EgoRoC: Towards Egocentric Robotic Control via Task-Agnostic Visual Alignment

Supplementary Material

6. EVAM Initialization Component Training

As illustrated in Fig. 2 in the main paper, within EVAM's inference initialization component, the trainable modules include a **LoRA adapter** [19] for parsing object entities and temporal dependencies, and a **pix2pix-turbo** [40] model responsible for inpainting the top-view images rendered from pointmaps by the renderer.

LoRA Adapter. We construct a dataset containing 7k training samples. For each task, the corresponding object entities and temporal dependencies are generated by GPT-4o [20] to act as supervision signals.

Pix2pix-turbo. We construct a dataset of 20k samples to fine-tune pix2pix-turbo, using Gemini 2.5 Flash Image [15]. During the fine-tuning process, the input image is the top-view rendered image of the pointmap, and the supervision signal is the output generated by Gemini 2.5 Flash Image.

7. Inference Pipeline Details of EVAM

7.1. Spatio-temporal Relation Extraction

We integrate the LoRA adapter trained in Sec. 6 with the LMM to extract spatiotemporal dependencies, where the input comprises third-person images and textual prompts. The prompt is formulated identically to the template used during training, as follows:

$< image >$ *Analyze the visual content of the provided image thoroughly. Given the task* ***{Task Description.}****, identify all core objects directly involved in completing this task, then determine their logical execution order (1 = first interacted object, 2 = second interacted object, etc.). Output ONLY a valid JSON string parsable by Python's json.load() function, no extra text, explanations, or formatting. Example:* $\{$*"carrot": 1, "plate": 2*$\}$

7.2. Top-View Rendering Strategy

First, we employ Grounded-SAM-2 [43] to segment the objects requiring manipulation from the third-person image. The corresponding point sets of these objects and the desktop are then extracted from the pointmap generated by VGGT [50]. For point cloud processing, we adopt an improved median filter combined with centroid method: we sort each coordinate dimension (X, Y, Z) of the object point cloud, truncate 10% of extreme values at both ends to eliminate outliers, and calculate the mean of the remaining inliers as the object center. We then fit the desktop plane using the inlier points of the desktop point set after the same outlier rejection. Finally, we determine the camera pose based on the maximum distance from inlier points to each object center (as the scale reference) and the direction facing the objects from above the desktop, and render the top-view image accordingly.

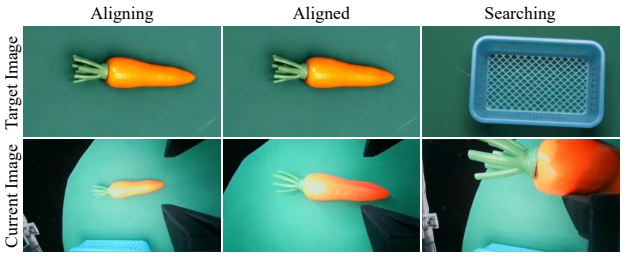

Figure 6. Visual sample of the *align state token*.

7.3. Align State Token

Figure 6 illustrates three types of align state token: **aligning** (*object in the field of view but unaligned*), **aligned** (*object in the field of view and aligned*), and **searching** (*object outside the field of view*). When EVAM outputs *searching*, a 45° rotational search is executed around the last joint of the robotic arm until the object re-enters the field of view. Additionally, during initialization, the robotic arm moves a short distance to supply an initial state for DHCM.

8. Architecture of Diffusion Model

As the four-point generation task is simple for Diffusion models, we use multiple Concatsquash MLPs [16] to implement our Diffusion-based online Hand-eye Calibration Module. As shown in Fig. 7, once we obtain the hand-eye relationship descriptor $\mathbf{CondFeature}$, the diffusion model $\mathcal{D}_\theta$ generates the spatial points $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$, which encoded the hand-eye transformation matrix $\mathbf{X}$. Note, after the diffusion process, a Geo-MLP layer enforces physical plausibility of the generated point cloud under the supervision of Eq. (6)–(8) in the main paper.

9. Runtime Analysis

Before model deployment, an initialization phase is required with a total latency of approximately 0.62 seconds. This phase includes 0.23 seconds for spatiotemporal relation extraction by the LMM, 0.13 seconds for VGGT point cloud generation, 0.15 seconds for the top-view rendering

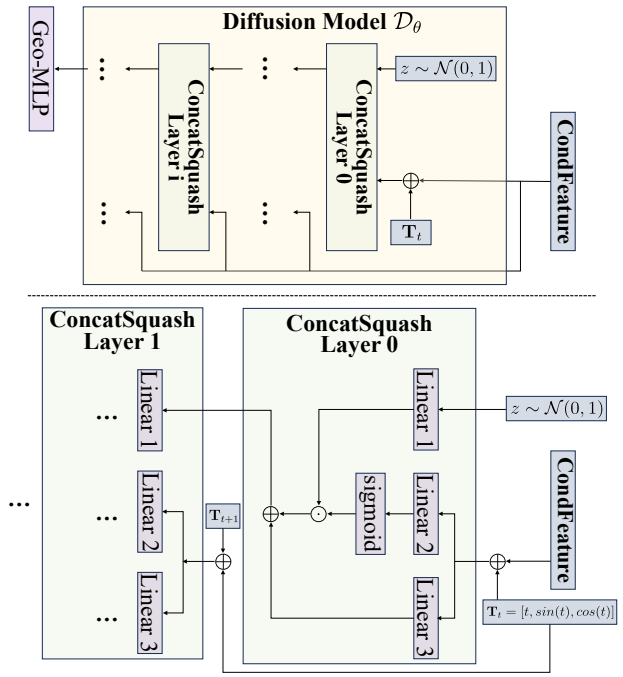

Figure 7. The architecture of the diffusion model in our EgoRoC.

strategy, and 0.11 seconds for inpainting. During the inference phase, the model operates at an approximate speed of 1 Hz, encompassing Grounded-SAM-2 processes the current image to eliminate interference from the gripper and other objects in 0.14 seconds, 0.29 seconds for EVAM module, 0.19 seconds for COTR within DHCM, 0.16 seconds for multimodal fusion, and 0.21 seconds for the diffusion process.