PAM: A Pose–Appearance–Motion Engine for Sim-to-Real HOI Video Generation

1. Implementation Details

Resource Usage	Stage-I	Stage-II	Stage-III
Memory (GB)	0.03	41.4	30.3
Time (s)	19.3	36.1	245.7

Table 1. Resource Usage across Different Stages

Training Details: Our model was trained on a setup consisting of 8 x NVIDIA 800 GPUs, with a batch size of 4 x 8 and a learning rate of 1×10^{-4}. The training process involved 8,000 training steps, using the AdamW optimizer and the DeepSpeed training architecture [3].

Evaluation Details: For the evaluation of video generation, we sample 1,600 videos, each consisting of 49 frames, from the test set. For the evaluation of Mean Per Joint Position Error (MPJPE), we utilize Hamer [2] to estimate the hand joints in the generated videos, and compute the loss by comparing the estimated joint positions with the ground truth hand joints. To assess the performance on downstream tasks, we train the SimpleHand model for 200 epochs using its official implementation.

Downstream Validation Details: For the downstream task, given the input, we first leverage the appearance generator (the controllable image diffusion model) to randomly sample 30 candidates. Subsequently, we filter out samples with low-accuracy hand poses using Hamer. Specifically, we predict the hand keypoints using Hamer for every frame, compare them with the ground truth, and discard the bottom 25% of the generated videos based on pose accuracy.

2. Additional Results

2.1. Compute and Memory Benchmarking

As shown in Table 1, we report per-stage resource usage on an NVIDIA H20 GPU. Stage I is lightweight, Stage II has the highest peak memory (41.4 GB), and Stage III is the slowest (245.7 s) due to diffusion computation. Overall, the full pipeline runs in **301.1 s** for 40 frames.

2.2. Ablation on Masking Probability and Input Condition Quality

We evaluate the robustness of our model by introducing random Gaussian noise into the input conditions. This noise is added to simulate real-world perturbations and assess how well the model can maintain performance despite such distortions. The results, as shown in Table 2, highlight a noticeable performance degradation when no masking is applied — the model's performance metrics decrease significantly when exposed to noise.

Table 2. Ablation Study on Masking Probability and Input Condition Quality

Settings	FVD (↓)	PSNR (↑)	MPJPE (↓)
0 Mask Prob + Clean Cond	28.56	30.99	19.01
0 Mask Prob + Noisy Cond	34.58	27.11	23.67
0.2 Mask Prob + Clean Cond (**Ours**)	**29.13**	**30.17**	**19.37**
0.2 Mask Prob + Noisy Cond	30.45	29.67	20.31

However, when masking is applied to the input conditions, the performance drops are considerably reduced. This observation strongly suggests that the random masking technique enhances the model's robustness. By masking portions of the input, the model appears to be less sensitive to noise, likely because it focuses on more stable, less noisy features of the data. Therefore, the use of random masking improves the model's generalization ability and resilience to external noise, thus supporting the hypothesis that masking is an effective strategy for improving robustness in challenging conditions.

2.3. Ablation on Appearance Generation Method

To evaluate the sensitivity of our method to Stage-I trajectories, we substituted GraspXL with D-Grasp. As shown in Table 3, GraspXL achieves superior performance, demonstrating that the final generation quality relies on the quality of Stage-I pose sequence.

Table 3. Ablation on Stage-I Method

Method (Stage-I)	FVD (↓)	MF (↑)	MPJPE (↓)
D-Grasp	58.17	0.599	36.18
GraspXL (**Ours**)	**49.98**	**0.645**	**30.96**

Table 4. Ablation on Hand Representation

Hand Representation	FVD (↓)	PSNR (↑)	MPJPE (↓)
Mesh Projection	29.33	**30.17**	36.18
Keypoints (**Ours**)	**29.13**	30.05	**30.96**

[0*]Equal Contribution. [†]Corresponding Author.

2.4. Ablation on Hand Representations

We ablate hand keypoints versus 2D hand mesh projections as conditioning. Table 4 shows that keypoints perform better on most metrics, especially on MPJPE. This is possibly because mesh projections are more prone to self-occlusion.

2.5. Ablation on Hand Encoder

Table 5. Ablation on Hand Encoder

Encoder Type	FVD (↓)	PSNR (↑)	MPJPE (↓)
MLP	31.59	30.07	21.96
VAE (**Ours**)	**29.13**	**30.17**	**19.37**

To validate the VAE-based condition encoder, we evaluate it on 1,000 sampled keypoint images and obtain a PSNR of 40.58, indicating a low reconstruction error for the conditioning images. As shown in Table 5, the VAE outperforms an MLP-based 2D coordinate encoder across all metrics. This superior performance is primarily attributed to the VAE's enhanced ability to preserve local spatial information.

2.6. Ablation on I2V Backbone

Table 6. Ablation on Backbone

Encoder Type	FVD (↓)	PSNR (↑)	MPJPE (↓)
InterDyn (SVD w/ single cond)	38.83	24.86	28.15
SVD w/ multi conds	<u>34.91</u>	<u>25.84</u>	<u>25.11</u>
Ours (CogVideo w/ multi conds)	**29.13**	**30.17**	**19.37**

To isolate the impact of our trimodal conditioning, we present a controlled ablation in Table 6. Notably, even when employing a comparatively weaker backbone, the multi-conditioned SVD [1] baseline consistently outperforms the single-condition InterDyn across all metrics. This comparison demonstrates that integrating multiple conditions significantly enhances the overall generation performance.

2.7. Investigation in Error Propagation

Experiments in Figure 1 show that Stage-I geometric errors (e.g., interpenetration or missing contact) can propagate, leading to physically implausible interactions even if the generated video appears photorealistic. Furthermore, we observe that Stage-III quality heavily relies on the appearance guidance from Stage II: low-quality initial reference frames degrade the final textures and exacerbate temporal flickering.

2.8. Additional Qualitative Results

We provide more qualitative results in Figure 2 and Figure 3 for DexYCB dataset, Figure 4 and Figure 5 for OANINK2 dataset.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[2] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836, 2024. 1

[3] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020. 1

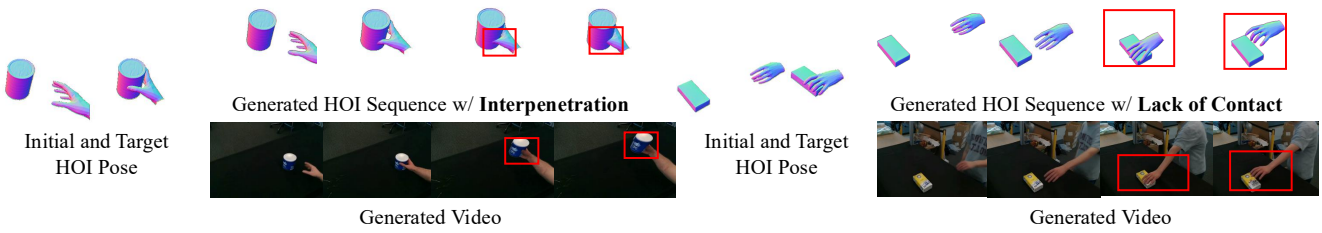

Figure 1. More qualitative results on DexYCB dataset (a).

Figure 2. More qualitative results on DexYCB dataset (a).

Figure 3. More qualitative results on DexYCB dataset (b).

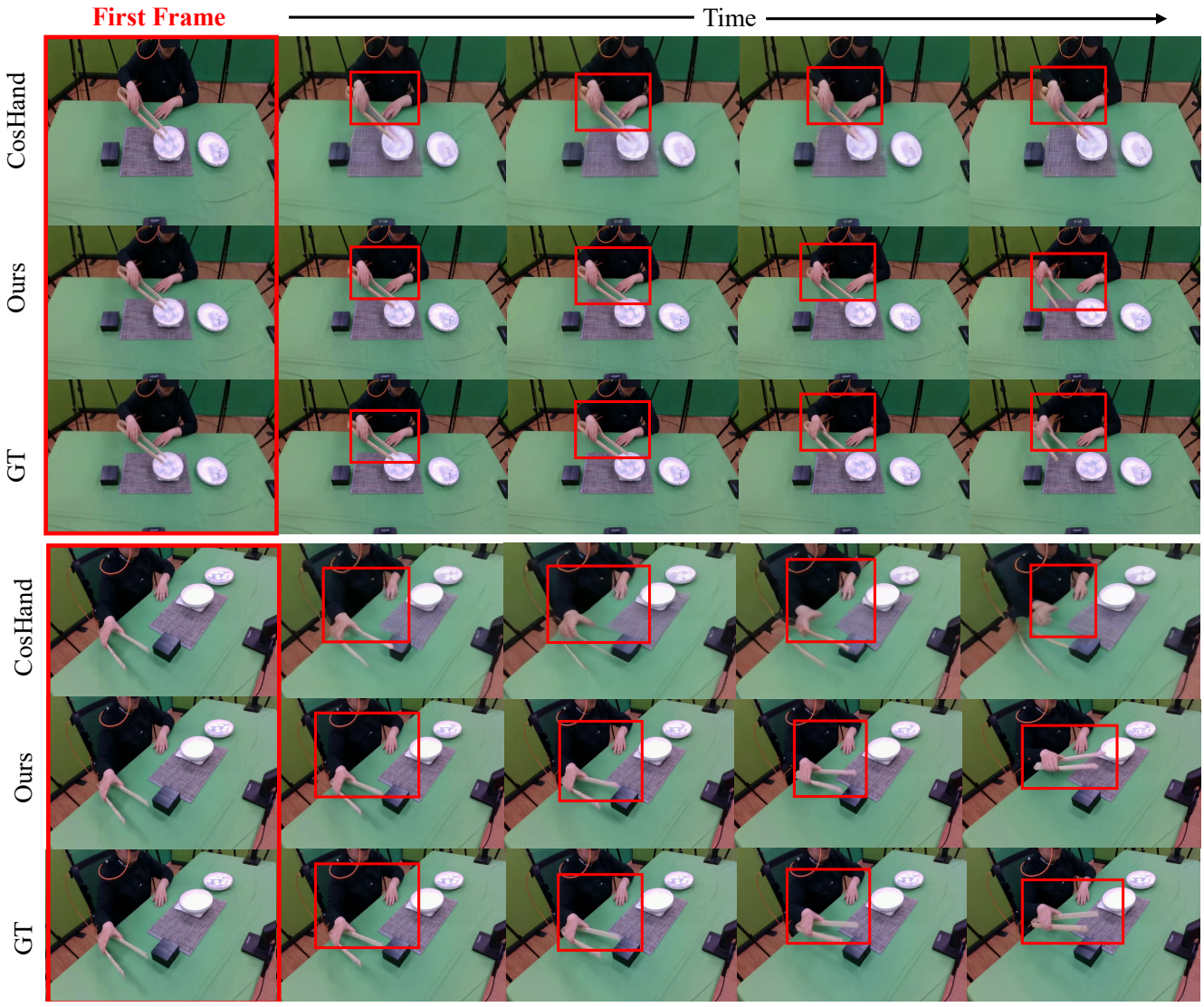

Figure 4. More qualitative results on OAKINK2 dataset (a).

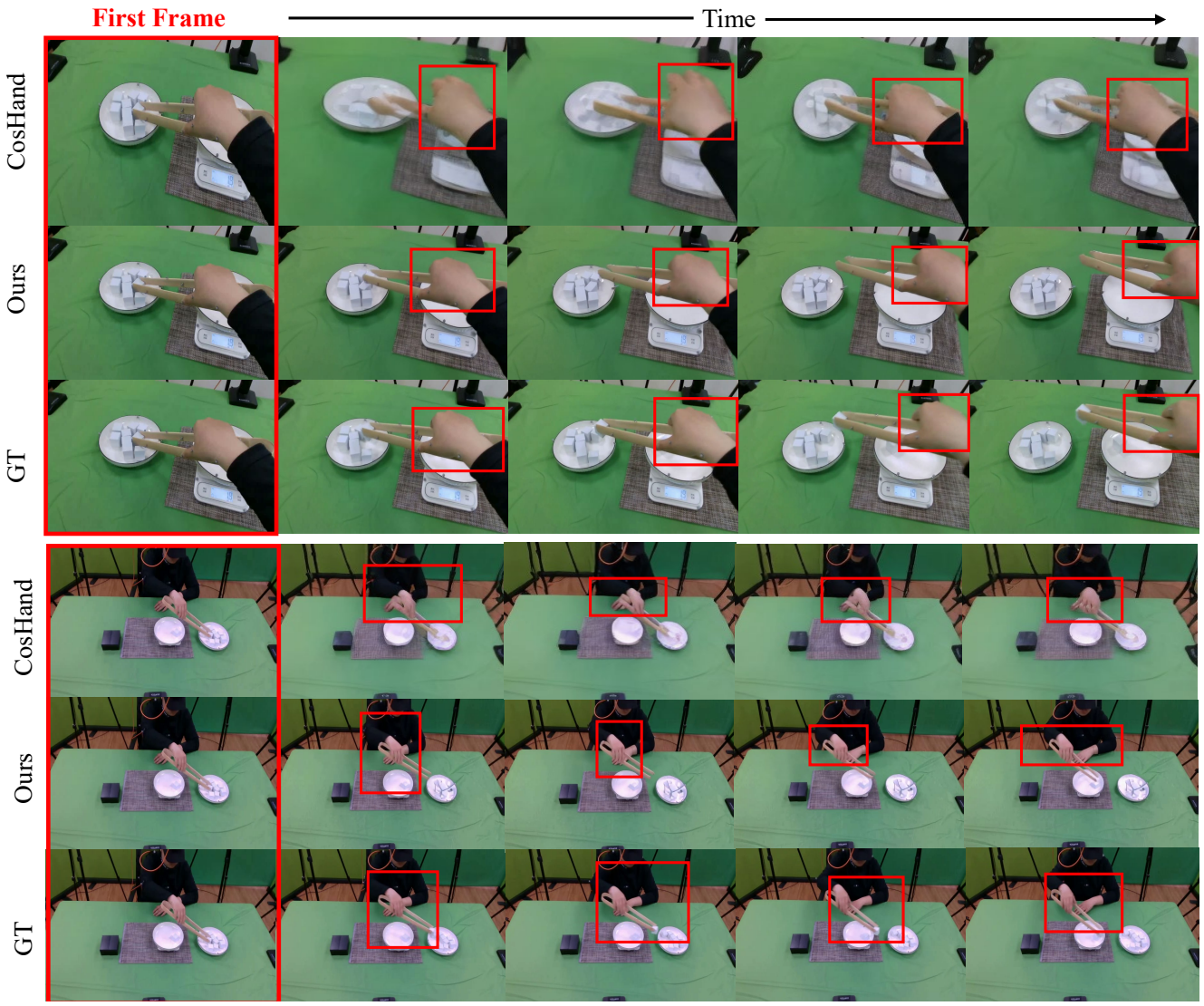

Figure 5. More qualitative results on OAKINK2 dataset (b).