OmniVGGT: Omni-Modality Driven Visual Geometry Grounded Transformer

Supplementary Material

A. Appendix

A.1. Dataset Details

We train our model on 19 datasets that contain a diverse range of scene types, including: ARKitScenes [12], BlendedMVS [73], DL3DV [34], Dynamic Replica [23], HyperSim [49], Kubric [15], MapFree [2], MegaDepth [32], Matterport 3D [46], MVS-Synth [20], ScanNet [10], ScanNet++ [74], Spring [39], TartanAir [68], UASOL [5], Unreal 4K [60], Virtual KITTI [7], Waymo [58], WildRGBD [70]. We modified the official DUSt3R [67] dataloader script to adapt it for the VGGT [65] training process. Table 8 summarizes the statistics of the datasets we used. During training, in each epoch, we sample a fixed total number of samples from the training datasets, with their proportions indicated by the "ratio" column in the table. Note that the number of images may differ from the full official release.

A.2. More Implementation details

Training Objective. In OmniVGGT, all input images, together with the available camera parameters and depth maps (if provided), are fed into the network $\mathcal{G}$, which predicts in an end-to-end manner the 3D point maps, complete camera poses, intrinsics, depth maps, and their corresponding confidence maps:

$$\mathcal{G}\left(\mathbf{I}, \mathbf{C}, \mathbf{D}\right) = (\hat{C}_i, \hat{P}_i, \hat{D}_i, \hat{Y}_i)_{i=1}^{N} \tag{9}$$

The training objectives of OmniVGGT consist of three components: camera, depth, and point map.

$$\mathcal{L} = \mathcal{L}_{\text{camera}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{pmap}}\,, \tag{10}$$

The camera loss $\mathcal{L}$ camera supervises the cameras $\hat{\mathbf{g}}_i$ with ground truth $\mathbf{g}_i$ using L1 loss:

$$\mathcal{L}_{\text{camera}} = \sum_{i=1}^{N} \|\hat{\mathbf{g}}_i - \mathbf{g}_i\|_1 \tag{11}$$

Following VGGT, we apply a confidence-aware regression loss to the depth and point map, both along with a gradient-based term:

$$\mathcal{L}_{\text{depth}} = \sum_{i=1}^{N} \left\| \Sigma_i^D \odot \left(\hat{D}_i - D_i\right) \right\| + \left\| \Sigma_i^D \odot \left(\nabla \hat{D}_i - \nabla D_i\right) \right\| - \alpha \log \Sigma_i^D \tag{12}$$

$$\mathcal{L}_{\text{pmap}} = \sum_{i=1}^{N} \left\| \Sigma_i^P \odot \left(\hat{P}_i - P_i\right) \right\| + \left\| \Sigma_i^P \odot \left(\nabla \hat{P}_i - \nabla P_i\right) \right\| - \alpha \log \Sigma_i^P \tag{13}$$

where $\odot$ is the channel-broadcast element-wise product and α is a hyper-parameter. In addition, our depth D, point map P, and camera translations t in ground truths are all normalized by dividing the average Euclidean distance of all 3D points in the point map P to the origin. It should be noted that this normalization process is different from the normalization used in the information injection stage of OmniVGGT.

Frame Sampling Strategy. For every batch, we select between 2 and 24 frames from multiple random training scenes while maintaining a constant total of 24 frames within each batch. We sample each batch of images based on camera pose similarity. For each frame, all other frames are ranked according to their pose similarity, and the top N most similar frames are selected as its valid range. Then, for each sequence, we randomly choose one frame as the anchor frame and sample the remaining frames from its valid range. In addition, our depth D, point map P, and camera translations t in ground truths are all normalized by dividing the average Euclidean distance of all 3D points in the point map P to the origin. It should be noted that this normalization process is different from the normalization used in the information injection stage.

Training Details. We initialize OmniVGGT by using pretrained weights from VGGT and fine-tune for 10 epochs of 12M iterations each. We train the model using the AdamW optimizer with a learning rate of 2×10^{-5} for prediction heads and 1×10^{-5} for backbone, which incorporates a 5K step linear warmup and cosine weight decay schedule. The p in our training objective is set to 10%. The spatial resolutions of the input images, depth maps, and point maps range from (518×168) to (518×518). We also use ColorJitter as the data augmentation to enhance the model's robustness in varying lighting conditions.

A.3. Baselines.

We mainly compare our approach with VGGT, Pow3R [21], and DUSt3R [67]. Both Pow3R and DUSt3R are feed-forward models that take a pair of views as input. The distinctive feature of Pow3R is that it can additionally incor-

Table 8. **Training datasets statistic.** Each training epoch is composed of 20 datasets, and their relative quantities are reported as a ratio.

Index	Dataset	Scene Type	Real/Synthetic	Dynamic	# of Frames	Training Prob (%)
1	**ARKitScene** [12]	Indoor	Real	Static	1.2M	2.07
2	**BlendedMVS** [73]	Mixed	Real	Static	1.1M	2.07
3	**DL3DV** [34]	Mixed	Real	Static	21M	17.81
4	**Dynamic Replica** [23]	Indoor	Synthetic	Dynamic	2.8M	4.68
5	**HyperSim** [49]	Indoor	Synthetic	Static	70K	3.55
6	**Kubric** [15]	Object	Synthetic	Dynamic	1.3M	1.67
7	**MapFree** [2]	Outdoor	Real	Static	2.6M	9.03
8	**MegaDepth** [32]	Outdoor	Real	Static	1.2M	2.07
9	**Matterport 3D** [46]	Indoor	Real	Static	1.9M	3.48
10	**MVS-Synth** [20]	Outdoor	Synthetic	Static	12K	1.34
11	**ScanNet** [10]	Indoor	Real	Static	23M	4.75
12	**ScanNet++** [74]	Indoor	Real	Static	7.8M	13.88
13	**Spring** [39]	Outdoor	Synthetic	Dynamic	4.9K	0.5
14	**Tartanair** [68]	Mixed	Synthetic	Static	3M	9.36
15	**Uasol** [5]	Outdoor	Real	Static	1.3M	2.41
16	**Unreal 4K** [60]	Outdoor	Synthetic	Static	16K	1.81
17	**Vkitti** [7]	Outdoor	Synthetic	Dynamic	42K	3.44
18	**Waymo** [58]	Outdoor	Real	Dynamic	7.9M	8.36
19	**WildRGBD** [70]	Object	Real	Static	1.9M	4.35

porate camera intrinsics, poses, and depth maps as auxiliary inputs, but only in a pairwise input.

A.4. More Results of Auxiliary Information Guidance

In this subsection, we present the performance of auxiliary information guidance on additional datasets. Tables 9 and 10 report the results on ARKitScene [12] and OmniWorld (Game) [80], respectively. The methods have not been trained on these datasets.

A.5. Full Results of Multi-View Depth Estimation

In this subsection, we present the complete results of the multi-view depth evaluation in Section 5.2. For OmniVGGT, all images are resized to a fixed width of 518 pixels, and the aspect ratio is adjusted to the closest aspect ratio used during training according to the original images. The reported results are averaged over all samples. As shown in Table 11, we further analyze the impact of injecting different percentages of depth information. We observe that both VGGT and OmniVGGT show relatively poor Rel performance on the ScanNet dataset, mainly because the ground-truth depth in ScanNet is noisy (e.g., walls and floors are not smooth). Although the predicted depth maps are visually reasonable, the quantitative metrics appear degraded.

A.6. More Results of 3D Reconstrunction

In this subsection, we present the complete 3D reconstruction results. Table 12 and 13 report the benchmark results on the 7-Scenes [55] and NRGB [3] datasets, respectively, including the effects of incorporating different types and percentages of auxiliary information.

A.7. More Visualizations

Fig. 7 presents a visual comparison between OmniVGGT and other methods on the 7-Scenes[55], NRGBD[56], and ETH3D [53] datasets. With the assistance of additional modality inputs (e.g., camera poses), OmniVGGT maintains accurate spatial relationships even in extreme cases—such as when two images have no overlap at all. Fig. 9 further demonstrates the strong generalization ability of OmniVGGT on in-the-wild data, where it achieves impressive results across both synthetic-engine-rendered and AI-generated images. Fig. 10 illustrates the reconstruction performance on image pairs that contain limited or no overlap between views. We also show the rollout examples of our method on the CALVIN benchmark in Fig. 11.

A.8. Architecture Ablation

In this subsection, we investigate the impact of different GeoAdapter strategies on model performance. We randomly select 100 batches from the Sintel [6] dataset, with 10 images per scene for inference, and evaluate the related metrics for depth and camera estimation. We trained the following three additional versions under the same training parameters and dataset. We experiment with three strategies: (a) *Replace*, where the original camera tokens are directly replaced by auxiliary camera tokens in each AA block, (b) *One-Layer Adapter*, where auxiliary camera tokens are only injected once before the AA blocks, and (c)

Table 9. **Impact of Auxiliary Information Injection on ARKitScenes (unseen) [12].** We also show in green the absolute improvement w.r.t. the results without auxiliary information. The best results in each category are **bold**.

Method	Aux. information (%)		Depth		Camera		
	Depth	Camera	Abs Rel↓	$\delta < 1.25$ ↑	RRA@5°↑	RTA@ 5°↑	AUC@30° ↑
VGGT	✗	✗	0.048	98.90	72.17	35.94	60.52
OmniVGGT	✗	✗	**0.035**	**99.02**	**73.66**	**50.03**	**65.82**
OmniVGGT + aux. information	30	✗	0.035 (+0.000)	99.57 (+0.55)	77.53 (+3.87)	50.59 (+0.56)	67.92 (+2.10)
	50	✗	0.029 (+0.006)	99.52 (+0.50)	77.62 (+3.96)	51.97 (+1.94)	**68.50 (+2.68)**
	70	✗	0.021 (+0.014)	99.57 (+0.55)	**76.34 (+2.68)**	53.15 (+3.12)	68.25 (+2.43)
	100	✗	**0.006 (+0.029)**	**99.97 (+0.95)**	74.59 (+0.93)	**53.69 (+3.66)**	67.99 (+2.17)
	✗	30	0.035 (+0.000)	99.31 (+0.29)	77.60 (+3.94)	51.40 (+1.37)	68.46 (+2.64)
	✗	50	0.035 (+0.000)	99.30 (+0.28)	79.69 (+6.03)	53.92 (+3.89)	70.69 (+4.87)
	✗	70	0.034 (+0.001)	99.28 (+0.26)	84.69 (+11.03)	56.59 (+6.56)	74.21 (+8.39)
	✗	100	**0.034 (+0.001)**	**99.89 (+0.87)**	**91.20 (+17.54)**	**64.98 (+14.95)**	**81.64 (+15.82)**
	100	100	**0.006 (+0.029)**	**99.97 (+0.95)**	**90.19 (+16.53)**	**67.00 (+16.97)**	**81.91 (+16.09)**

Table 10. **Impact of Auxiliary Information Injection on OmniWorld-Game (unseen) [80].** We also show in green the absolute improvement w.r.t. the results without auxiliary information. The best results in each category are **bold**.

Method	Aux. information (%)		Depth		Camera		
	Depth	Camera	Abs Rel↓	$\delta < 1.25$ ↑	RRA@5°↑	RTA@ 5°↑	AUC@30° ↑
VGGT [65]	✗	✗	0.260	79.10	73.97	58.57	63.75
OmniVGGT	✗	✗	**0.240**	**80.50**	**74.20**	**59.39**	**63.86**
OmniVGGT + aux. information	30	✗	0.208 (+0.032)	87.28 (+6.78)	74.42 (+0.22)	59.57 (+0.18)	63.91 (+0.05)
	50	✗	0.178 (+0.062)	90.68 (+10.18)	74.38 (+0.18)	59.48 (+0.09)	63.98 (+0.12)
	70	✗	0.151 (+0.089)	92.23 (+11.73)	**74.63 (+0.43)**	59.82 (+0.43)	64.20 (+0.34)
	100	✗	**0.095 (+0.145)**	**94.54 (+14.04)**	74.31 (+0.11)	**60.54 (+1.15)**	**64.87 (+1.01)**
	✗	30	0.239 (+0.001)	80.50 (+0.00)	74.29 (+0.09)	60.70 (+1.31)	64.75 (+0.89)
	✗	50	0.238 (+0.002)	80.51 (+0.01)	74.37 (+0.17)	61.93 (+2.54)	65.48 (+1.62)
	✗	70	0.237 (+0.003)	80.51 (+0.01)	74.42 (+0.22)	64.15 (+4.76)	68.06 (+4.20)
	✗	100	**0.237 (+0.003)**	**80.52 (+0.02)**	**77.91 (+3.71)**	**65.46 (+6.07)**	**71.91 (+8.05)**
	100	100	**0.094 (+0.146)**	**94.57 (+14.07)**	**79.43 (+5.23)**	**68.89 (+9.50)**	**74.35 (+10.49)**

Depth ZeroConV, where a ZeroConV layer is used for depth information injection. Table 14 reports the full results of our ablation experiments.

In addition, Fig. 8 illustrates the feature maps of the spatial tokens and auxiliary depth tokens in OmniVGGT, with and without the use of the depth ZeroConV. We use Principal Component Analysis (PCA) to reduce the high-dimensional token embeddings to three dimensions and save them as RGB images. We observe that when the ZeroConv layer is applied to depth injection, the meaningful information within the auxiliary depth features (e.g., edges and background structures) is largely suppressed, causing the network to treat the injected depth as noise. In this case, the feature representations of OmniVGGT with and without auxiliary depth inputs remain nearly identical, and the resulting performance shows almost no difference. Therefore, we chose not to include the depth ZeroConV in our final network design.

B. Comparison with MapAnything

We have conducted direct comparisons using the official MapAnything [24] model on our benchmarks. As reported in Table. 15 (Sintel) and Table. 16 (CO3D), OmniVGGT outperforms MapAnything in camera pose estimation when no auxiliary information is provided. When auxiliary depth or camera inputs are available, OmniVGGT achieves comparable performance to MapAnything. Furthermore, in the object-centric domain, OmniVGGT consistently and substantially outperforms MapAnything across all evaluated settings, demonstrating that OmniVGGT remains highly competitive and effective.

Table 11. **Multi-view Depth Evaluation.** (Parentheses) denote training on data from the same domain. K", "RT", and "D" denote intrinsic, relative pose, and depth information, respectively. The best and second best results are **bold** and underlined respectively.

Method	GT range	Extra info	ScanNet [10]		ETH3D [53]		DTU [1]		T&T [27]		Average	
			rel↓	τ ↑	rel↓	τ ↑	rel↓	τ ↑	rel↓	τ ↑	rel↓	τ ↑
COLMAP [51, 52] (K+RT)	×	×	14.6	34.2	16.4	55.1	0.7	96.5	2.7	95.0	8.6	70.2
COLMAP Dense [51, 52] K+RT	×	×	38.0	22.5	89.8	23.2	20.8	69.3	25.7	76.4	43.6	47.9
MVSNet [72] (K+RT)	✓	×	22.7	20.9	21.6	35.6	(1.8)	(86.7)	6.5	74.6	12.3	11.1
Vis-MVSNet [76] (K+RT)	✓	×	8.9	33.5	10.8	43.3	1.8	87.4	4.1	7.2	6.4	42.9
MVS-Former++ (K+RT)	✓	×	15.2	21.9	21.4	32.5	(1.2)	(91.9)	7.6	71.5	10.8	8.5
CER-MVS [9] (K+RT)	×	×	21.1	24.3	11.7	47.5	4.1	71.3	6.4	82.1	10.8	56.3
DUSt3R [67]	×	med	(3.1)	(71.8)	3.0	76.0	3.9	68.6	3.3	75.1	3.3	72.9
Pow3R [21]	×	med	(3.2)	(68.8)	3.0	74.7	3.0	74.3	3.3	76.6	3.1	73.6
VGGT [65]	×	med	(3.7)	(70.0)	1.7	87.2	0.9	95.4	1.7	90.6	2.0	85.8
OmniVGGT	×	med	(3.6)	(72.3)	1.8	87.5	1.1	93.9	1.8	90.0	2.1	85.9
Pow3R [21] w/ 100% (K+RT)	×	med	(3.1)	(71.4)	2.8	77.1	1.5	91.1	3.2	78.2	2.7	79.5
OmniVGGT w/ 100% (K+RT)	×	med	(3.7)	(72.2)	1.8	87.8	1.2	93.6	1.8	89.9	2.1	85.9
OmniVGGT w/ 30% D	×	med	(2.3)	(85.5)	0.7	97.9	0.5	99.2	1.2	93.2	1.2	94.0
OmniVGGT w/ 50% D	×	med	(2.3)	(85.9)	0.6	98.6	0.4	99.4	1.0	94.4	1.1	94.6
OmniVGGT w/ 70% D	×	med	(2.3)	(86.0)	0.6	98.7	0.4	99.4	0.9	95.5	1.1	94.9
OmniVGGT w/ 100% D	×	med	(2.3)	(85.6)	0.5	98.7	0.3	**99.5**	0.9	95.5	1.0	94.8
OmniVGGT w/ 100% (K+RT+D)	×	med	**(2.2)**	**(86.7)**	**0.5**	**98.7**	**0.3**	99.4	**0.9**	**95.6**	**1.0**	**95.1**

Table 12. **3D reconstruction on the 7-scenes [55] datasets.** K", "RT", and "D" denote intrinsic, relative pose, and depth information, respectively. The best and second best results in each category are **bold** and underlined respectively.

Method	Acc↓		Comp↓		NC↑	
	Mean	Med.	Mean	Med.	Mean	Med.
VGGT [65]	**0.087**	0.039	**0.091**	0.039	**0.787**	**0.890**
Fast3R [71]	0.164	0.108	0.163	0.080	0.686	0.775
DUSt3R-GA [67]	0.146	0.077	0.181	0.067	0.736	0.839
MASt3R-GA [30]	0.185	0.081	0.180	0.069	0.701	0.792
MonST3R-GA [77]	0.248	0.185	0.266	0.167	0.672	0.759
Spann3R [62]	0.298	0.226	0.205	0.112	0.650	0.730
SLAM3R [35]	0.287	0.155	0.226	0.066	0.644	0.720
CUT3R [66]	0.126	0.047	0.154	0.031	0.727	0.834
OmniVGGT	0.104	**0.037**	0.112	**0.031**	0.763	0.875
OmniVGGT w/ 30% D	0.098	0.034	0.110	0.028	0.778	0.889
OmniVGGT w/ 50% D	0.103	0.033	0.106	0.028	0.781	0.890
OmniVGGT w/ 70% D	0.094	0.045	0.094	0.033	0.785	0.885
OmniVGGT w/ 100% D	0.085	0.034	0.085	0.027	0.789	0.894
OmniVGGT w/ 30% (K+RT)	0.100	0.034	0.107	0.027	0.761	0.875
OmniVGGT w/ 50% (K+RT)	0.101	0.035	0.108	0.027	0.761	0.875
OmniVGGT w/ 70% (K+RT)	0.089	0.022	0.096	0.022	0.771	0.886
OmniVGGT w/ 100% (K+RT)	0.037	0.017	0.049	0.019	0.778	0.893
OmniVGGT w/ 100% (K+RT+D)	**0.036**	**0.017**	**0.036**	**0.017**	**0.810**	**0.912**

Table 13. **3D reconstruction on the NRGB [3] datasets.** K", "RT", and "D" denote intrinsic, relative pose, and depth information, respectively. The best and second best results in each category are **bold** and underlined respectively.

Method	Acc↓		Comp↓		NC↑	
	Mean	Med.	Mean	Med.	Mean	Med.
VGGT [65]	**0.087**	0.039	**0.091**	0.039	**0.787**	**0.890**
Fast3R [71]	0.164	0.108	0.163	0.080	0.686	0.775
DUSt3R-GA [67]	0.146	0.077	0.181	0.067	0.736	0.839
MASt3R-GA [30]	0.185	0.081	0.180	0.069	0.701	0.792
MonST3R-GA [77]	0.248	0.185	0.266	0.167	0.672	0.759
Spann3R [62]	0.298	0.226	0.205	0.112	0.650	0.730
SLAM3R [35]	0.287	0.155	0.226	0.066	0.644	0.720
CUT3R [66]	0.126	0.047	0.154	0.031	0.727	0.834
OmniVGGT	0.104	**0.037**	0.112	**0.031**	0.763	0.875
OmniVGGT w/ 30% D	0.098	0.034	0.110	0.028	0.778	0.889
OmniVGGT w/ 50% D	0.103	0.033	0.106	0.028	0.781	0.890
OmniVGGT w/ 70% D	0.094	0.045	0.094	0.033	0.785	0.885
OmniVGGT w/ 100% D	0.085	0.034	0.085	0.027	0.789	0.894
OmniVGGT w/ 30% (K+RT)	0.100	0.034	0.107	0.027	0.761	0.875
OmniVGGT w/ 50% (K+RT)	0.101	0.035	0.108	0.027	0.761	0.875
OmniVGGT w/ 70% (K+RT)	0.089	0.022	0.096	0.022	0.771	0.886
OmniVGGT w/ 100% (K+RT)	0.037	0.017	0.049	0.019	0.778	0.893
OmniVGGT w/ 100% (K+RT+D)	**0.036**	**0.017**	**0.036**	**0.017**	**0.810**	**0.912**

Figure 7. **Visual Comparisons on 7-Scenes [55], NRGBD [3], and ETH3D [53] datasets.**

Table 14. **Ablation of GeoAdapter architectures.** We compare different GeoAdapter designs on the Sintel [6] dataset. The 'Replace' refers to directly substituting the original camera tokens with auxiliary camera tokens, whereas 'One-Layer GeoAdapter' represents injecting auxiliary camera tokens only once before the encoder. 'Depth ZeroConV' denotes using ZeroConV for depth injection.

Architecture	Depth		Camera		
	Abs Rel↓	$\delta < 1.25$ ↑	RRA@5° ↑	RTA@5° ↑	AUC@30° ↑
w/o (K+RT+D) Auxiliary Information					
(a) Replace	0.845	64.74	93.40	30.88	64.74
(b) One-Layer Adapter	0.604	68.74	96.78	44.92	68.74
(c) Depth ZeroConV	0.569	70.71	96.44	51.86	69.70
(d) OmniVGGT	**0.558**	**71.46**	96.15	**54.01**	**70.83**
w/ (K+RT) Auxiliary Information					
(a) Replace	0.842	65.11	96.98	55.66	76.28
(b) One-Layer Adapter	0.563	70.16	97.54	58.84	77.00
(c) Depth ZeroConV	0.569	70.71	99.65	69.29	83.18
(d) OmniVGGT	**0.553**	**72.36**	**99.97**	**75.83**	**85.35**
w/ Depth Auxiliary Information					
(a) Replace	0.670	82.96	94.69	52.04	71.61
(b) One-Layer Adapter	0.107	85.91	95.97	52.34	74.66
(c) Depth ZeroConV	0.570	71.02	95.45	56.10	72.93
(d) OmniVGGT	**0.106**	**85.95**	**96.93**	**59.73**	**77.16**
w/ (K+RT+D) Auxiliary Information					
(a) Replace	0.655	82.96	97.08	57.61	77.83
(b) One-Layer Adapter	0.133	85.65	99.97	60.89	81.66
(c) Depth ZeroConV	0.505	71.11	99.72	71.66	84.12
(d) OmniVGGT	**0.106**	**85.95**	**99.97**	**76.33**	**85.99**

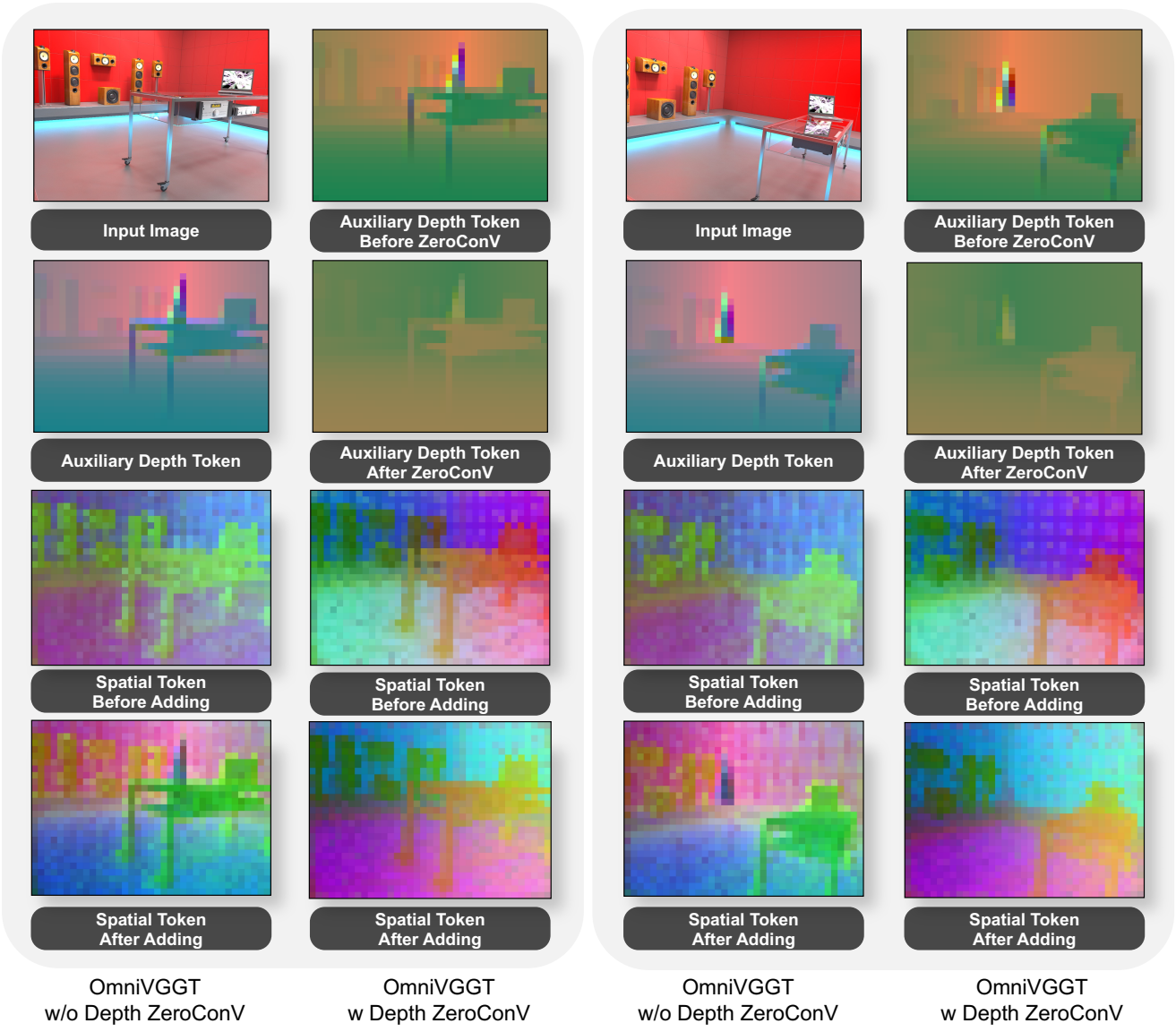

Figure 8. **Feature Map Visualization Comparison Between OmniVGGT and OmniVGGT w/ Depth ZeroConV. Left Column**: OmniVGGT without applying the ZeroConV to the depth adapter. From top to bottom: the input image, the feature map of the auxiliary depth token after passing through the depth encoder; the spatial token feature map obtained from the image after the DINO encoder; and the spatial token feature map after adding the auxiliary depth tokens. **Right Colum**: OmniVGGT with the ZeroConV applied to the auxiliary depth information. From top to bottom: the auxiliary depth token feature map before and after the ZeroConV; and the spatial token feature maps before and after the addition of auxiliary depth tokens.

Figure 9. **Feed-Forward 3D Point Map by OmniVGGT with In-The-Wild Inputs.**

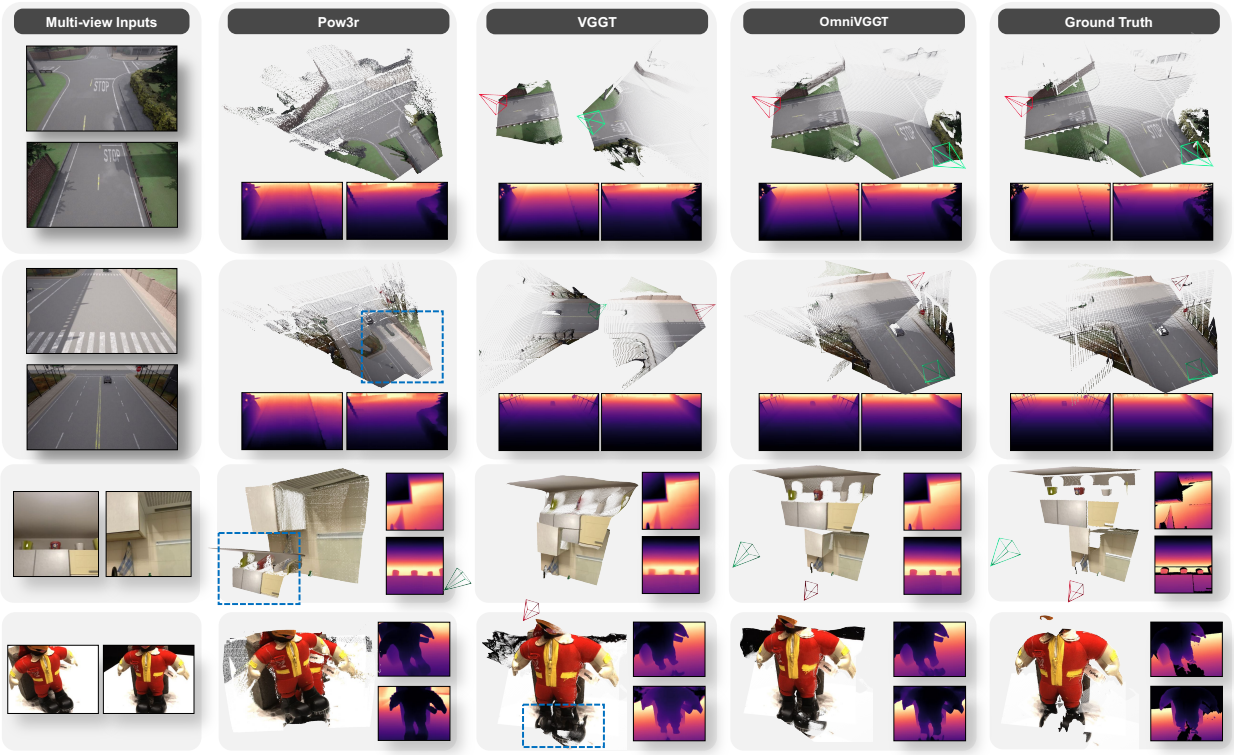

Figure 10. **More Visualizations on Image Pairs Input.**

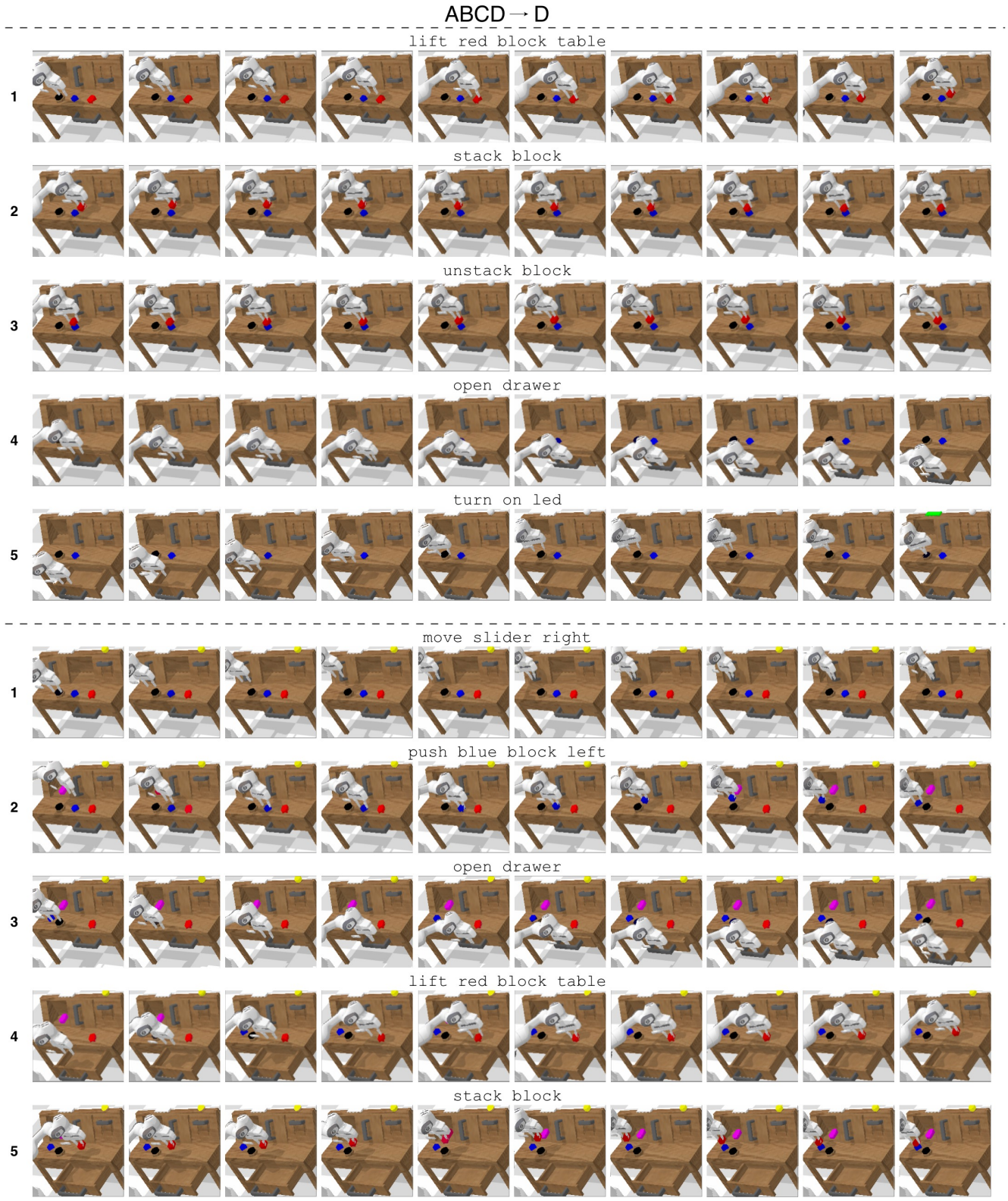

Figure 11. **Ours Rollouts on the ABCD→D split of the CALVIN benchmark.**

Table 15. **The main performance comparison of OmniVGGT and mapanything [24] on the Sintel [6] datasets.**

Method	Aux. information		Depth		Camera		
	Depth	Camera	Abs Rel↓	$\delta < 1.25$ ↑	RRA@5°↑	RTA@ 5°↑	AUC@30° ↑
VGGT	✗	✗	0.722	70.81	95.69	53.92	70.55
MapAnything	✗	✗	**0.348**	68.87	89.20	30.19	52.80
OmniVGGT	✗	✗	0.558	**71.46**	**96.15**	**54.01**	**70.83**
OmniVGGT + aux. information	✓	✗	0.106	85.95	96.93	59.73	77.16
	✗	✓	0.553	72.36	99.97	75.83	85.35
	✓	✓	0.106	85.95	99.97	76.33	85.99
MapAnything + aux. information	✓	✗	0.108	87.90	90.87	42.09	63.37
	✗	✓	0.305	70.99	100	89.88	93.00
	✓	✓	**0.100**	**87.92**	**100**	**77.68**	**88.42**

Table 16. **The main performance comparison of OmniVGGT and mapanything [24] on the CO3D [48] (Object-Centric) datasets.**

Method	Aux. information		Depth		Camera		
	Depth	Camera	Abs Rel↓	$\delta < 1.25$ ↑	RRA@5°↑	RTA@ 5°↑	AUC@30° ↑
VGGT	✗	✗	0.048	95.68	97.73	78.82	88.24
MapAnything	✗	✗	0.127	88.18	68.48	36.52	60.50
OmniVGGT	✗	✗	**0.048**	**96.58**	**97.42**	**80.07**	**88.38**
OmniVGGT + aux. information	✓	✗	0.026	98.21	97.13	91.15	92.81
	✗	✓	0.042	96.94	98.98	93.16	95.43
	✓	✓	**0.026**	**98.20**	**98.89**	**94.24**	**96.13**
MapAnything + aux. information	✓	✗	0.040	97.84	84.49	58.50	72.57
	✗	✓	0.134	81.44	96.09	81.89	89.48
	✓	✓	0.029	98.08	99.44	89.20	93.40