VGGT-Segmentor: Geometry-Enhanced Cross-View Segmentation

Supplementary Material

1. More Implementation Details

1.1. K-Means Center Correction

The vanilla K-Means algorithm computes cluster centers as the arithmetic mean of all assigned points. When applied to foreground masks, this may place cluster centers on background regions rather than inside the object (such as a ring-shaped mask), which in turn misguides the Point-Guided Prediction stage and leads to incorrect mask generation. To address this issue, we explicitly verify whether each center lies within the mask. If a center falls outside, we pull it back to the nearest foreground mask pixel, ensuring all cluster centers remain valid geometric representatives of the source mask.

1.2. Point Perturbation

During pretraining under the Single-Image Self-Training strategy, the VGGT-non-adaptive augmentation family introduces strong geometric distortions that invalidate VGGT's point mappings. In this setting, we sample representative points from the ground-truth mask in the target view using K-Means, then generate perturbations by sampling random offsets within a 50-pixel radius around each point. The perturbed points are used as synthetic projections of VGGT, enabling the model to learn to handle severely misaligned or corrupted point prompts. The VGGT-non-adaptive and adaptive strategies are scheduled randomly, with an overall 1:1 ratio between them.

1.3. Precision Setup

The VGGT image encoder outputs features in `torch.bfloat16` by default. However, our Union Segmentation Head exhibits noticeably improved training stability when implemented in `torch.float32`. In our experiments, using `torch.bfloat16` during training leads to non-convergent losses, so we adopt `torch.float32` for the entire training pipeline. For consistency between training and inference, we also run inference fully in `torch.float32`. This ensures numerical stability and prevents subtle degradation that may arise from mixing precision formats.

1.4. Crop and Remapping Strategy

Training Stage. The source view is cropped to one-half of the original image size, centered on the target object. Cropping is applied to the target view only when it is an Exo view, where the image is cropped to approximately two-thirds of the original size, also centered on the target object.

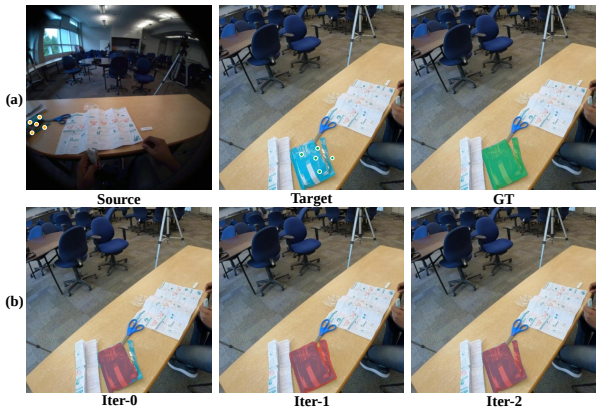

Figure 1. **Visualization of Mask Refinement Iterations.** (a) The source view, VGGT's projected point in the target view, and the ground-truth mask. (b) Iterative mask predictions produced by VGGT-S across refinement steps. Zooming in provides better results.

The cropping is based on the full target object information available during training.

Inference Stage. The strategy is consistent with training, and cropping is applied to the target view only when it is an Exo view. Specifically, a **Remapping Strategy** is adopted: the Exo image is first padded to a square and an initial target point is predicted. Although this prediction may be affected by padding and is not highly precise, it provides a coarse spatial estimate. Based on this initial prediction, the Exo image is then cropped into a square region centered at the predicted point, with the side length equal to the image width. A second round of target point prediction is then performed on the cropped image. This procedure introduces only a small and controllable computational overhead, requiring just one additional inference pass of the VGGT track head.

2. Effectiveness of the Mask Refinement Stage

The mask refinement stage is shown in Figure 2. It primarily improves the predicted masks by sharpening their boundaries and filling missing regions caused by occlusions or fragmented initial predictions, as illustrated in Figure 1. The first refinement step provides the most substantial improvement by merging disjoint mask fragments into a complete region, while the second refinement further polishes the boundaries to produce more precise masks.

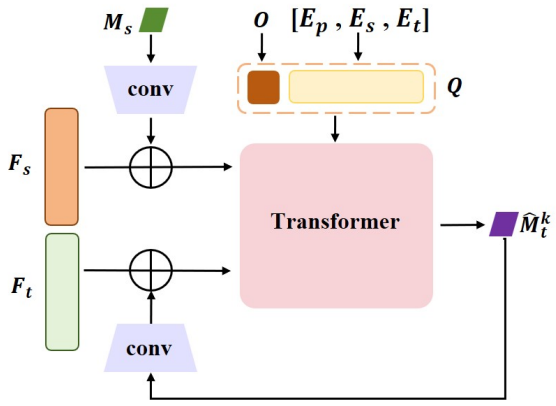

Figure 2. **Mask Refinement Module of VGGT-S.** This module iteratively refines the coarse target mask by utilizing geometry-aware features, source mask, prompt queries and last predicted mask.

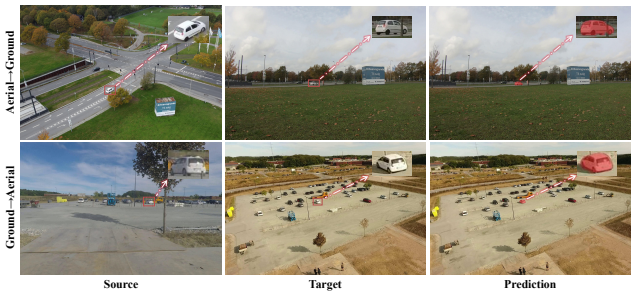

Figure 3. **Visualization of the Outdoor Dataset.** The top row shows the Aerial→Ground task and the bottom row shows the Ground→Aerial task. Zooming in provides better results.

3. Test on MAVREC

Figure 3 compares cross-view segmentation on challenging aerial–ground pairs from the MAVREC dataset. It includes challenging outdoor scenes with large-scale changes, cluttered backgrounds, and varying illumination. The target objects undergo dramatic appearance and scale changes across viewpoints. Even without any fine-tuning, our correspondence-free pretrained model successfully localizes and segments the correct targets across these extreme perspective shifts, demonstrating strong robustness and powerful generalization to viewpoint variation and outdoor scene complexity.

4. Multi-View Experiment

In the multi-view setting, the task becomes predicting object masks across multiple target views given a single source-view mask, which is inherently more challenging than predicting a single-view mask. However, the multi-view context also brings additional benefits to VGGT-S. Since VGGT is naturally stronger at modeling scenes with multiple viewpoints rather than just two, providing more valid views enables VGGT to form a richer understanding of the underlying 3D structure. As a result, its image feature becomes more accurate and its point-tracking becomes more reliable, further enhancing the mask prediction capability of our Union Segmentation Head.

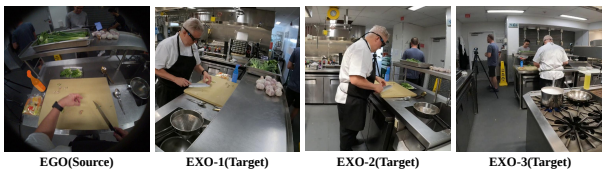

Figure 4. **Visualization of the multi-view experiment.** The tracked object is a sauce bottle. Under this multi-view setting, VGGT-S remains robust, effectively correcting projection errors and producing consistent mask predictions.

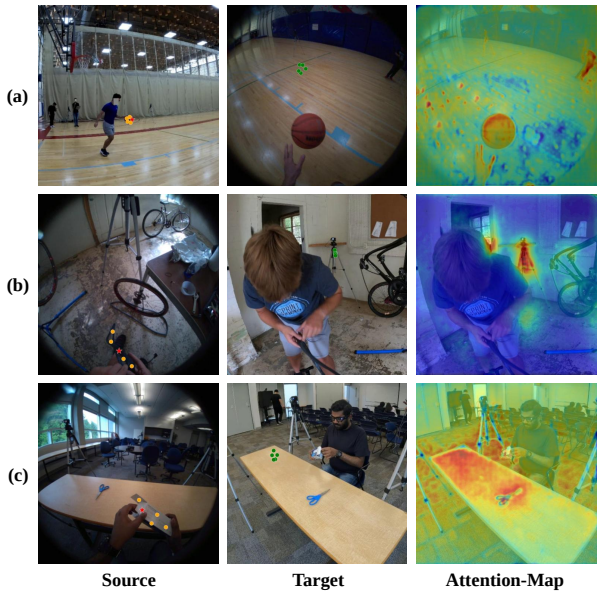

Figure 5. **Visualization of Limitations.** (a) VGGT produces inaccurate point projections on the basketball when the two views are nearly orthogonal. (b) VGGT struggles to project points on the tire under opposing views in a highly complex scene, with attention maps indicating that the extracted features are highly likely to mislead the mask prediction. (c) The same box appears with markedly different color and shape across the two views, reflecting its obverse and reverse sides, which causes VGGT to misproject points. Zooming in provides better results.

VGGT-S can be directly applied to the multi-view scenario **without** any **additional training**. Concretely, all available views are processed jointly by the VGGT encoder, producing stronger image features and more reliable point locations. In the decoder stage, each target view is paired with the source view to form an independent ego–exo pair, and the decoder predicts the mask for each pair separately,

Table 1. Model architecture with convolutional and transformer stages.

Stage	Module	Type	Kernel/Head	Dim / Channels	Depth
Prompt Encoder	Mask Encoder	Conv	2×2, stride 2	128	1
Transformer	Feature Downsample	Conv	7×7, stride 7	128×7	$\times 2$
	Feature Upsample	ConvTranspose	7×7, stride 7	128	$\times 2$
	Image Feature Self-Attn	Attention	8	128×7	$\times 2$
	Image Feature FFN	MLP	–	128×7	$\times 2$
	Cross-Attn	Attention	8	128	$\times 2$
	Query Self-Attn	Attention	8	128	$\times 2$
Mask Decoder	Mask Upsample	ConvTranspose	2×2, stride 2	128 / 8	$\times 2$
	Query MLP	MLP	–	128 / 8	1

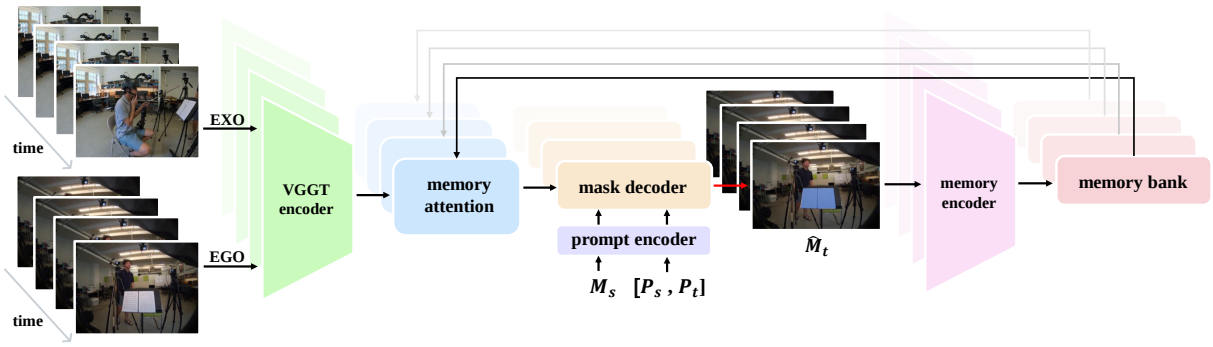

Figure 6. **Architecture of Temporal VGGT-S.** Temporal VGGT-S augments VGGT-S with a memory encoder, memory bank, and memory attention module. Predicted masks from previous frames are encoded and stored, then injected into the current frame via cross-attention to provide temporal object cues and stabilize cross-view mask prediction.

rather than feeding all views into the decoder simultaneously. This design allows us to fully leverage the existing architecture and training pipeline while generalizing seamlessly to multi-view conditions. We visualize predictions in Figure 4.

5. Model Architecture

Detailed model parameters, including convolutional layer sizes and model depth, are provided in Table 1 for details.

6. Limitations

Although VGGT provides strong 3D modeling capabilities, there remain certain challenging scenarios where its point-tracking performance becomes unreliable. These challenges generally fall into two categories. First, when the ego and exo views are oriented in opposite or approximately orthogonal directions, the two views share very limited scene overlap, making it inherently difficult for VGGT to establish stable geometric structure (as shown in Figure 5 (a)). In addition, highly complex or cluttered environments can further complicate geometric reasoning, as illustrated in Figure 5 (b), where the internal attention mechanism becomes inaccurate and offers limited guidance. Second, substantial appearance changes of the same object across views, including variations in shape or color, may also hinder VGGT's correspondence estimation, as shown in Figure 5(c).

In such cases, the geometric cues provided by VGGT may not be sufficiently reliable to support our Union Segmentation Head, which naturally limits the quality of the final mask predictions. Fundamentally, these scenarios highlight the inherent difficulty of 3D scene modeling under extreme viewpoint changes or severe appearance variations. We believe that future advances in stronger 3D reasoning backbones or more robust geometric representations could further alleviate these limitations.

7. Future Work

For cases with occlusion or partial visibility, where single-frame spatial information cannot capture relationships among different parts of the same object, or in dynamic scenes with fast motion and motion blur that destabilize VGGT matching, single-frame dual-view prediction often fails. Temporal information is helpful, as historical frames can stabilize mask predictions over time while processing a video.

We design the **Temporal VGGT-S**, with its architecture shown in Figure 6. Specifically, we propose a memory

encoder that encodes each predicted mask and stores it in a memory bank. For the next frame, the VGGT encoder features are enhanced via cross-attention with the memory bank, injecting temporal object cues into the image features. Because the features are fused with predicted masks, this process naturally performs a form of mask refinement when consecutive frames are visually similar.

This pipeline enables VGGT-S to leverage temporal information in videos, improving cross-view segmentation accuracy.