During training, we apply a pretrained image harmonization network [2] to adjust the color tone of the front and back reference images, I'_{fr} and I'_{bk} . This network takes a composite (unharmonized) image and a foreground mask as input, then produces a harmonized image where the foreground seamlessly blends with the background.

For each reference image, we first apply a pretrained image matting method [15] to obtain the foreground mask. We then randomly select a background image from our background database and composite it with the extracted human foreground to create a composite image. The composite image and its corresponding foreground mask are fed into the image harmonization network, which adjusts the foreground color to better match the background. After obtaining the harmonized output images, we remove their backgrounds and use the resulting images as training data. We apply this process independently to both I'_{fr} and I'_{bk} , using different background images for each. This is to accommodate the natural color tone variations between front and back selfies at test time, even when captured almost simultaneously.

Model Architecture and Parameters. We adopt the 3D denoising UNet, image encoder, pose encoder, VAE en-

coder, and VAE decoder architectures from MimicMotion [21]. Due to computational constraints, we set T to 6, which means each training batch contains 8 frames (including front and back reference image). Larger T can be used if more computational resource is available. While we train on 8 video frames per batch, we find that the model can be extended to generate 16-frame or 24-frame sequences at test time, improving efficiency without a noticeable loss in quality.

Model Training. We initialize the model using the pretrained checkpoint “MimicMotion.1.pth” from MimicMotion. We did not apply regional loss amplification in MimicMotion because we found that this does not improve the results in our experiments. During training, we randomly sample the front and back view images, I'_{fr} and I'_{bk} , from the training video $V'_{1:N}$. To sample a T -length sequence $V'_{1:T}$ from the training video, we apply the following strategy:

- Randomly select frames from the video (20% of the time)
- Select a sequence containing at least one front-facing frame (not necessarily I'_{fr}) (40% of the time)
- Select a sequence containing at least one back-facing frame (not necessarily I'_{bk}) (40% of the time)

Additionally, we apply reference image augmentation to both I'_{fr} and I'_{bk} for 50% of time.

During training, each conditioning feature – f_v , f_{im} and f_p – is randomly dropped (set to zero) 10% of the time, following the classifier-free guidance method [5]. This allows us to control the strength of each conditional signal during inference. Training runs on a single NVIDIA A100 GPU with a batch size of 1 and a learning rate of 1e-5, for 220K steps (around 112 hours). Fig. 2 presents examples of our



Figure 3. **Collected Front-Facing Samples for Fine-Tuning.** We collect front-facing images from the web that contain visible shadows or reflections. These images are used to generate data pairs for the fine-tuning dataset. Fine-tuning on high-quality images enhances sharpness and improves the generation of shadows and reflections.

training data.

Model Fine-Tuning. We fine-tune the trained model on a high-quality image dataset. Fig. 3 presents examples of the front-facing human images we collected, which are later used to generate data pairs for fine-tuning. The shadow and reflection regions are manually annotated. During fine-tuning, we omit reference image augmentation, as we observe that the model becomes confused by color tone shifts, resulting in frames with unnatural colors. We use the same strategy as the training time to drop the conditioning feature. We apply a weighted loss strategy during fine-tuning. Specifically, we assign a higher weight $\beta = 2$ to the loss computed in the shadow and reflection regions, while maintaining a weight of 1 for the rest of the region. For optimization, we use a learning rate of $1e-6$ and fine-tune for 1K steps, which takes approximately 30 minutes.

Model Inference. At inference, we set the guidance scale to 2 and the number of overlapping frames to 4. The denoising time step is set to 25, and we use the Euler scheduler [9].

2.2. Baseline Details

Human Animation Baselines. We initialize all baselines from their pretrained checkpoints and train them on our dataset on a single NVIDIA A100 GPU for a fair comparison. Additionally, we set the frame length for all baselines to 8, aligning with our settings.

For Animate Anyone[6], since the official code is unavailable, we choose to use a widely-adopted unofficial implementation[12]. We follow the same hyperparameter settings as this codebase. The first stage of training runs for 100K steps, taking approximately 40 hours to converge. The second stage runs for 40K steps, requiring around 24 hours to complete. We observe that further training degrades performance.

For Champ [22], we use the official code. The first stage of training is conducted for 100K steps, taking approximately 35 hours to converge. The second stage runs for

40K steps, requiring about 20 hours to complete. Similar to Animate Anyone, we find that additional training negatively impacts performance.

For StableAnimator [17], we follow the official implementation and train the model for 220K steps with a learning rate of $1e-5$, taking approximately 132 hours to complete.

For MimicMotion [21], since no training code is provided, we implement our own training procedure. We train the model for 220K steps with a learning rate of $1e-5$, which takes around 99 hours to complete.

Motion Retrieval Baseline. As there are no existing IMU-to-motion retrieval baselines, we adopt the text-to-motion retrieval method TMR[14] as our baseline. We use the official implementation and run the pretrained model (trained on the HumanML3D dataset[4]) on the recorded motion and our motion database to retrieve the top-k matching motions.

3. Experiments

3.1. More Results for Selfie Input

Fig. 5 presents additional results of our method on real selfie captures. Our approach generates high-quality fit check videos featuring diverse outfits in both indoor and outdoor settings, accurately capturing a wide range of poses with realistic shading, reflections, and shadows.

3.2. Comparison with Baseline

3.2.1. Datasets

In addition to evaluating our model on the self-captured real selfie dataset presented in the main paper, we conduct further analysis on three additional test sets: (1) the test set from our self-collected dataset, (2) the test set of UBC Fashion dataset [20], and (3) the TikTok dataset [7]. We filter out videos that do not include a back view. After filtering, our dataset test set contains 149 videos, with an average of 68 frames per video. The UBC Fashion dataset consists of 100 videos, averaging 98 frames per video, while the TikTok dataset includes 19 videos, with an average of 115 frames per video.

For evaluation, we randomly sample a front and back image as reference inputs, while the input pose sequence is extracted from the corresponding ground truth (GT) video. The input backgrounds in our test set and the TikTok dataset are obtained using the same inpainting strategy as in our training set. However, for the UBC Fashion dataset, we use a plain white background instead, as inpainting a nearly white background with Stable Diffusion introduces artifacts. Finally, we compare the generated video with the GT video.

Notably, the input reference images in these datasets are captured from a third-person perspective rather than as selfies. This setup enables us to evaluate model performance on



Figure 4. **Results with the Same Capture under Different Virtual Backgrounds.** The first column shows the input selfies and target pose; the others show results under different virtual backgrounds (insets: input backgrounds). Despite strong left lighting in the selfies, our method adapts shading to each background.

non-selfie inputs. Furthermore, since these images are sampled from the GT video, they share similar lighting conditions with the GT. This also differs from our selfie setup, but we still evaluate on these datasets for a more comprehensive evaluation.

Please note that all methods are trained solely on our training set, with baselines initialized from their official checkpoints.

3.2.2. Qualitative Comparison

Fig. 4 shows more results of the same captures under different virtual backgrounds.

Fig. 6, Fig. 7, and Fig. 8 present additional comparisons on real selfie captures. We observe the following: (1) Animate Anyone and Champ exhibit artifacts such as inaccurate clothing patterns and artifacts around the shoes (row 3, column 3 in Fig. 7). (2) MimicMotion and StableAnimator fail to accurately reconstruct the appearance of the back view, demonstrating that simple modifications to these methods do not effectively utilize the additional reference image input. Additionally, they struggle to capture fine details in the front view (e.g. missing logo in row 5 in Fig. 8) and produce blurry patterns, whereas our method preserves accurate and sharp patterns due to the fine-tuning stage. (3) Our method surpasses all baselines in both appearance and pose fidelity while also generating more realistic reflections and shadows on the floor.

Fig. 9 presents a qualitative comparison on the UBC Fashion dataset. We observe the following: (1) Image diffusion-based methods (Animate Anyone and Champ) exhibit noticeable background color shifts, indicating their limited generalization ability to unseen backgrounds. Additionally, they introduce visible artifacts on faces and bodies and fail to accurately capture body shape. (2) Video diffusion-based baselines (MimicMotion and StableAnimator) struggle to reconstruct the appearance of the back view accurately, highlighting that simple modifications to these methods do not effectively utilize the additional reference image input. (3) Our method outperforms all baselines in

Table 1. **Quantitative Comparisons on Our Test Set.** All methods are evaluated without face refinement as post-processing. For each metric, the best and second-best methods are highlighted in bold and underline, respectively. Our method outperforms all tested baselines across all metrics. The ablation variant, *Ours-RIA*, achieves results comparable to *Ours* on this dataset because the input front and back images share the same background as the ground truth (GT) frames, making reference image augmentation less necessary in this case.

Method	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	FID \downarrow	FVD-VID \downarrow	FVD \downarrow
Animate Anyone [6]	0.733	0.242	19.60	128.8	57.43	421.7
Champ [22]	0.763	0.231	20.60	91.72	32.40	372.9
StableAnimator [17]	0.771	0.219	21.03	97.26	33.33	394.7
MimicMotion [21]	0.779	0.211	21.34	88.02	30.51	366.1
Ours-FG-MRA-FT	0.776	0.205	21.20	89.19	29.30	356.6
Naive+RefNet	0.781	0.202	22.38	87.52	29.40	342.4
Ours-MRA-FT	0.782	0.198	22.64	86.74	28.16	308.7
Ours-FG	0.781	0.203	22.40	87.64	25.76	311.0
Ours-MRA	<u>0.796</u>	0.186	23.08	82.82	23.71	292.2
Ours-RIA	0.795	<u>0.184</u>	<u>23.15</u>	<u>79.07</u>	22.44	<u>281.5</u>
Ours-FT	0.786	0.196	22.81	86.68	26.00	298.0
Ours-FT + Joint Training	0.789	0.188	23.01	83.22	26.42	289.3
Ours-FT + Full FT	0.792	0.186	23.07	81.52	25.42	324.1
Ours	0.799	0.183	23.61	79.02	<u>23.11</u>	279.3

both appearance and pose fidelity, producing more realistic and coherent results.

Fig. 10 presents a qualitative comparison on the TikTok dataset. We observe the following: (1) Animate Anyone and Champ exhibit noticeable artifacts, such as missing body parts (e.g., row 1, column 4, and row 3, column 4) and inaccurate clothing patterns (rows 2 to 5). (2) MimicMotion and StableAnimator struggle to accurately reconstruct the appearance of the back view, demonstrating that simple modifications to these methods do not effectively utilize the additional reference image input. Additionally, they produce blurry patterns (e.g., shorts in row 1), whereas our method generates sharper details due to the design of the fine-tuning stage. (3) Our method surpasses all baselines in both appearance and pose fidelity, delivering more realistic and coherent results.

3.2.3. Quantitative Comparison

Tab. 1, 2, and 3 present the quantitative results of our model compared to the baselines across the three datasets. We observe that Champ performs competitively among the baselines on our test set in terms of video-related metrics, FVD-VID and FVD, but performs worse on the other two datasets. This indicates that this image diffusion-based method achieves better temporal consistency when the input reference images and background are in-distribution. However, its performance degrades significantly for out-of-distribution inputs, demonstrating poor generalization ability.

As discussed in the main paper, our method outperforms all baselines on all metrics by employing a novel frame generation strategy with multi-reference attention and a fine-tuning approach, leading to enhanced appearance fidelity

Table 2. **Quantitative Comparisons on the UBC Fashion Dataset.** All methods are evaluated without face refinement as post-processing. For each metric, the best and second-best methods are highlighted in bold and underline, respectively. Our method outperforms all tested baselines across all metrics. The ablation variant, *Ours-RIA*, achieves results comparable to *Ours* on this dataset because the input front and back images share the same background as the ground truth (GT) frames, making reference image augmentation less necessary in this case.

Method	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	FID \downarrow	FVD-VID \downarrow	FVD \downarrow
Animate Anyone [6]	0.902	0.130	17.08	60.27	51.94	480.0
Champ [22]	0.888	0.130	16.43	59.34	56.06	363.7
StableAnimator [17]	0.914	0.071	21.61	58.47	31.67	191.5
MimicMotion [21]	0.918	0.069	22.05	56.27	28.04	185.7
Ours-FG-MRA-FT	0.922	0.065	21.76	56.55	27.88	181.7
Naive+RefNet	0.922	0.064	22.47	49.18	24.64	183.1
Ours-MRA-FT	0.924	0.061	22.58	48.99	21.02	170.7
Ours-FG	0.921	0.068	21.94	53.17	28.13	169.4
Ours-MRA	0.928	<u>0.055</u>	<u>23.68</u>	47.41	13.64	149.8
Ours-RIA	<u>0.932</u>	<u>0.055</u>	23.59	<u>46.42</u>	<u>12.70</u>	<u>144.2</u>
Ours-FT	0.925	0.057	23.39	48.68	19.31	166.0
Ours-FT + Joint Training	0.923	0.059	23.54	48.32	20.14	148.7
Ours-FT + Full FT	0.928	0.057	23.66	47.39	18.95	174.2
Ours	0.937	0.052	23.73	45.33	12.36	138.7

Table 3. **Quantitative Comparisons on the TikTok Dataset.** All methods are evaluated without face refinement as post-processing. For each metric, the best and second-best methods are highlighted in bold and underline, respectively. Our method outperforms all tested baselines across all metrics. The ablation variant, *Ours-RIA*, achieves results comparable to *Ours* on this dataset because the input front and back images share the same background as the ground truth (GT) frames, making reference image augmentation less necessary in this case.

Method	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	FID \downarrow	FVD-VID \downarrow	FVD \downarrow
Animate Anyone [6]	0.779	0.236	18.78	81.02	44.55	551.9
Champ [22]	0.774	0.243	18.24	90.89	53.81	669.2
StableAnimator [17]	0.784	0.242	18.48	90.43	46.20	473.4
MimicMotion [21]	0.787	0.235	18.67	87.22	38.14	433.5
Ours-FG-MRA-FT	0.790	0.234	18.64	87.36	37.54	435.6
Naive+RefNet	0.795	0.226	18.72	87.11	37.15	433.9
Ours-MRA-FT	0.802	0.224	18.98	85.19	37.22	436.3
Ours-FG	0.799	0.229	18.39	81.96	36.11	399.8
Ours-MRA	0.805	0.217	19.42	78.60	32.35	387.5
Ours-RIA	0.808	<u>0.216</u>	19.71	<u>76.70</u>	30.85	<u>384.5</u>
Ours-FT	0.803	0.221	19.33	83.58	35.77	390.8
Ours-FT + Joint Training	0.794	0.221	19.13	79.48	34.21	390.2
Ours-FT + Full FT	0.801	0.219	19.51	78.79	33.85	428.5
Ours	<u>0.807</u>	0.215	<u>19.65</u>	76.65	<u>31.21</u>	382.1

and frame quality.

3.3. Body Size, Garment Accuracy, Realism

Tab. 4 reports the full results of our human study, which evaluates body size, garment accuracy, and overall realism. Our method consistently outperforms all baselines across all criteria.

Tab. 5 presents the corresponding automatic evaluation results for body size, garment accuracy, and realism. Detailed descriptions and discussions are provided below.

For quantitative body size evaluation, we used the SMPL-based estimator CLIFF [10] to estimate shape pa-

Table 4. **Results of Human Study.** Our method outperforms all baselines.

Method	Body Shape Accuracy \uparrow	Garment Accuracy \uparrow	Realism \uparrow
Animate Anyone [6]	2.55	2.45	2.03
Champ [22]	3.02	2.88	2.32
StableAnimator [17]	3.10	2.18	2.76
MimicMotion [21]	3.20	2.05	2.82
Ours	3.88	4.08	3.75

Table 5. **Results of Quantitative Evaluation of Body Size, Garment Accuracy, and Realism.** Our method outperforms all baselines.

Method	Shape Parameter Difference \downarrow	Garment Region IoU \uparrow	Garment Appearance LPIPS \downarrow	VLM Realism Score \uparrow
Animate Anyone [6]	0.073	0.853	0.539	5.8
Champ [22]	0.075	0.861	0.512	5.6
StableAnimator [17]	0.071	0.884	0.616	6.2
MimicMotion [21]	0.062	0.895	0.601	6.7
Ours	0.053	0.923	0.485	7.5

rameters on predicted and real frames from our self-captured dataset. We calculated shape difference by averaging the absolute differences between shape parameters of predicted and real frames. Our method achieved a shape difference of 0.053, outperforming the best baseline, MimicMotion (0.062), by 17%.

We evaluated garment accuracy via region alignment and appearance similarity. For region alignment, we used SAM2 [15] to segment garment regions in predicted and ground-truth frames from self-captured dataset and computed the average IoU. Our method achieved 0.923, better than the best baseline, MimicMotion (0.895). For appearance similarity, we calculated LPIPS within the segmented garment regions, with our method scoring 0.485, outperforming the best baseline, Champ (0.512).

To evaluate realism, we used a VLM [1] to rate generated videos on a 1–10 scale (higher is better). Our method scored 7.5, outperforming the best baseline, MimicMotion (6.7).

3.4. Comparison to Models Trained on Other Datasets

We further compare our method with models trained on different datasets: Veo 3.1 (References to Video) [3] and Phantom (built on Wan2.1 [18]) [11]. These models take one or multiple reference images and a text prompt as input to generate a video. In our experiments, we provide three reference images – the front selfie, back selfie, and background image – along with a text description of motion to generate fit-check videos.

We use the following text prompt for Veo 3.1 (References to Video): “Generate a fit-check video. The person [specify the motion]. The person’s appearance must remain consistent with the first and second images (front and back selfies), and the background should match the third image.

The camera should remain static and the viewpoint must not change. ”

We additionally include a background description in the text input for Phantom, as it tends to ignore the provided background image. The text prompt is: *“Generate a fit-check video. The person [specify the motion]. The person’s appearance must remain consistent with the first and second images (front and back selfies), and the background should match the third image. The background is [description of background]. The camera should remain static and the viewpoint must not change. ”*

Note that, unlike our method, which takes a motion sequence as an explicit input, these models rely solely on text to determine motion. Therefore, their generated motions may differ from ours.

Fig. 11, Fig. 12, and Fig. 13 show qualitative comparisons. Both Veo 3.1 (References to Video) and Phantom only support landscape-mode video generation. Phantom generates inaccurate outfit (see Fig. 11). It also struggles to maintain consistency with the provided background image, often leading to incorrect scene composition. Veo 3.1 may produce inaccurate visual details (see the red arrow in Fig. 11 and Fig. 12).

3.5. Ablation Study

3.5.1. Qualitative Comparison

Fig. 14 shows additional comparisons of our variants with the full method. We observe the following: (1) *Ours-FG-MRA-FT (naïve)* fails to render accurate back views, indicating that simple modifications do not effectively encode features from additional reference images. (2) *Naïve+RefNet* incorporates ReferenceNet, improving back views but introducing artifacts (rows 1 to 2) and failing in the case of a white background (rows 3 to 4). This demonstrates that using ReferenceNet reduces the model’s generalization ability to unseen backgrounds. Additionally, it introduces extra parameters for training, which negatively impacts training efficiency. (3) *Ours-MRA-FT (Naïve + FG)* applies our frame generation strategy, producing better back views than naive methods, but still suffers from blurry patterns. (4) *Ours-FT (Naïve + FG + MRA)* further integrates multi-reference attention, enhancing back view patterns.

Importantly, all the variants fail to produce realistic reflections or generate weaker, blurry reflections on the ground (rows 1 and 2). In contrast, our method effectively achieves this by leveraging a fine-tuning strategy specifically designed to enhance shadows and reflections. In summary, our full model produces sharper results with more accurate patterns while effectively generating realistic shadows and reflections.

Table 6. **Video quality under different numbers of input views.** Our method generalizes better than MimicMotion.

Method	Two Selfies		Front-only		Back-only	
	LPIPS	FVD	LPIPS	FVD	LPIPS	FVD
MimicMotion	0.413	947.8	0.435	951.5	0.452	947.1
Ours	0.381	854.9	0.395	859.2	0.411	847.5

3.5.2. Quantitative Comparison

Tab. 1, 2, and 3 present the quantitative results of our model compared to its variants across the three datasets. As discussed in the main paper, our method outperforms most variants across all metrics by employing a novel frame generation strategy, a multi-reference attention, and a fine-tuning approach, resulting in improved appearance fidelity and temporal consistency.

The ablation variant, *Ours-RIA*, achieves results comparable to *Ours* on this dataset because the input front and back images share the same background as the ground truth (GT) frames, making data augmentation less essential in this scenario.

3.5.3. Number of Input Views

Tab. 6 reports video quality under different numbers of input views (same dataset as Table 1, main paper). Our method supports a variable number of views and consistently outperforms MimicMotion, with more views improving rendering accuracy.

3.5.4. Inaccurate Segmentation

Our method relies on pre-segmentation of the input selfies. Therefore, we evaluate its robustness to segmentation inaccuracies. In practice, we found the adopted SAM2 [15] robust for segmenting mirror selfies. For study purposes, we dilated and eroded segmentation masks in self-captured dataset by 5% and 10%, feeding these inaccurately segmented selfies to our model. The resulting LPIPS scores – 0.383 (5% dilation), 0.386 (10% dilation), 0.384 (5% erosion), and 0.388 (10% erosion) – were only slightly worse than with accurate masks (0.381).

3.6. Evaluation on Motion Retrieval

3.6.1. Motion Retrieval Accuracy.

We collect a test set by asking five participants to perform fit-check motions (*e.g.*, walks, twirls) while using a phone to record IMU data. Simultaneously, we capture video recordings of these motions with an external camera. From each video, we extract the corresponding SMPL sequence as ground truth (GT), which is later used for evaluation, constructing a dataset of 20 IMU inputs and GT SMPL sequences.

Since there are no existing IMU-to-motion retrieval methods, we compare our approach against TMR [14], a

Table 7. **IMU Robustness.** We report similarity score, showing our method outperforms TMR.

	C.P.	C.W.	N.P.
TMR	0.541	0.521	0.439
Ours	0.743	0.728	0.671

text-to-motion retrieval baseline. To provide input for this baseline, we manually annotate textual descriptions for the motions in our test set. An example text annotation for a motion is: “A person walks away from the camera with their back facing it, then turns left by half a circle.”

We compare the top-k retrieved motions with the GT motion using a pretrained motion encoder [16] to evaluate their similarity. This motion encoder takes as input a sequence of motion parameters in the HumanML3D format [4], which can be obtained from an SMPL sequence (details in [16]), and outputs a motion embedding. For evaluation, we input both the retrieved and GT motions into the encoder and compute their similarity based on the dot product of their motion embeddings. Our method achieves a similarity score of 0.848 for $k = 1$ and 0.716 for $k = 5$, outperforming TMR’s 0.641 ($k = 1$) and 0.522 ($k = 5$). These results demonstrate the effectiveness of our approach, highlighting that IMU-based retrieval provides more fine-grained motion guidance than text-based retrieval.

3.6.2. Motion Retrieval Robustness.

To evaluate robustness to noisy IMU inputs and cross-device generalization, we collect a *new* dataset with 5 participants performing 20 motion sequences. For each motion, we record 3 IMU inputs – clean phone IMU (**C.P.**, phone held steadily), clean watch IMU (**C.W.**, Apple Watch worn on the same hand), and noisy phone IMU (**N.P.**, another phone loosely held in the other hand) – together with synchronized video capture for ground-truth (GT) motion (the GT capture setup follows Sec. 3.6.1). For each IMU input, we perform top-k ($k=5$) motion retrieval using our method and TMR, and compute similarity scores by comparing the retrieved motions with the GT motion (following Sec. 3.6.1).

Tab. 7 reports similarity scores, showing that our method consistently outperforms TMR, is robust to noisy IMU inputs, and generalizes well across devices.



Figure 5. **More Results of Our Method.** The left two columns display the input selfies and background, while the right six columns show the generated results (inset: pose input, adjusted to prevent occlusion). Given mirror selfies with various outfits and lighting conditions, our method produces realistic fit-check videos, accurately capturing appearance from diverse poses. It also generates reflections and shadows on the ground, ensuring seamless integration between the subject and both indoor and outdoor backgrounds.



Figure 6. **Qualitative Comparison with Baselines on Selfie Inputs.** The left two columns display the input selfies, pose, and background, while the right five columns showcase the generated results from various methods. The inset of Champ illustrates its corresponding SMPL pose input. All results are post-processed using face refinement. Our method surpasses all tested baselines by more accurately reconstructing appearance across diverse poses while also producing more realistic shadows and reflections on the floor.

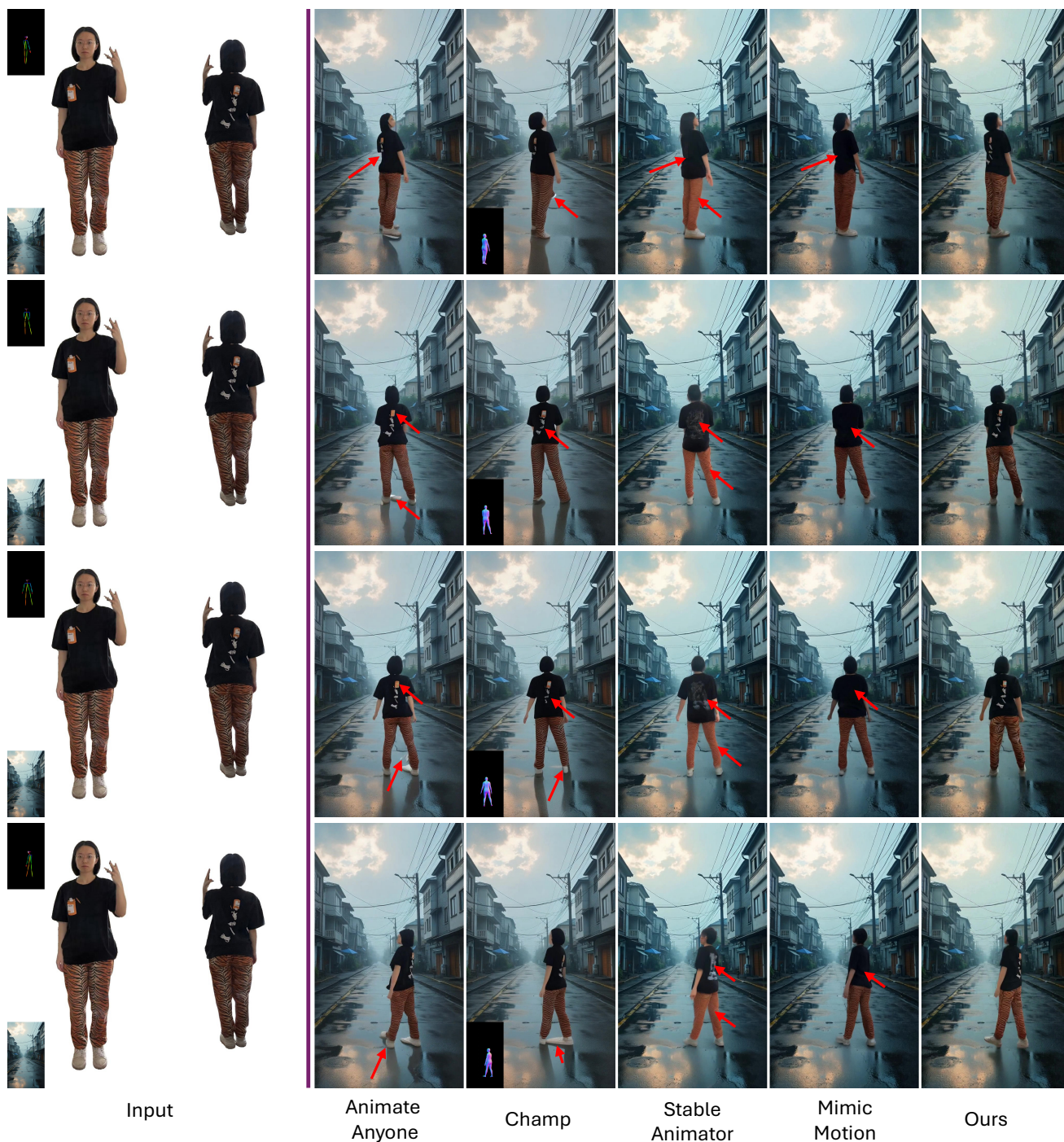


Figure 7. **Qualitative Comparison with Baselines on Selfie Inputs.** The left two columns display the input selfies, pose, and background, while the right five columns showcase the generated results from various methods. The inset of Champ illustrates its corresponding SMPL pose input. All results are post-processed using face refinement. Our method surpasses all tested baselines by more accurately reconstructing appearance across diverse poses while also producing more realistic shadows and reflections on the floor.

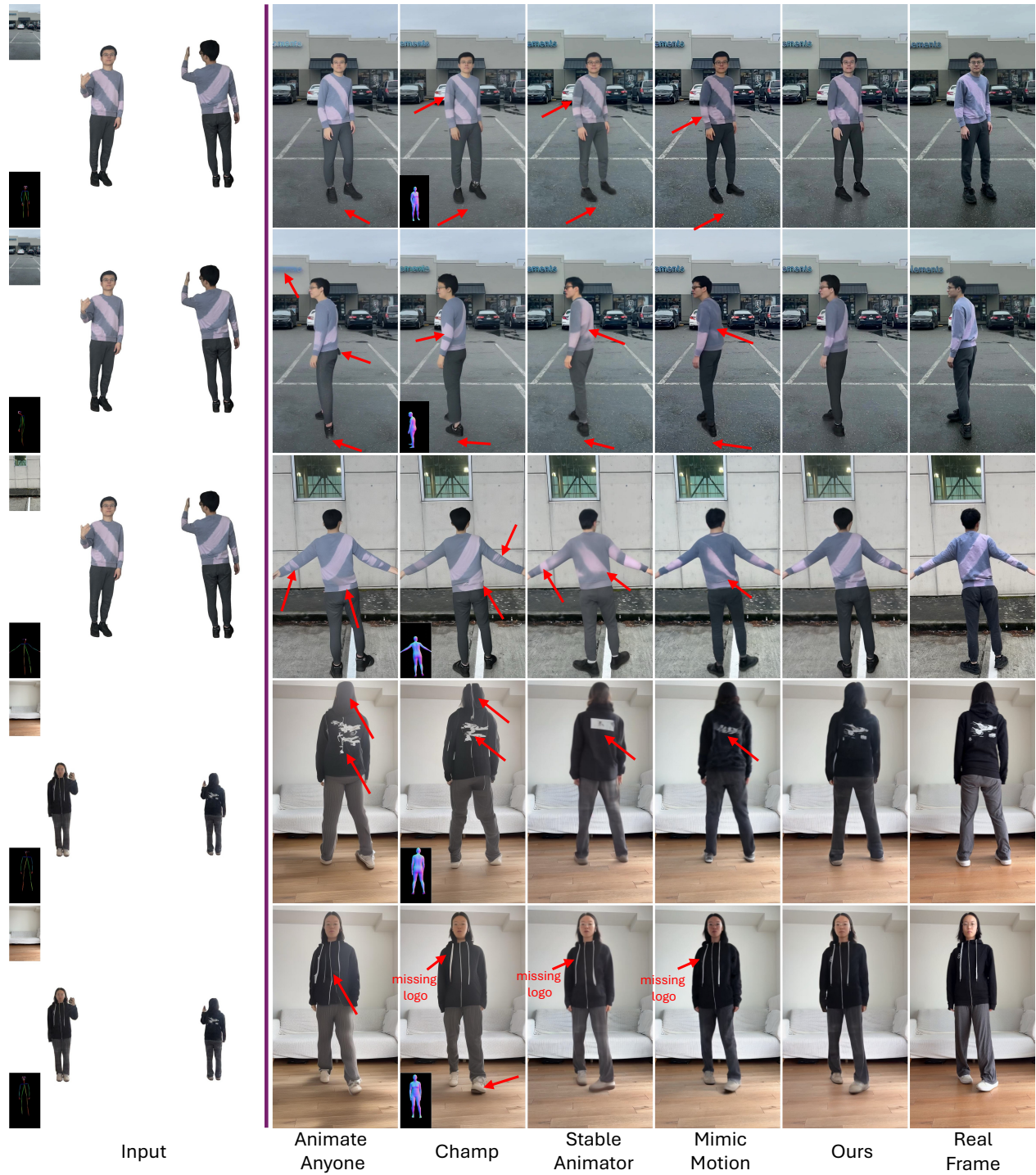


Figure 8. **Qualitative Comparison with Baselines on Self-Captured Dataset.** The left two columns display the input selfies, pose, and background, while the right six columns showcase the generated results from various methods alongside the real frame. The target poses are detected from the real frames. The inset of Champ illustrates its corresponding SMPL pose input. All results are post-processed using face refinement. Our method surpasses all tested baselines by more accurately reconstructing appearance across diverse poses while also producing more realistic shadows and reflections on the floor. Note that background images and real videos, though captured in the same session, may vary in intensity and color tone due to auto-exposure and white balance adjustments.



Figure 9. **Qualitative Comparison with Baselines on UBC Fashion Dataset.** The left two columns present the input reference images, pose, and background, while the right six columns showcase the generated results from various methods alongside the ground truth (GT). The target poses are detected from the GT. The inset of Champ displays its corresponding SMPL pose input. All results are post-processed using face refinement. Note that none of the methods were trained on the UBC Fashion Dataset’s training set. Our method outperforms all tested baselines in accurately reconstructing appearance across a diverse range of poses.

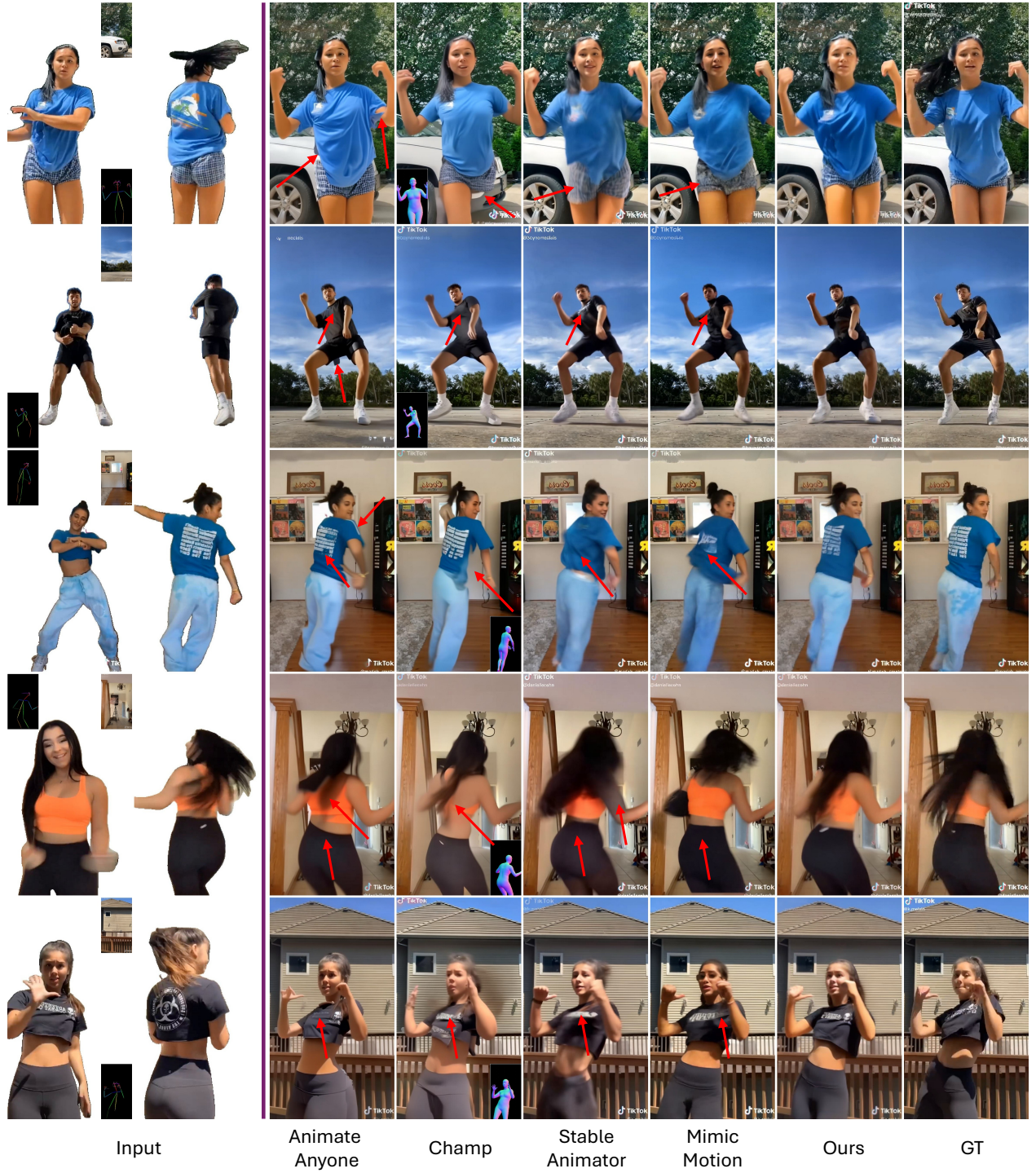


Figure 10. **Qualitative Comparison with Baselines on TikTok Dataset.** The left two columns present the input reference images, pose, and background, while the right six columns showcase the generated results from various methods alongside the ground truth (GT). The target poses are detected from the GT. The inset of Champ displays its corresponding SMPL pose input. Note that none of the methods were trained on the TikTok Dataset. All results are post-processed using face refinement. Our method outperforms all tested baselines in accurately reconstructing appearance across a diverse range of poses. *Disclaimer: In Fig. 2 of the main paper, the example in row 3 of this figure is used solely for illustrative purposes and was not included in the training data.*

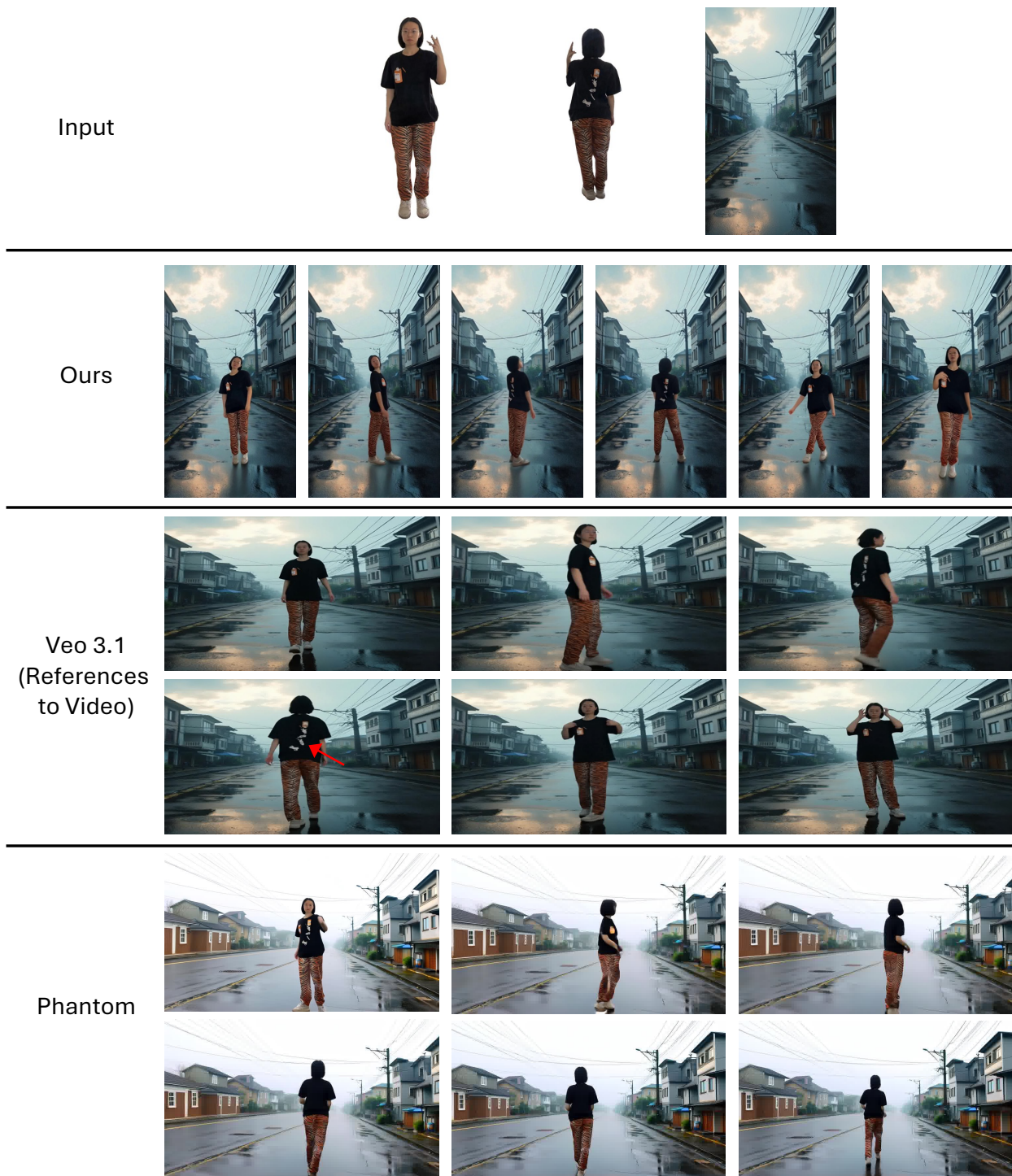


Figure 11. **Qualitative Comparison with Methods Trained on Other Datasets.** The first row shows the input: front selfie, back selfie, and background image. The remaining rows present the outputs from different methods. Both Veo 3.1 (References to Video) and Phantom take these three images as input along with a text description of motion. Also, they only support landscape-mode video generation. In addition, Phantom fails to effectively utilize the input background image, leading to inconsistent scene composition.

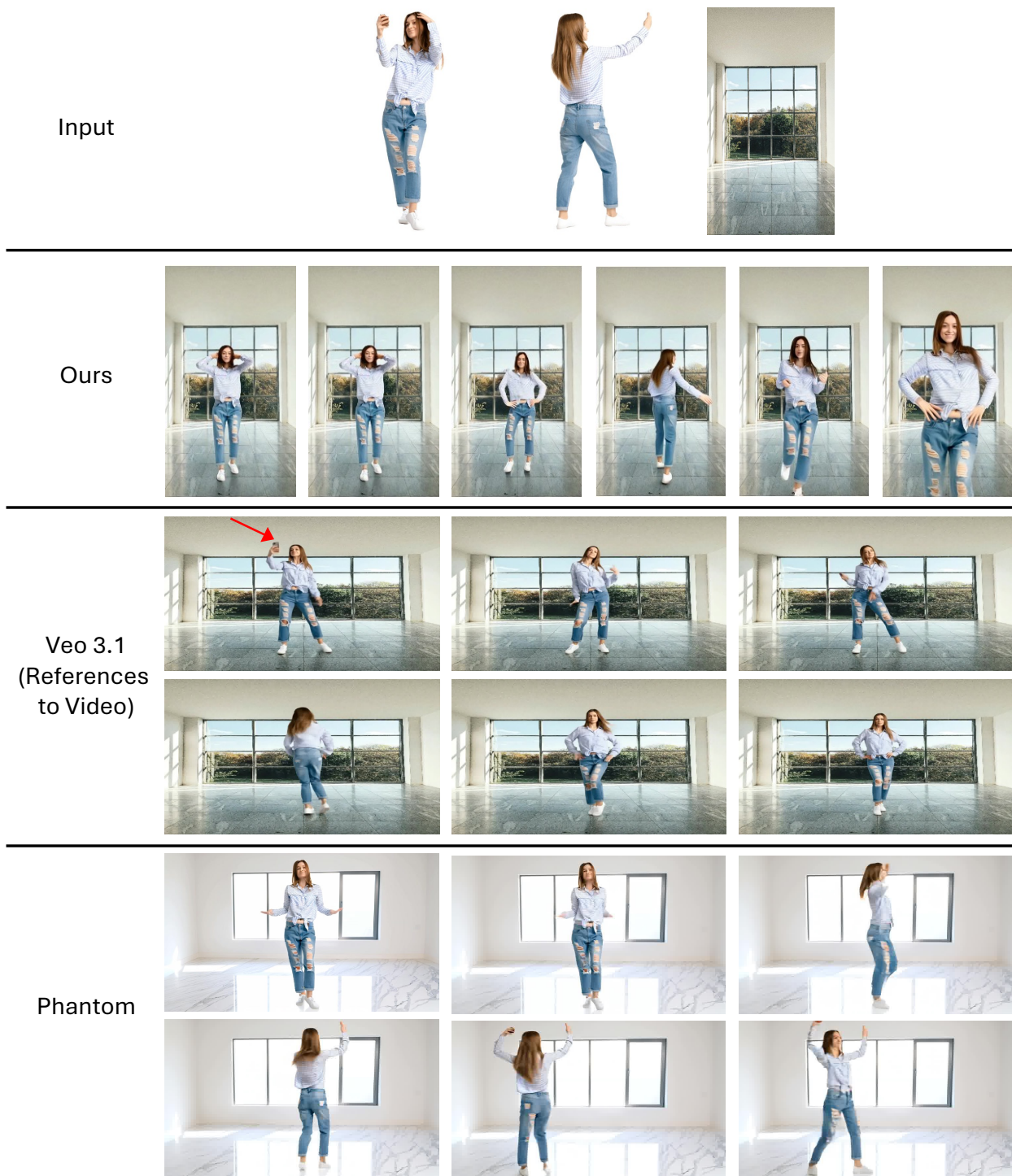


Figure 12. **Qualitative Comparison with Methods Trained on Other Datasets.** The first row shows the input: front selfie, back selfie, and background image. The remaining rows present the outputs from different methods. Both Veo 3.1 (References to Video) and Phantom take these three images as input along with a text description of motion. Also, they only support landscape-mode video generation. In addition, Phantom fails to effectively utilize the input background image, leading to inconsistent scene composition.

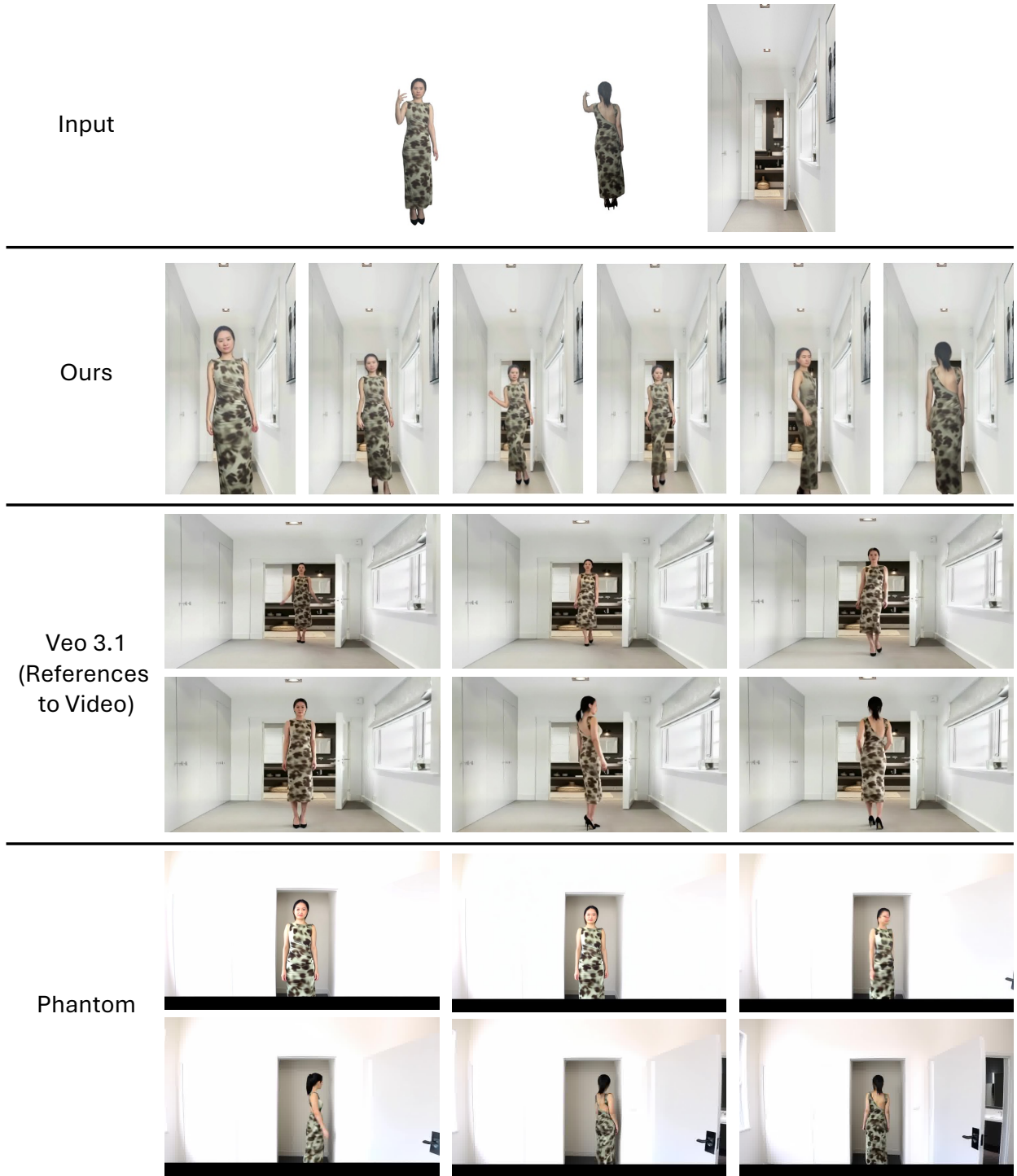


Figure 13. **Qualitative Comparison with Methods Trained on Other Datasets.** The first row shows the input: front selfie, back selfie, and background image. The remaining rows present the outputs from different methods. Both Veo 3.1 (References to Video) and Phantom take these three images as input along with a text description of motion. Also, they only support landscape-mode video generation. In addition, Phantom fails to effectively utilize the input background image, leading to inconsistent scene composition.



Figure 14. **Model Ablations.** The inputs are shown in the left two columns. The right five columns showcase the generated results from various variants and our full model. All results are post-processed using face refinement. *Ours-FG-MRA-FT (naïve)* fails to render accurate back views. *Naive+RefNet* incorporates ReferenceNet, improving back views but creates artifacts (row 1 to 2) and fails in the case of a white background (row 3 to 4). *Ours-MRA-FT (Naïve + FG)* uses our frame generation strategy and produces better back views than naïve methods. *Ours-FT (Naïve + FG + MRA)* additionally uses multi-reference attention and enhances back view patterns. Importantly, all the variants fail to produce realistic reflections or generate weaker, blurry reflections on the ground (rows 1 and 2), whereas our method successfully achieves this. In summary, our full model delivers sharper results with more accurate patterns, along with reasonable shadow and reflection generation.

References

- [1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhao-hai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5
- [2] Jianqi Chen, Yilan Zhang, Zhengxia Zou, Keyan Chen, and Zhenwei Shi. Dense pixel-to-pixel harmonization via continuous image representation. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2023. 2
- [3] Google DeepMind. Veo 3.1 — video meets audio: advanced creative controls. <https://deepmind.google/models/veo/>, 2025. Accessed: YYYY-MM-DD. 5
- [4] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, 2022. 3, 7
- [5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2
- [6] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117*, 2023. 3, 4, 5
- [7] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12753–12762, 2021. 3
- [8] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. *arXiv*, 2022. 1
- [9] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 3
- [10] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022. 1, 5
- [11] Lijie Liu, Tianxiang Ma, Bingchuan Li, Zhuowei Chen, Jiawei Liu, Gen Li, Siyu Zhou, Qian He, and Xinglong Wu. Phantom: Subject-consistent video generation via cross-modal alignment. *arXiv preprint arXiv:2502.11079*, 2025. 5
- [12] MooreThreads. Moore animate anyone. <https://github.com/MooreThreads/Moore-AnimateAnyone>, 2024. 3
- [13] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 1
- [14] Mathis Petrovich, Michael J. Black, and Gül Varol. TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis. In *International Conference on Computer Vision (ICCV)*, 2023. 3, 6
- [15] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 5, 6
- [16] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 7
- [17] Shuyuan Tu, Zhen Xing, Xintong Han, Zhi-Qi Cheng, Qi Dai, Chong Luo, and Zuxuan Wu. Stableanimator: High-quality identity-preserving human image animation. *arXiv preprint arXiv:2411.17697*, 2024. 3, 4, 5
- [18] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Fei Wu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wenten Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 5
- [19] 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, 2024. 1
- [20] Polina Zablotskaia, Aliaksandr Siarohin, Bo Zhao, and Leonid Sigal. Dwnet: Dense warp-based network for pose-guided human video generation. *arXiv preprint arXiv:1910.09139*, 2019. 3
- [21] Yuang Zhang, Jiayi Gu, Li-Wen Wang, Han Wang, Junqi Cheng, Yuefeng Zhu, and Fangyuan Zou. Mimicmotion: High-quality human motion video generation with confidence-aware pose guidance. *arXiv preprint arXiv:2406.19680*, 2024. 2, 3, 4, 5
- [22] Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Zilong Dong, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. Champ: Controllable and consistent human image animation with 3d parametric guidance. In *European Conference on Computer Vision*, pages 145–162. Springer, 2024. 3, 4, 5