Unposed-to-3D: Learning Simulation-Ready Vehicles from Real-World Images

Supplementary Material

1. Implementation Details

1.1. Network Architectures

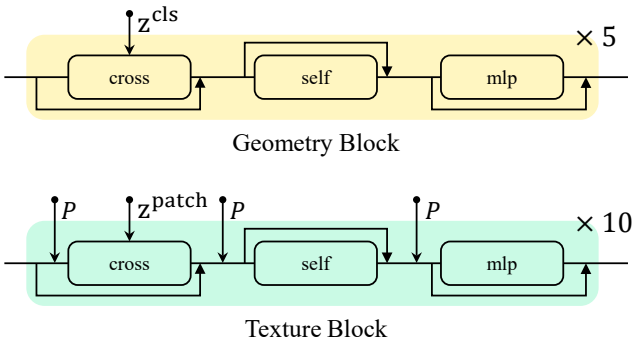

Figure 1. The two base modules of the backbone network.

Aggregation Module. We adopt the DINOv2-base [5] encoder and uniformly resize all input images to 224×224. A learnable camera embedding is introduced as a single token without positional encoding. The aggregation module consists of four Alternating Self-Attention blocks, each containing two self-attention layers. An attention-pooling layer implemented with a single-layer MLP predicts view-dependent weighting coefficients over all tokens and produces the final aggregated feature representation.

Learnable Object Embedding. We parameterize the structural representation as a learnable embedding of shape $N \times C$, where $N = 4096$ and $C = 768$, matching the dimensionality of DINO features. Each token is equipped with a sinusoidal positional encoding defined as:

$$\mathrm{PE}(x) = [\sin(x \cdot f_i), \cos(x \cdot f_i)]_{i=1}^{C/2}, \quad f_i = \frac{1}{10000^{i/(C/2)}}, \tag{1}$$

where $x \in \{0, 1, \ldots, N-1\}$ denotes the token index. This encoding provides spatial inductive bias for the learnable object tokens.

Backbone. As illustrated in Figure 1. The Geometry Block is composed of five blocks. It outputs geometry features G, which are decoded by three single-layer fully connected heads to predict spherical coordinates (r, θ, ϕ). These are then converted into Cartesian coordinates $P = (x, y, z)$. The predicted 3D positions are subsequently used inside the Texture Blocks to modulate per-token features, injecting explicit spatial information into the texture representation. The Texture Block contains ten base blocks. The resulting feature tokens are decoded by zero-initialized single-layer linear heads into $m = 8$ Gaussian attributes per token. Given $N = 4096$ tokens, this results in a total of $8 \times 4096 = 32768$ 3D Gaussians.

Training. We train the entire network using AdamW with an initial learning rate lr=0.00016. A warm-up phase linearly ramps the learning rate from $0.1 \times$ lr over the first T_{warmup} iterations. This is followed by a cosine annealing schedule that decays the learning rate to $0.01 \times$ lr over the remaining training steps. Our experiments are conducted on 24 NVIDIA A800 GPUs. The full training process takes approximately three days. We first train the model for 30 epochs on the 3DRealCar [2] dataset, followed by an additional 5 epochs of fine-tuning on the MAD-Cars [4] dataset.

1.2. Data Preparation Details

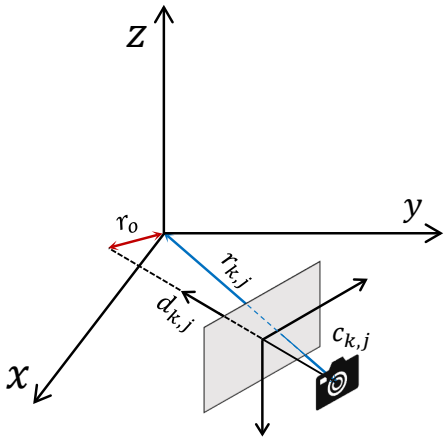

Figure 2. Schematic illustration of the camera construction

3DRealCar. To generate training views for each object instance, we place virtual cameras on randomized spherical trajectories surrounding the normalized 3D model. We sample $T = 4$ orbits, each containing $V = 32$ equally spaced azimuth angles. For the k-th orbit, the elevation angle ϕ_k is drawn from a uniform range $[75°, 90°]$, and a per-view camera radius $r_{k,j} \sim \mathcal{U}(1.2, 1.8)$ is used to introduce additional geometric diversity. The spherical position of the j-th camera on orbit k is given by

$$\mathbf{c}_{k,j} = \begin{bmatrix} r_{k,j} \sin\phi_k \cos\theta_{k,j} \\ r_{k,j} \sin\phi_k \sin\theta_{k,j} \\ r_{k,j} \cos\phi_k \end{bmatrix}, \theta_{k,j} = \frac{2\pi j}{V} + \delta_j, \tag{2}$$

where δ_j is a small random offset to avoid perfectly symmetric camera layouts. Rather than enforcing a fixed look-at

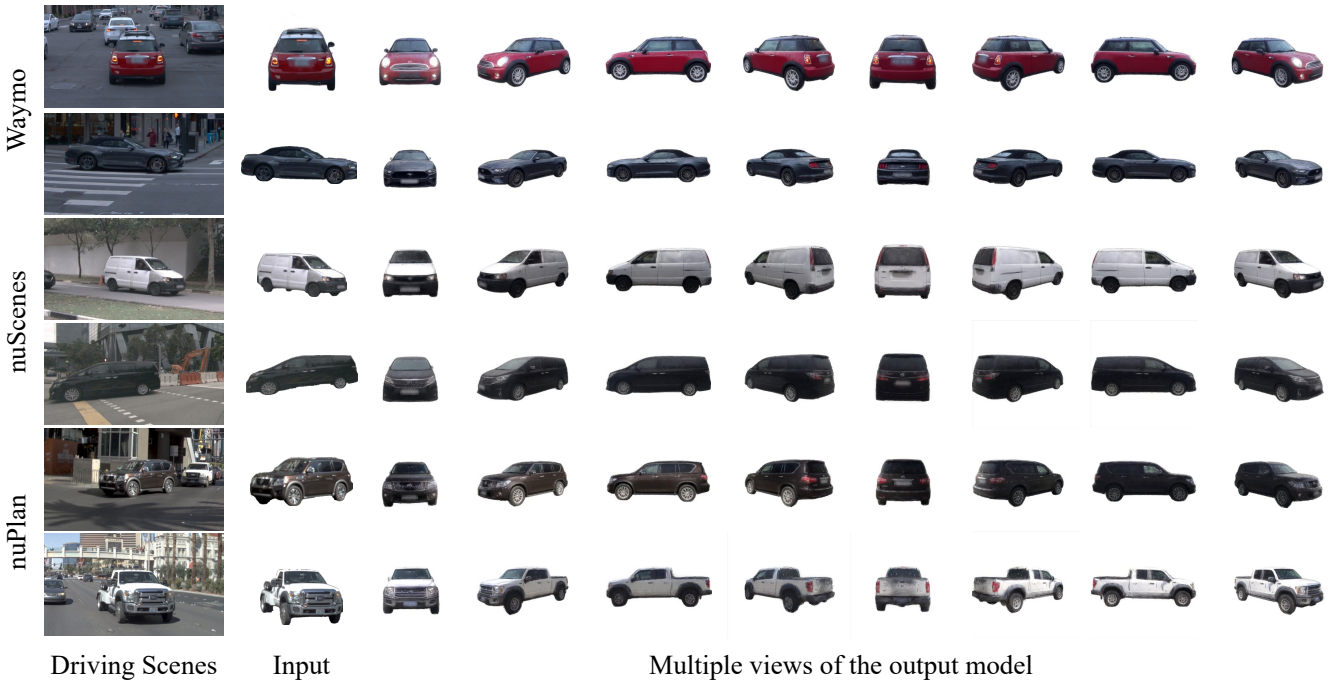

Figure 3. Qualitative generalization results of single-image reconstruction on autonomous driving datasets.

direction, each camera is oriented toward a randomly sampled point:

$$\hat{v}_i = \frac{v_i}{|v_i|_2}, v_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3), \tag{3}$$

where $i = \{1, 2, \ldots, TV\}$. The points $p_{k,j}$ are sampled within a small sphere centered at the origin, with the radius $r_o \sim \mathcal{U}(0, 0.1)$, i.e., $p_{k,j} = r_o \hat{v}_{ij}$. This defines the viewing direction as

$$d_{k,j} = \frac{p_{k,j} - c_{k,j}}{|p_{k,j} - c_{k,j}|}, \tag{4}$$

which induces natural variations in camera orientation, thereby mitigating overfitting. The camera intrinsics are also randomized for each orbit: a field of view FOV $\sim \mathcal{U}(50°, 70°)$ is sampled, while the principal point remains fixed at the image center. For each pair of extrinsic parameters $(\mathbf{q}, \mathbf{t})$ and intrinsics $\mathbf{f}$, we render a 512×512 RGB image along with its corresponding alpha mask.

MAD-Cars. The MAD-Cars [4] dataset contains only image data. For each image, we first extract instance-level bounding boxes and masks using YOLO [3] and MobileSAM [7]. We then discard images whose bounding boxes lie near the image boundaries, as such crops are likely to be incomplete. For each remaining instance, we normalize all associated images by scaling them according to the longest side of its bounding box and centering the bounding box within the image.

2. More Results

Zero-shot in Driving Scenario. As shown in Figure 3, we provide additional qualitative results on autonomous driving datasets. Relying solely on single-view inputs, our method is able to recover high-quality vehicle models from real driving environments, demonstrating its strong generalization capability and practical applicability.

More Comparisons. To facilitate a qualitative comparison between the proposed method and the baselines, we present additional qualitative comparison on the 3DRealCar [2] and CFV [1] datasets, as shown in Figure 7, Figure 8, Figure 9, Figure 10. Across varying lighting conditions and diverse asset types, our method consistently outperforms the baseline approaches.

Camera Estimation. To validate the accuracy of our camera parameter predictions, we provide qualitative results in Figure 11. For each instance, the first row shows the input ground truth images without camera poses, and the second row presents the rendered outputs obtained by applying the predicted camera parameters to our reconstructed models. As observed, our camera predictions achieve pixel-level precision, which is crucial for enabling high-quality 3D reconstruction using only unposed image supervision. This capability, absent in prior 3D generative models, highlights the scalability of our method.

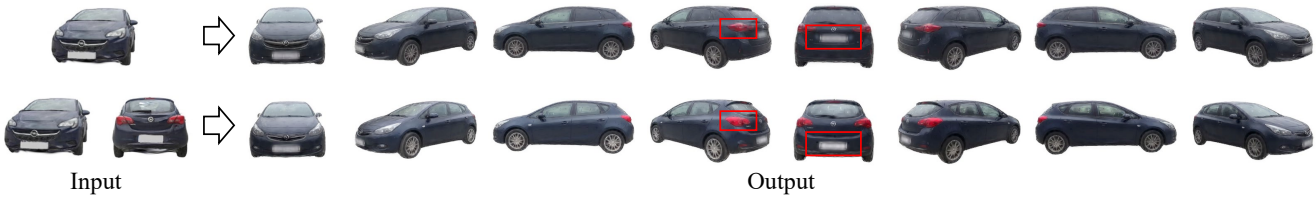

Figure 4. Qualitative reconstruction results under different input settings. We present results using a single input view as well as using two input views, where an additional viewpoint is provided to enhance reconstruction quality.

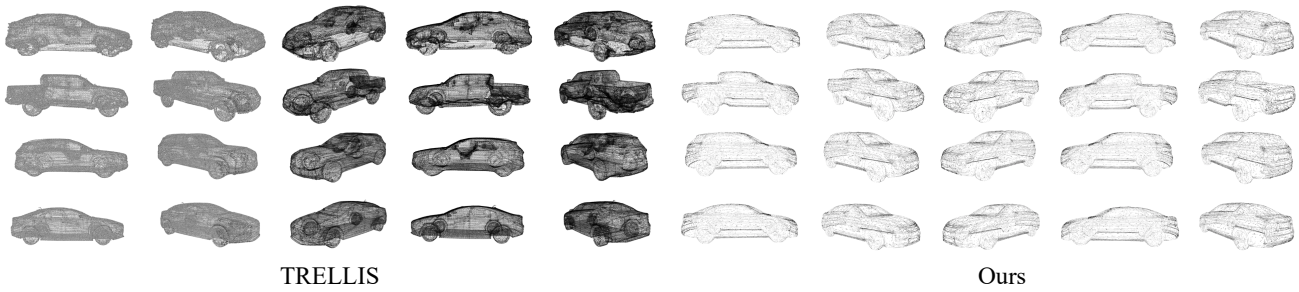

Figure 5. We visualize the Gaussian points to illustrate geometric quality. Compared to TRELLIS [6], our method achieves high-fidelity geometric reconstruction using substantially fewer Gaussian points.

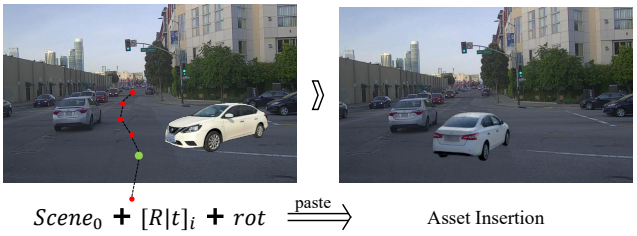

Figure 6. Harmonized Training Scene Construction.

Multi View Reconstruction. To validate the effectiveness of our multi-view input strategy, we present qualitative comparisons in Figure 4. The first row shows reconstruction results using only a frontal-view image, while the second row incorporates an additional rear-view image as input. Adding an extra viewpoint improves the reconstruction quality of rear details, such as taillights and license plates, demonstrating the advantages of multi-view input in our method. Because the single-view setting already yields strong reconstruction quality, incorporating multi-view inputs only further refines some fine-grained geometric details, so the resulting quantitative improvements remain marginal, which is consistent with the main paper.

Geometric Reconstruction. To evaluate the geometric reconstruction quality of our method, we visualize the generated Gaussian point clouds in Figure 5. The distribution of Gaussians produced by our model closely aligns with the underlying image textures: high-frequency regions are represented with a higher density of Gaussian points, whereas low-frequency areas require substantially fewer points. In contrast, the baseline method TRELLIS [6] typically relies on hundreds of thousands of points to represent a single object. Remarkably, our approach achieves superior reconstruction quality using only about thirty thousand Gaussians, demonstrating a compact yet more expressive geometric representation.

Harmonization. The module consists of a single self-attention block followed by a two-layer MLP, and is trained under MSE, SSIM, LPIPS, and L_1 losses to jointly regularize appearance. The learning rate is set to 0.1, and the weights of all image-level losses are fixed to 10. As shown in Figure 6, unlike prior approaches that require reconstructing the entire background and rely on manual adjustments for 3D asset insertion, we only retain the background image of the initial frame $scene_0$, along with the ego relative rotation and translation $[R|t]_t$ from the target keyframe to the initial frame. For the harmonized generated assets, we further apply an additional rotation rot around the upright axis to adjust their pose and modulate the texture properties at different views. Finally, we render only the foreground object and paste it onto the background, producing a geometrically consistent scene. We provide additional visual results of harmonization across diverse scenes and assets in Figure 12.

More Ablation. As shown in Table 1, we conduct further ablation studies to evaluate our method. Training explicit geometry generation without 3D geometric supervision is a severely ill-posed problem. Without our Progres-

Sphere	Filter	3DRealcar SSIM↑	PSNR↑	LPIPS↓	CD↓	F-score↑	CFV SSIM↑	PSNR↑	LPIPS↓	CD↓	F-score↑
✗	✗			–					–		
✓	✗	0.9059	19.8572	0.0742	0.9348	0.3917	0.9043	20.1337	0.0703	1.1861	0.2940
✓	1σ	0.9136	20.7738	0.0617	0.7195	0.4776	0.9125	21.1185	0.0570	0.9032	0.3455
✓	2σ	0.9121	20.5656	0.0637	0.7339	0.4654	0.9118	21.0839	0.0571	0.9482	0.3374
✓	3σ	**0.9172**	**21.2033**	**0.0571**	**0.5782**	**0.5341**	**0.9183**	**21.7250**	**0.0508**	**0.7439**	**0.4252**

Table 1. Ablation study of sphere initialization and gradient filtering strategies.

sive Spherical Optimization, the network suffers from collapse and fails to produce meaningful results. 3σ Gradient Filtering strategy is grounded in the Standard Deviation Rule, where gradients exceeding 3σ are treated as outliers. This effectively prevents erroneous gradients from initial camera-geometry ambiguities.

3. Limitations and Future works

Our method is unable to handle input images that exhibit geometric distortions; severe non-uniform stretching or compression typically leads to corresponding deformations in the reconstructed 3D model. In addition we do not explicitly model shadows cast by environmental illumination. The realism of asset insertion can still be improved during scene integration [8].

References

[1] Andy Catruna, Pavel Betiu, Emanuel Tertes, Vladimir Ghita, Emilian Radoi, Irina Mocanu, and Mihai Dascalu. Car full view dataset: Fine-grained predictions of car orientation from images. *Electronics*, 12(24):4947, 2023. 2, 7, 8

[2] Xiaobiao Du, Yida Wang, Haiyang Sun, Zhuojie Wu, Hongwei Sheng, Shuyun Wang, Jiaying Ying, Ming Lu, Tianqing Zhu, Kun Zhan, et al. 3drealcar: An in-the-wild rgb-d car dataset with 360-degree views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26488–26498, 2025. 1, 2, 5, 6

[3] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. 2

[4] Polina Karpikova, Daniil Selikhanovych, Kirill Struminsky, Ruslan Musaev, Maria Golitsyna, and Dmitry Baranchuk. Madrive: Memory-augmented driving scene modeling. *arXiv preprint arXiv:2506.21520*, 2025. 1, 2

[5] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[6] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 3

[7] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 2

[8] Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 4

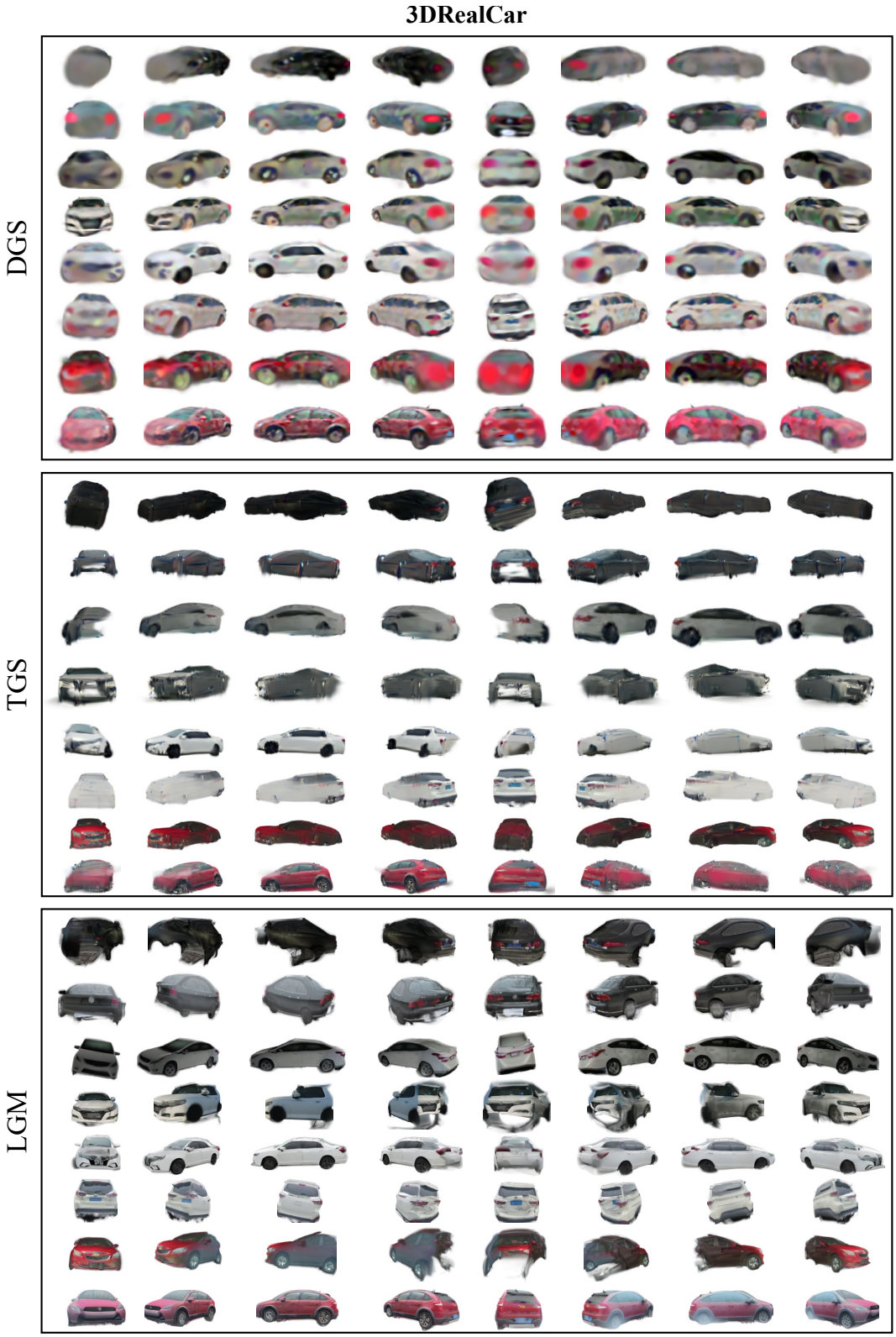

Figure 7. Qualitative results of LGM, TGS, DGS on 3DRealCar [2].

Figure 8. Qualitative results of TRELLIS and ours on 3DRealCar [2].

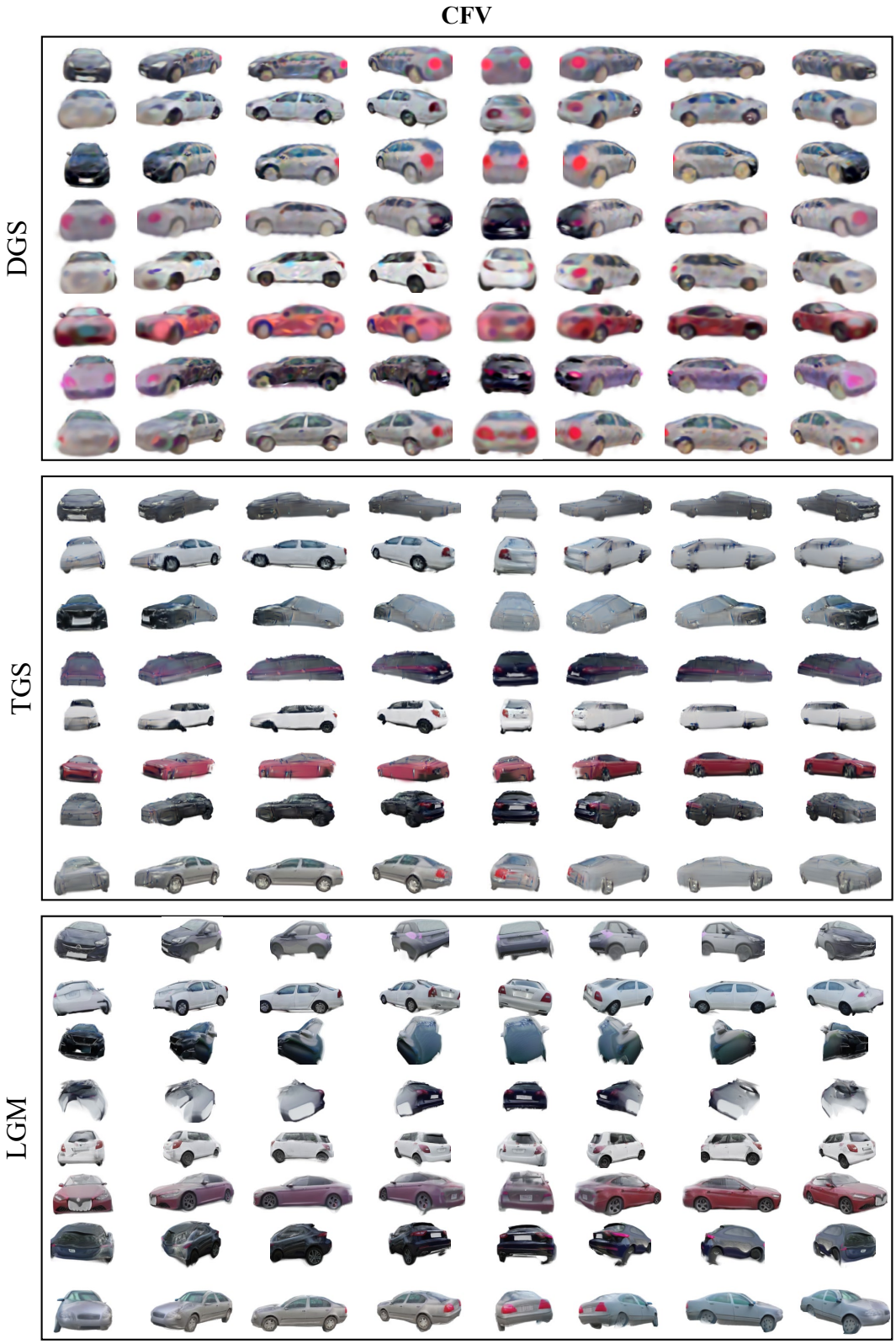

Figure 9. Qualitative results of LGM, TGS, DGS on CFV [1].

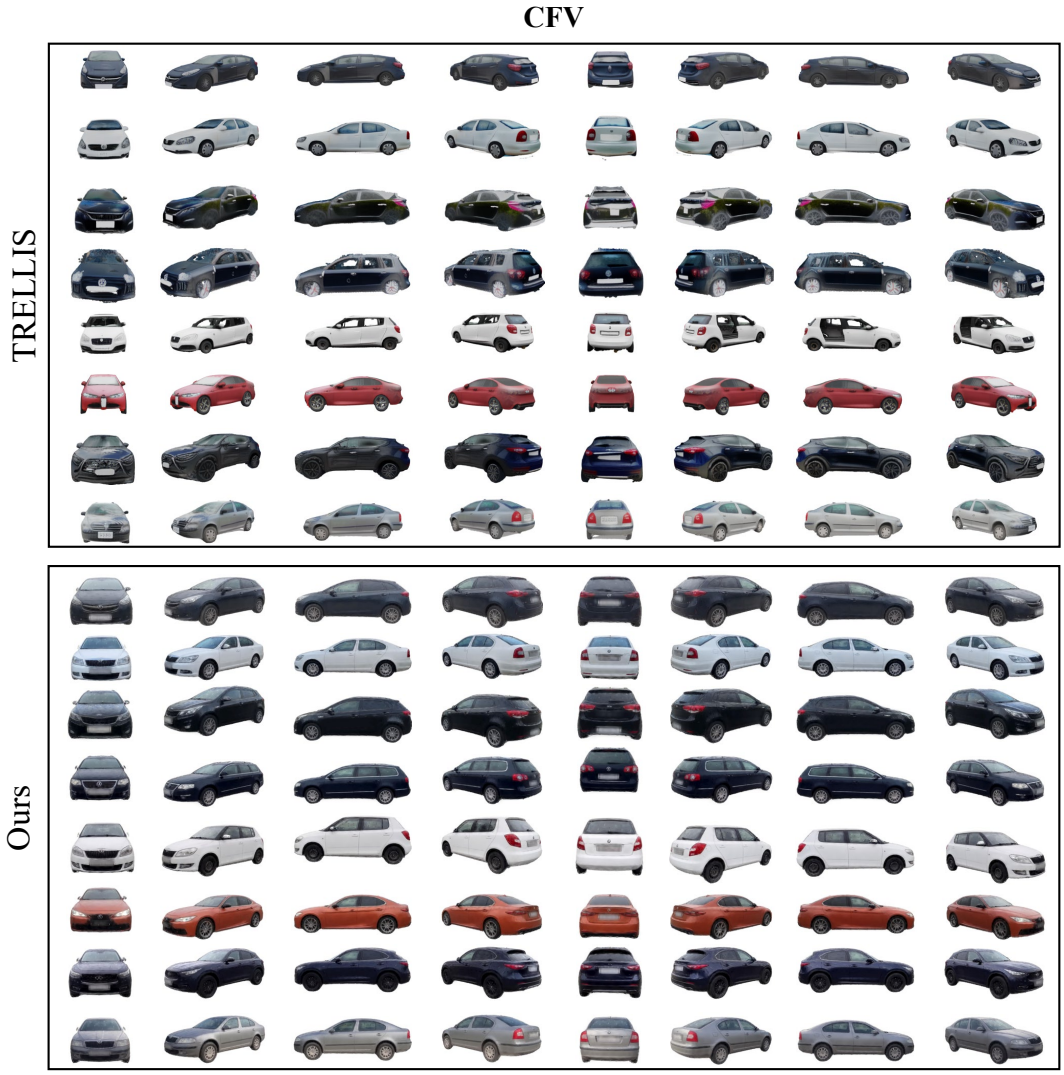

Figure 10. Qualitative results of TRELLIS and ours on CFV [1].

Figure 11. Qualitative results of camera parameters prediction.

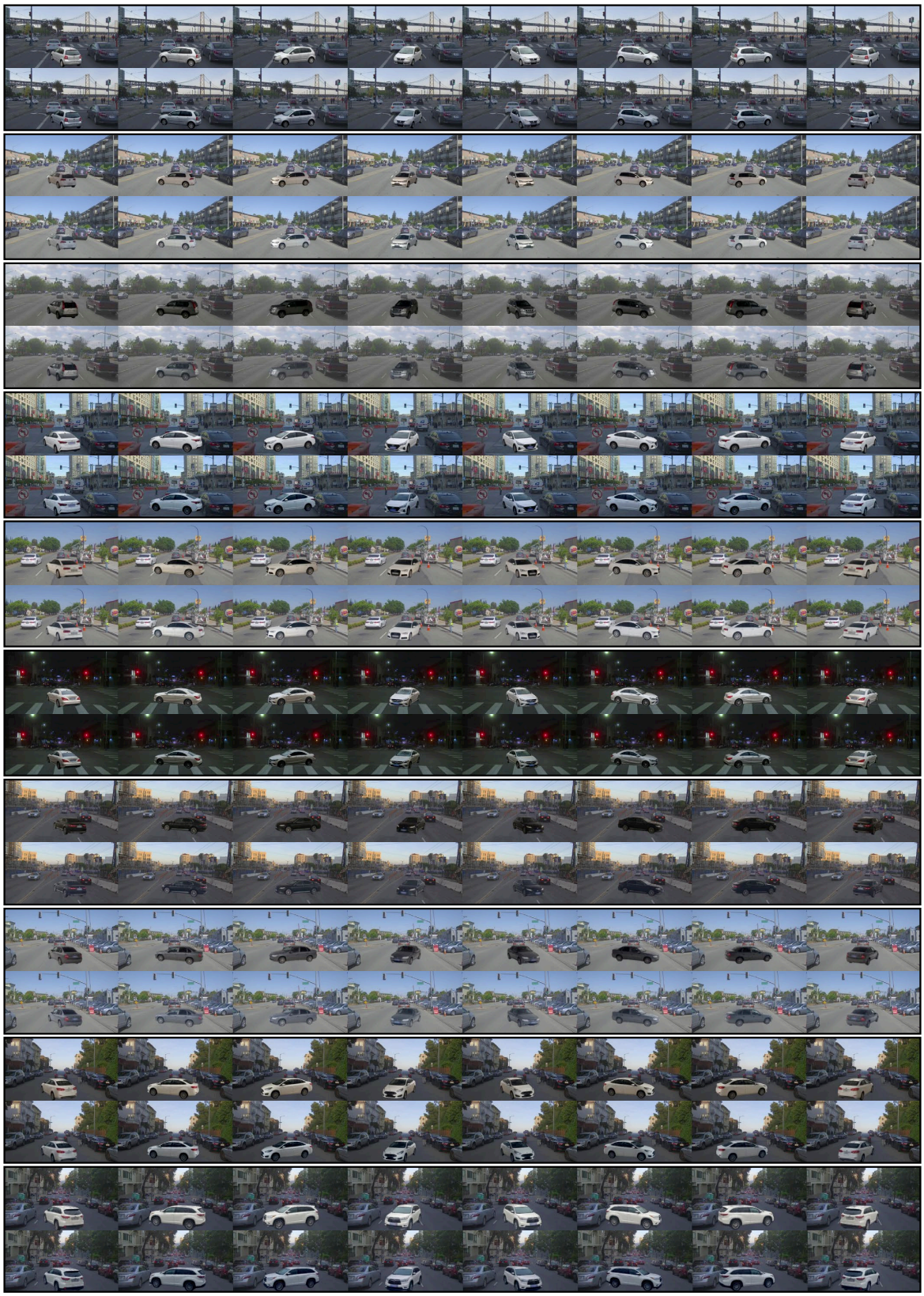

Figure 12. Qualitative results of harmonization. The first row presents the pre-harmonization outputs, while the second row shows the harmonized results. Examples are shown across multiple poses of the same asset within the scene to illustrate consistency and robustness.