Lighting-grounded Video Generation with Renderer-based Agent Reasoning

Supplementary Material

Ziqi Cai[1,2,4] Taoyu Yang[1,2] Zheng Chang[5] Si Li[5] Han Jiang[4] Shuchen Weng[3,1] Boxin Shi[1,2,*]

[1]State Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
[2]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University
[3]Beijing Academy of Artificial Intelligence [4]OpenBayes Information Technology Co., Ltd.
[5]School of Artificial Intelligence, Beijing University of Posts and Telecommunications

{czq,yangty1031}@stu.pku.edu.cn, {zhengchang98,lisi}@bupt.edu.cn,
hahn@openbayes.com, {shuchenweng,shiboxin}@pku.edu.cn

A. Additional Results

Due to space limitations in the main paper, we present additional experimental results in this section, including dynamic subject generation, image-to-video generation, and video-to-video generation. These results further demonstrate the controllability and robustness of our method.

A.1. Dynamic Subject Generation

Since most of our comparison methods [4, 6] are primarily evaluated on static scenes, we present static scenes in the main paper for fairness and direct comparison. However, our method is not limited to these scene types. Our renderer-based agent can render dynamic subjects, therefore enabling our model to generate dynamic scenes with physically realistic results. We present these cases in Fig. A.

A.2. Image-to-Video Generation

Our LiVER model inherently supports image-to-video generation. After reconstructing 3D meshes from a single input image, our renderer-based agent plans the camera trajectory and lighting conditions according to a user-provided text description, thereby constructing a lighting-grounded scene proxy that guides the video synthesis process. We demonstrate this capability in Fig. B, showcasing the same scene layout under two different camera trajectories and three user prompts. The top three rows correspond to the first camera trajectory, with each row reflecting a distinct user prompt; the bottom three rows show results from an alternative camera trajectory.

A.3. Video-to-Video Generation

Our LiVER model also supports video-to-video generation. We can fully extract the lighting-aware scene proxy from the reference video, and synthesize the edited video based on this proxy and user-provided text descriptions. We present this application in Fig. C, where the lighting, scene layout, and camera trajectory are well preserved.

A.4. Failure Cases

We present failure cases in Fig. D to highlight potential areas for improvement. In the first row, our model correctly renders the scene layout and follows the camera trajectory. However, the car is incorrectly depicted as having two rear ends and no discernible front. In the second row, our model fails to correctly render the details of the table tennis set. These issues can primarily be attributed to the semantic limitations of the underlying video generation backbone. We expect that further scaling up the model capacity would mitigate such errors.

B. Dataset Details

B.1. Dataset Sample Visualization

As illustrated in Sec. 3, we collect **LiVERSet** to facilitate model training and evaluation. We separately showcase three randomly selected samples from the real-world subset **LiVER-Real** and the synthetic subset **LiVER-Syn** in Fig. E to demonstrate the diversity and scope of our data.

B.2. Dataset Caption Annotation

We utilize Qwen2.5-VL-32B-Instruct [1] to generate captions from full video sequences. The model is prompted to produce concise and semantically rich paragraphs detailing scene content and object interactions. Notably, we explicitly exclude camera motion and lighting information to ensure the captions focus mainly on scene semantics, decoupling them from the proxy's physical control signals. These captions serve as text descriptions for the video backbone. The specific prompt is detailed below:

```
Task: You are a video caption generator.
Produce one concise paragraph caption suitable
 for a video-generation model (wan2.2).
Rules:
- You should NOT describe camera movements or
angles.
```

Figure A. Additional qualitative results for dynamic subject generation.

Figure B. Additional qualitative results for image-to-video generation.

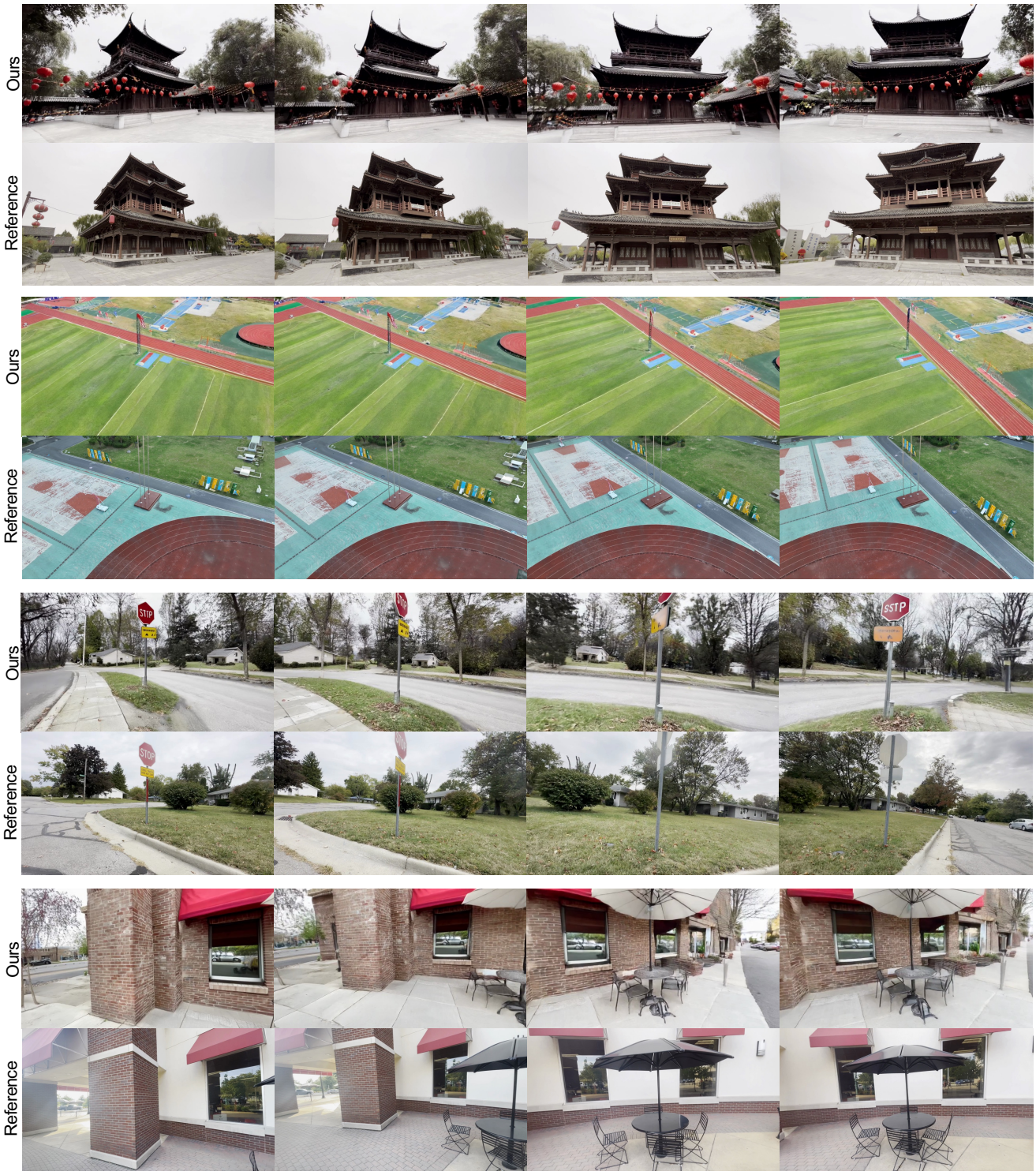

Figure C. Additional qualitative results for video-to-video generation.

Figure D. Visualization of failure cases, primarily attributed to the video backbone's semantic limitation.

```
- You should NOT include any lighting
information.
Constraints:
- Keep the caption concise and actionable for
a video generator (around 1 paragraph, up to
~100 words).
- Output exactly one paragraph (no extra lines
).
Generate caption:
```

C. Experiment Details

C.1. Baselines

In this section, we detail the configurations for all baseline methods. To ensure a fair comparison, all models are evaluated using the same prompts, scene specifications, and ground-truth camera trajectories unless otherwise noted.

CameraCtrl [4] enables explicit 6-DoF camera pose control by accepting per-frame extrinsics as input. As the standard model is limited to generating 16-frame sequences, we restrict our quantitative comparison to the first 16 frames of the ground-truth trajectories to match its sequence length.

MotionCtrl [13] accepts 6-DoF camera extrinsics and controls object motion via sparse trajectories. Similar to CameraCtrl, this method generates 16-frame videos. We adopt the same evaluation protocol, comparing the model's output against the first 16 frames of our reference data.

VideoFrom3D [6] takes 3D geometry, a camera trajectory, and a style image as input. Unlike the explicit per-frame conditioning of the other baselines, it employs a two-stage pipeline: first generating geometric anchor frames (start, middle, and end) via an image diffusion model, followed by frame interpolation using video diffusion. Following the official implementation, we synthesize the sequence and uniformly sample 81 frames from the 92-frame output to match our evaluation setting. Due to the significant computational cost ($\sim$ 40 min on a H100 GPU) of training a style-specific LoRA for every sample, we evaluate this method on a subset of 20 scenes.

C.2. Evaluation Details

Due to the varying generation capabilities of the baselines, we employ two evaluation protocols. For comparisons involving CameraCtrl and MotionCtrl (limited to 16 frames), we compute metrics on the first 16 frames of the generated and ground-truth sequences. For comparisons involving VideoFrom3D, we use the full 81-frame sequences on a representative 20-video subset. During evaluation, we utilize nine quantitative metrics to evaluate LiVER's performance across six aspects:

Frame Quality (FID). To measure the quality of each video frame, we calculate Fréchet Inception Distance (FID) [5] with standard Inception-V3 features [10]. To ensure a fair

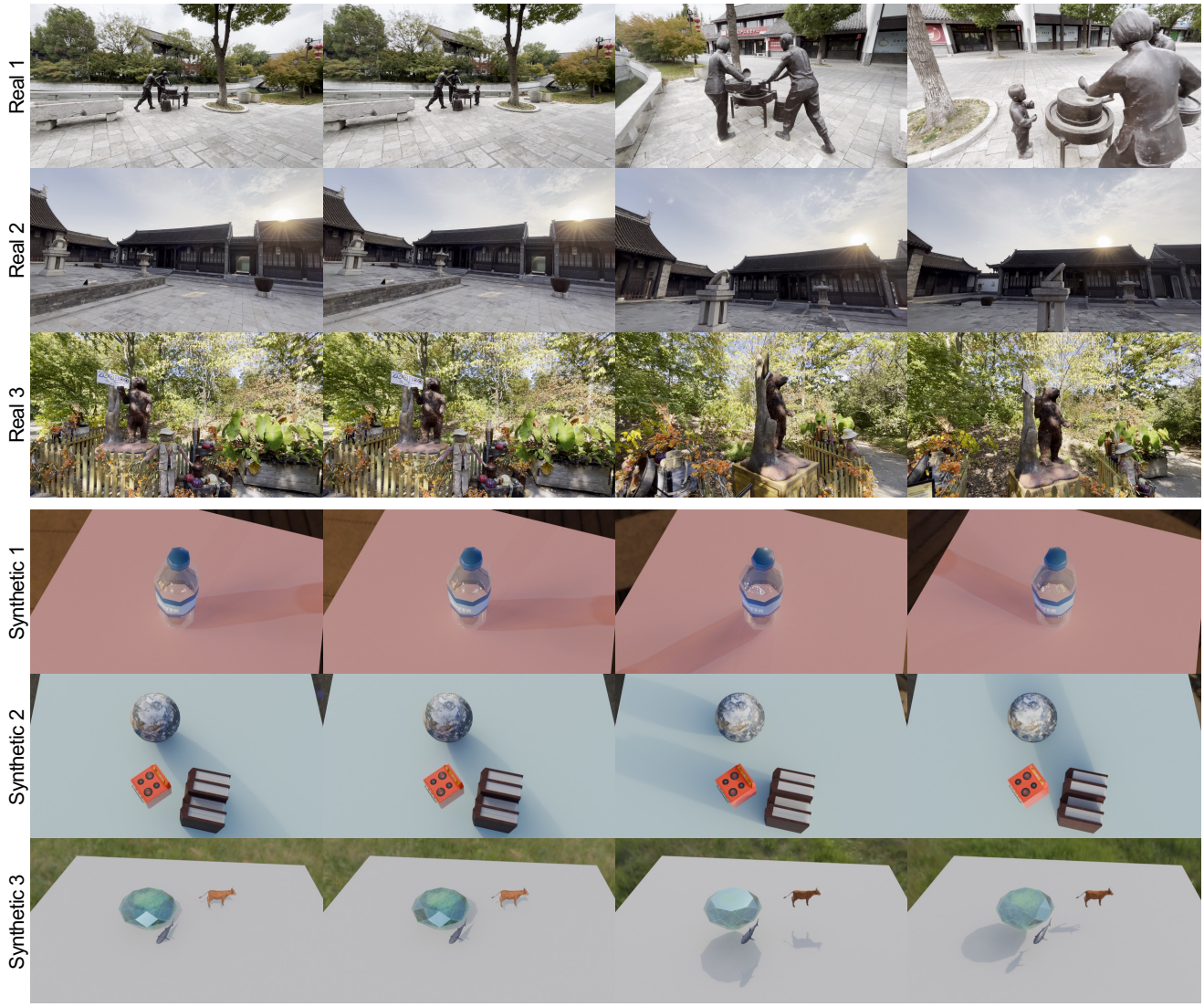

Figure E. Dataset samples. **Top:** Samples from LiVER-Real capture real-world scenes with complex, naturally occurring illumination. **Bottom:** Samples from LiVER-Syn provide physically based, controllable lighting variations.

comparison across methods with differing native resolutions, we resize all frames to 299×299 pixels. These standardized frames are processed via `pytorch-fid` [9] to compute scores on the test subsets.

Video Quality. We evaluate temporal coherence and video realism using Fréchet Video Distance (FVD) [11]. Frames are resized to a 256-pixel short side, center-cropped to 224×224, and normalized using Kinetics statistics. These inputs are fed into a pretrained I3D network [2] to extract features. We report the Fréchet distance between the generated and real video features.

Image–Text Similarity. We assess semantic alignment using OpenCLIP ViT-B/32 [7]. For each video, we uniformly subsample frames and compute the cosine similarity between their embeddings and the corresponding captions. We report the CLIP Score by calculating the average similarity across the dataset.

Trajectory Error. We evaluate camera control precision by estimating trajectories from generated videos using VGGT [12]. The estimated trajectories are aligned to the ground truth via Sim(3) Umeyama alignment on camera centers. We report three error metrics: the root mean squared Absolute Trajectory Error (ATE), the translational Relative Pose Error (RPE_t), and the rotational Relative Pose Error (RPE_r).

Lighting Error. We measure illumination consistency via

a reverse-rendering approach. Using a lighting estimator [3], we derive HDR environment maps from the generated frames and compare them to the ground truth using scale-invariant mean squared error. We report its mean as the overall Lighting Error (LE) and the temporal standard deviation to quantify Lighting Instability (LI).

Layout Accuracy. To quantify spatial alignment with the input guidance, we employ a segmentation model [8] to extract subject masks from the generated videos. We calculate the mean Intersection-over-Union (mIoU) between these predicted masks and the ground-truth instance masks; a higher mIoU indicates superior adherence to the specified scene layout.

D. Organization of Supplementary Video

We provide a supplementary video to dynamically showcase our generation results. The video is structured as follows: *(i)* **Scene proxy rendering.** We visualize the real-world video annotation pipeline to construct the scene proxy. *(ii)* **Representative video results.** We demonstrate the video generation results conditioned on the scene proxy, and present the diverse application scenarios. *(iii)* **Comparison with baselines.** We showcase comparisons with baseline methods [4, 6, 13], followed by ablation studies. *(iv)* **Failure cases.** We finally include failure cases to show potential areas for improvement.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai 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.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[3] Worameth Chinchuthakun, Pakkapon Phongthawee, Nontaphat Sinsunthithet, Amit Raj, Varun Jampani, Pramook Khungurn, and Supasorn Suwajanakorn. DiffusionLight-Turbo: Accelerated light probes for free via single-pass chrome ball inpainting. *arXiv preprint arXiv:2507.01305*, 2025.

[4] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations*, 2025.

[5] 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. In *Advances in Neural Information Processing Systems*, 2017.

[6] Geonung Kim, Janghyeok Han, and Sunghyun Cho. VideoFrom3D: 3D scene video generation via complementary image and video diffusion models. In *ACM SIGGRAPH Asia Conference Papers*, 2025.

[7] 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*, 2021.

[8] 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.

[9] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, 2020. Version 0.3.0.

[10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[11] 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.

[12] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[13] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH Conference Papers*, 2024.