Mind the Hitch: Dynamic Calibration and Articulated Perception for Autonomous Trucks

Supplementary Material

In this Appendix, we provide the following:

- Implementation details in Appendix A.
- Additional dataset descriptions and visualizations in Appendix B.
- Runtime and training parameters in Appendix C.
- More qualitative results under different articulated driving scenarios in Appendix D.
- Limitations and future work in Appendix E.

A. Implementation Details

Metric-Scale Prediction. Dynamic trailer calibration requires camera poses expressed in metric units because the tractor–trailer linkage evolves in real distance and orientation. The magnitude of this inter-rig baseline determines the validity of multiview geometry, influences parallax and triangulation depth, and conditions the spatial accuracy of downstream modules such as BEV feature lifting, object localization, and motion estimation. Any ambiguity in scale introduces inconsistent baselines across frames, which directly degrades geometric alignment and accumulates as bias in downstream perception.

Unlike existing learning-based geometric methods, our framework dCAP directly predicts trailer rear camera poses in **metric scale** without requiring any post-hoc scale recovery. Current learning-based geometric models, including VGGT and DUSt3R, cannot provide such metric-scale poses. Their regression heads are optimized under a similarity-equivariant formulation due to limited availability and inconsistency of metric-supervised data across training sources. As a consequence, these models estimate camera motion only up to an arbitrary global scale. This yields relative geometry but omits absolute metric information, leading to per-sequence scale drift that prevents their direct use for articulated calibration, where the inter-rig translation must be recovered in meters at every frame. To enable a fair comparison, we convert their normalized predictions into metric scale via a per-frame scale estimation procedure described below.

Problem Setup. Let $T_i^\star \in \mathrm{SE}(3)$ denote the ground-truth extrinsic matrix of camera i (tractor front, trailer rear, etc.) expressed in the world frame, and let $\hat{T}_i$ be the corresponding prediction from a baseline method. We use the tractor's front camera and the trailer's rear camera as a calibration pair. Their ground-truth relative transform is

$$T^\star_{F\to B} = (T^\star_B)^{-1} T^\star_F, \tag{1}$$

and the baseline prediction yields

$$\hat{T}_{F\to B} = (\hat{T}_B)^{-1} \hat{T}_F. \tag{2}$$

We decompose both into rotation and translation, $T^\star_{F\to B} = (R^\star,\, t^\star)$ and $\hat{T}_{F\to B} = (\hat{R},\, \hat{t})$, and treat the unknown global scale as a scalar factor s on the translation component.

Scale Estimation. To recover a metric scale for the baseline prediction, we align the ground-truth and predicted relative translations by a one-dimensional least-squares fit. Concretely, we compute

$$s = \frac{(t^\star)^\top \hat{t}}{\hat{t}^\top \hat{t}}, \tag{3}$$

which is the optimal scalar minimizing $\|t^\star - s\hat{t}\|_2^2$. The rotation $\hat{R}$ is left unchanged. This scale factor is recomputed independently for each frame, and we discard frames where $\|\hat{t}\|_2^2$ falls below a small threshold (indicating an unreliable baseline prediction).

Metric-Scale Trailer Poses. Once the scale factor s is obtained for a given frame, we apply it to all trailer-mounted cameras. For any trailer camera $j \in \{\text{rear}, \text{rear-left}, \text{rear-right}\}$, we first compute its baseline relative transform to the tractor front camera,

$$\hat{T}_{F\to j} = (\hat{T}_j)^{-1} \hat{T}_F = (\hat{R}_{F\to j},\, \hat{t}_{F\to j}), \tag{4}$$

and then construct a metric-scale relative transform by rescaling only the translation:

$$\widetilde{T}_{F\to j} = (\hat{R}_{F\to j},\, s\,\hat{t}_{F\to j}). \tag{5}$$

Finally, we map this metric-scale relative pose back to the world frame using the ground-truth tractor front extrinsic:

$$\widetilde{T}_j = T^\star_F\, \widetilde{T}^{-1}_{F\to j}. \tag{6}$$

We extract the translation $\widetilde{t}_j$ and rotation $\widetilde{R}_j$ from $\widetilde{T}_j$ and use them as the metric-scale trailer camera extrinsics when evaluating VGGT and DUSt3R. The same procedure is applied to the trailer rear-left and rear-right cameras, using the known rigid intra-trailer transforms to maintain consistency.

B. Dataset

Sensor Setup. Figure 2 illustrates the placement of all onboard sensors. The tractor is equipped with three forward-facing RGB cameras and a roof-mounted 128-beam LiDAR. The trailer carries a rigid tri-camera rig mounted at

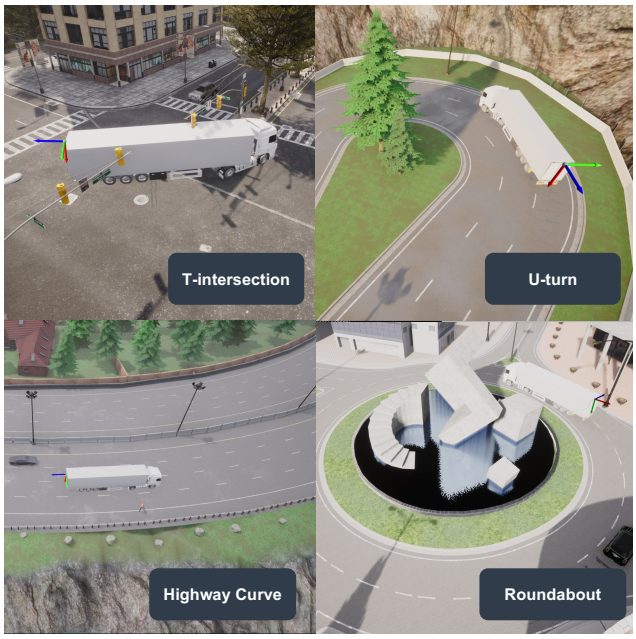

Figure 1. Representative articulated-driving scenarios in the STT4AT dataset. Each example shows the tractor–trailer configuration and trailer-mounted camera poses under typical high-articulation maneuvers.

Figure 2. Placement of the sensors and their corresponding coordinate systems from a top-down view perspective

its rear. The extrinsics of tractor-mounted sensors remain fixed throughout each sequence, while the trailer-mounted rig undergoes time-varying motion due to articulation.

Visualizations of Articulated Scenarios. Figure 1 shows representative articulated driving scenarios included in the dataset. The collection spans common maneuvers such as T-intersections, U-turns, highway curves, and roundabouts. These scenes cover a range of articulation magnitudes and occlusion patterns, providing diverse geometric conditions for evaluating calibration and perception models.

C. Runtime and Memory

This section reports the computational footprint of the proposed model, including GPU memory usage, inference latency, and the number of learnable parameters.

Training. Since the VGGT encoder remains frozen during training, only the CCA module, the CTA module, and the modulation–refinement head contribute to parameter updates. This design eliminates the need for heavy prediction branches such as depth estimation or dense point generation, resulting in a lightweight and computationally efficient framework. Table 1 lists the number of trainable parameters for each component.

Module	#Params (M)	Trainable
VGGT backbone	909.1	✗
Camera Cross-Attention (CCA)	16.8	✓
Camera Temporal Self-Attention (CTA)	16.8	✓
AdaLN-guided pose refinement module	216.2	✓

Table 1. Number of parameters for core modules in dCAP.

During training, CTA requires the previous global token for temporal fusion. We randomly sample three consecutive frames; when no temporal neighbors exist (e.g., the first frame of a sequence), the model falls back to using the current frame's own global token.

Inference. All experiments are performed on a single NVIDIA RTX A6000. dCAP with CCA requires 8.5 GB of GPU memory during inference, while dCAP with CTA requires 10 GB. The inference time is 1.2 s per frame for both settings. CTA does not increase latency because temporal self-attention uses only a cached global token from the previous frame, and maintaining this cache lies outside the inference-time critical path.

D. More Results

More visual results. In Figure 3, Figure 4, Figure 5, and Figure 6, we demonstrate more visualization results of our method for different scenarios.

E. Limitations and Future Work

While STT4AT is equipped with synchronized multi-modal sensors supporting tasks from mapping to end-to-end driving, this paper only focuses on establishing a baseline for 3D object detection. Moreover, the potential of the dataset for other downstream tasks (tracking, motion forecasting, mapping, and path planning) are still underexplored.

Furthermore, despite covering 87 scenes, the current dataset scale may limit generalization to long-tail articulated behaviors and rare environmental conditions. In real-world operations, a single tractor often couples with diverse trailer types, a complexity only partially captured here.

Future work will extend the benchmark to additional downstream tasks that are central to articulated navigation. We will also increase the variety of trailer configurations and scenarios to better support research on dynamic articulation.

References

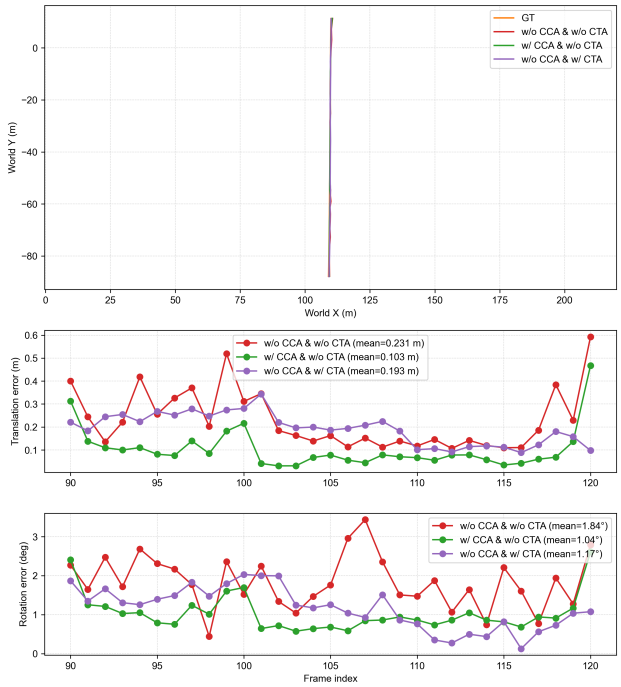

Figure 3. Qualitative results comparison between different attention modules under the straight scenario.

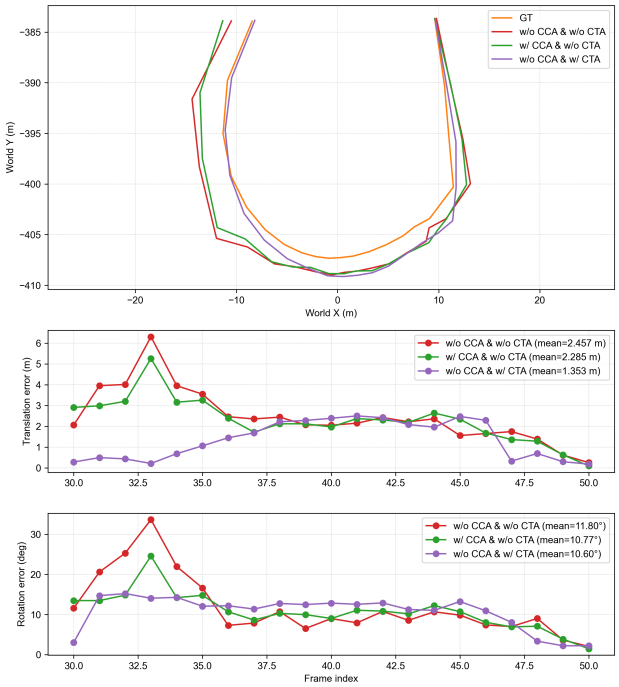

Figure 5. Qualitative results comparison between different attention modules under the u-turn scenario.

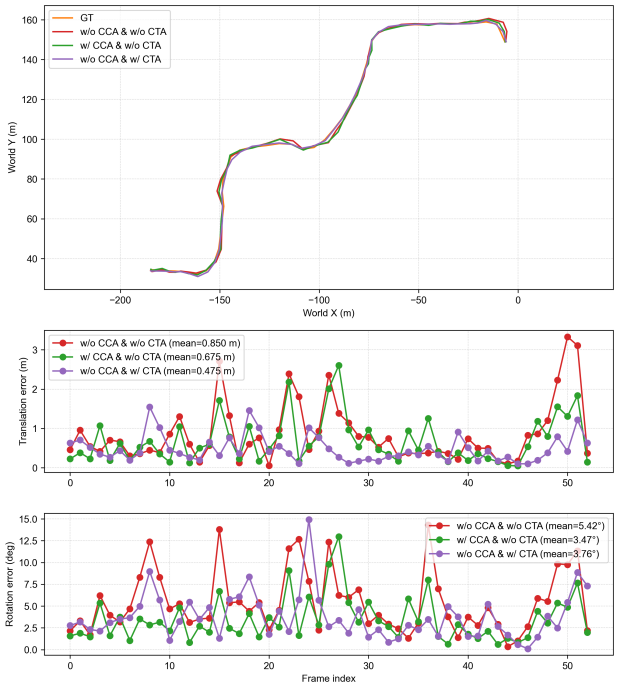

Figure 4. Qualitative results comparison between different attention modules under the roundabout scenario.

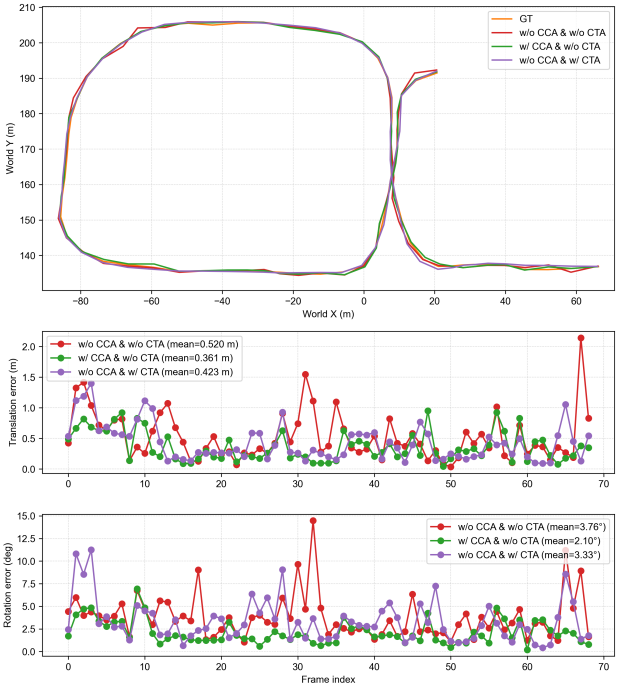

Figure 6. Qualitative results comparison between different attention modules under the multi-turn scenario.