Boosting Camera Motion Control for Video Diffusion Transformers

Soon Yau CHEONG Duygu CEYLAN Armin MUSTAFA
Andrew GILBERT Chun-Hao Paul HUANG
Corresponding email: chunhaoh@adobe.com

SUMMARY

- Camera control methods (MotionCtrl[1], CameraCtrl[2]) work well for U-NET video diffusion models but fail when applied to transformer-based (DiT) models.
- U-Net has smaller embedding dimension in first-and-last layers, making it easier for camera parameter to influence the video generation; transformer have large dimension through out compared to camera embedding, result in diminished camera control.
- We analysed different combinations of camera representations and encoding methods but still witnessed significant degradation in camera control.
- We propose **Camera Motion Guidance (CMG)** - a classifier-free guidance method boosting camera motion by over 400%.

Camera Motion Guidance

Classifier-free guidance (CFG) operates by performing both unconditional and conditional denoising passes, followed by latent-space extrapolation to strengthen alignment with the conditioning input:

$$\hat{e_\theta}(z_t, C_T) = e_\theta(z_t, \emptyset_T) + s_T\{e_\theta(z_t, C_T) - e_\theta(z_t, \emptyset_T)\}$$

where e_θ is denoising model (U-Net or DiT), z_t is noisy latent at time step *t*. We denote C for condition, ϕ for null condition, and subscript T and C for text and camera feature respectively; s is scalar scale.

We add a separate guidance term for camera condition in which the null camera condition is static video i.e. repeated still images.

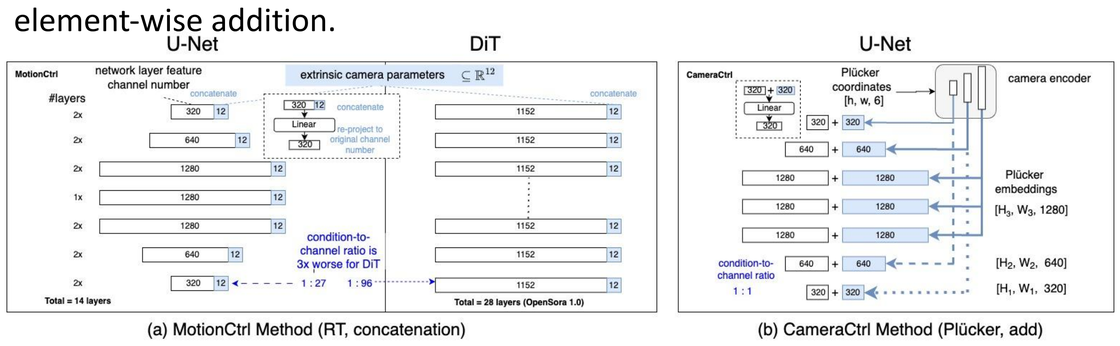
$$\begin{aligned}\hat{e_\theta}(z_t, C_T, C_C) = {} & e_\theta(z_t, \emptyset_T, \emptyset_C) \\ & + s_T\{e_\theta(z_t, C_T, \emptyset_C) - e_\theta(z_t, \emptyset_T, \emptyset_C)\} \\ & + s_C\{e_\theta(z_t, C_T, C_C) - e_\theta(z_t, C_T, \emptyset_C)\}\end{aligned}$$

This allows controlling of camera movement in dependent of other conditioning.

Experiments

Conditioning Methods

We experimented with different architectures : 1) MotionCtrl (Fig 1a) that repeats the camera parameters across spatial dimension and concatenate with model layer features along channel dimension before reprojecting to original dimension; 2) CameraCtrl (Fig 2b) utilises encoder to to project camera parameters into higher dimension matching layer features and perform element-wise addition.



(a) MotionCtrl Method (RT, concatenation) (b) CameraCtrl Method (Plücker, add)

Figure 1. Comparing two conditioning methods. For clarity, only channel dimension is shown, omitting batch, frame, height and width of features.

Camera Representations

Two camera representations were explored: i) extrinsics camera parameters - rotation matrix and translation vector flatten into 12 parameters, and ii) Plucker embedding.

Quantitative Results

1. **DiT architecture causes motion degradation** across different conditioning methods and camera representations. MotionCtrl method results in complete loss of control and CameraCtrl induces very little camera motion.
2. Our method **CMG effectively restores camera control**, achieving lower rotation and translation errors, and simultaneously **enhances camera motion** as evidenced by improved optical flow.
3. **Plucker embedding is better than extrinsic parameters** for camera control.

Model	CCR	Camera	RotErr ↓	TransErr ↓	Motion ↑
(1a) MotionCtrl (U-Net)	1:27	RT	0.168	0.640	7.786
(2a) CameraCtrl (U-Net)	**1: 1**	Plücker	0.176	0.754	9.686
(1b) DiT-MotionCtrl (baseline)	1:96	RT	0.224	0.716	1.485
(1c) DiT-MotionCtrl w CMG (Ours)	1:96	RT	0.208	0.702	1.806
(2b) DiT-CameraCtrl (baseline)	**1: 1**	Plücker	0.186	0.687	1.564
(2c) DiT-CameraCtrl w CMG (Ours)	**1: 1**	Plücker	**0.176**	**0.577**	**6.450**

(a)

Model	Camera	CCR ↑	RotErr ↓	TransErr ↓	Motion ↑
(1b) DiT-MotionCtrl	RT	1:96	0.224	0.716	1.485
(2b) DiT-CameraCtrl	Plücker	**1:1**	0.186	**0.687**	1.564
(2d) DiT-CameraCtrl	RT	**1:1**	**0.177**	0.748	**2.101**
(2c) DiT-CameraCtrl w CMG	Plücker	**1:1**	**0.176**	**0.577**	**6.450**
(2e) DiT-CameraCtrl w CMG	RT	**1:1**	0.177	0.666	5.721

(b)

Table 1: Quantitative results (a) Our method outperform baseline DITs in accuracy and motion. (b) Effects of camera representation. Results for Model 1b, 2b, 2c are repeated for ease of comparison.

Qualitative Results

The vertical lines in Fig. 2 indicate the displacement of the reference point between the first and last frames, illustrating the extent of camera motion. The DiT implementation of CameraCtrl (Fig. 2b) exhibits smaller motion compared to the U-Net implementation (Fig. 2a). In contrast, our CMG method (Fig. 2c) significantly enhances camera motion over the DiT.

Figure 2. Comparing CameraCtrl conditioning method on (a) U-Net (b) DiT and (c) DiT with CMG.

CMG can control the magnitude of camera motion by varying guidance scale.

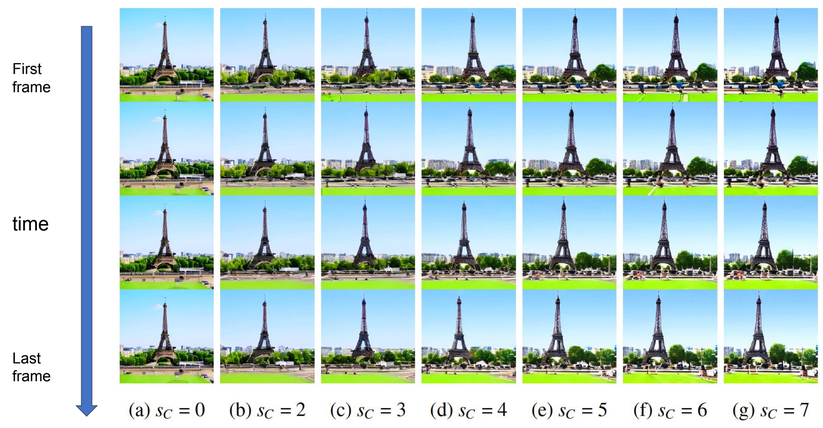
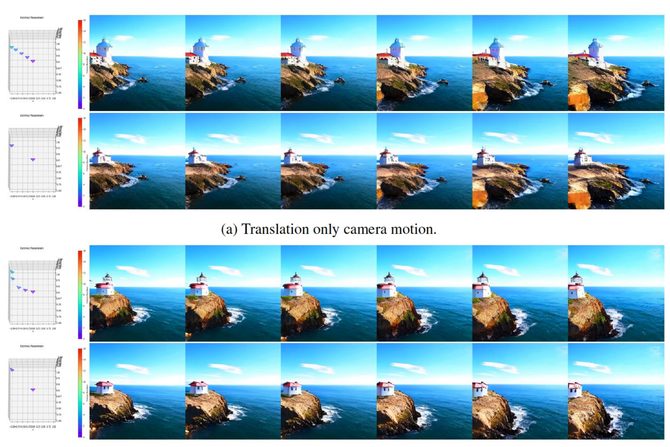


Figure 3. Increasing camera guidance scale s_c increase camera motion.

Sparse Camera Control. With frame dropout augmentation, sparse camera control is achieved—the model can infer and interpolate missing frames when only the last or selected intermediate frames are specified.



(a) Translation only camera motion.

(b) Simultaneous translation and rotation camera motion.

Figure 4. Videos following specified camera trajectories with sparse camera control as shown in the left.

References
1. Wang, Zhouxia, et al. "Motionctrl: A unified and flexible motion controller for video generation." *ACM SIGGRAPH 2024 Conference Papers*. 2024.
2. He, Hao, et al. "Cameractrl: Enabling camera control for video diffusion models." *The Thirteenth International Conference on Learning Representations*. 2025.