# ALIGNTRACK: ALIGNING SEMANTICS, GEOMETRY, AND ACTIONS FOR EMBODIED VISUAL TRACKING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Embodied visual tracking (EVT) requires agents to jointly understand target semantics, reason about spatial relations, and generate actions for continuous target following. While vision-language-action (VLA) models have shown strong potential for EVT, their implicit action prediction leaves semantic, geometric, and action reasoning insufficiently coupled. To address this issue, we propose AlignTrack, a lightweight VLA that explicitly bridges these representations through a semantic-aligned depth encoder and an action-relevant understanding decoder. A two-stage strategy first aligns depth features with pretrained visual-semantic representations and then jointly supervises target-centric spatial understanding and action prediction. Extensive experiments on EVT-Benchmark show that AlignTrack with only a 0.6B backbone achieves performance comparable to the 7B TrackVLA, demonstrating the effectiveness of explicit spatial knowledge for lightweight embodied tracking. The code and trained models will be released publicly.

## 1 INTRODUCTION

Embodied visual tracking (EVT) requires an agent to continuously perceive a specified target and actively control its motion to maintain effective observation. As a fundamental capability for embodied agents, EVT benefits a wide range of applications, including person following, service robotics, search and rescue, and human-robot interaction. (Ye et al., 2023; Yong et al., 2022; Nivashini & Rajesh, 2026; Zender et al., 2007) Unlike passive visual tracking, EVT tightly couples target understanding with spatial reasoning and action generation, requiring the agent to determine not only *what* to track, but also *where* the target is and *how* to move accordingly.

Recent advances in vision-language-action (VLA) models (Zhang et al., 2025; Wang et al., 2025; Liu et al., 2026; Zhang et al., 2026) have enabled embodied tracking by leveraging pretrained multimodal representations to directly translate visual observations and language instructions into actions. However, existing VLAs largely treat action prediction as an implicit process, leaving unclear how semantic understanding and geometric reasoning are organized and translated into executable actions. Such implicit perception-to-action coupling makes it difficult to explicitly exploit spatial knowledge and further improve the design of tracking VLAs.

An intuitive solution is to incorporate expert semantic and geometric representations into the VLA by concatenating them with visual tokens. However, these independently learned representations often reside in different feature spaces, making direct fusion disrupt the pretrained visual-semantic representation. End-to-end co-training must therefore jointly resolve representation alignment, spatial reasoning, and action prediction, potentially compromising the pretrained reasoning capability.

To address this issue, we propose AlignTrack, a lightweight VLA framework that explicitly bridges semantic understanding, geometric perception, and embodied action through two propsed components including a *semantic-aligned depth encoder* for geometric integration and an *action-relevant understanding decoder* for explicit spatial grounding. Accordingly, we adopt a two-stage strategy of feature-alignment pretraining and co-training. At first, a lightweight depth encoder is aligned with frozen RGB representations through token- and distribution-level objectives, enabling geometric integration without disrupting pretrained semantics. Then, spatial and action queries are jointly optimized to predict target states such as bounding box, relative yaw, and visibility and future trajectories, explicitly connecting *what to track*, *where the target is*, and *how to move*.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Last, extensive experiments on EVT-Benchmark demonstrate the effectiveness of AlignTrack, which achieves performance comparable to the 7B TrackVLA with only a 0.6B backbone. The gains from our semantic-aligned depth encoder and action-relevant understanding decoder demonstrate that explicit spatial knowledge can compensate for limited multimodal reasoning, offering an alternative to scaling up VLA models for embodied tracking.

Our main contributions are summarized as follows:

- We propose AlignTrack, a lightweight VLA framework that explicitly bridges semantic understanding, geometric perception, and embodied actions for embodied visual tracking.
- We introduce a *semantic-aligned depth encoder* and an *action-relevant understanding decoder*, which progressively align geometric representations with pretrained visual semantics and explicitly couple spatial understanding with action prediction through feature-alignment pretraining and joint co-training.
- Extensive experiments on EVT-Benchmark show that our 0.6B model achieves performance comparable to the 7B TrackVLA, demonstrating the effectiveness of explicit spatial knowledge for lightweight embodied tracking.

## 2 Related Work

### 2.1 Passive Visual Tracking

Visual tracking has traditionally focused on estimating target states from passively observed video streams without actively controlling the camera or the observing agent. In single-object tracking (SOT), a target specified in an initial frame is continuously localized in subsequent observations. Siamese trackers such as SiamFC (Bertinetto et al., 2016) and SiamRPN (Li et al., 2018) learn target–search matching representations, while more recent Transformer-based approaches such as OSTrack (Ye et al., 2022) jointly model feature extraction and target–search interaction within a unified architecture. In multi-object tracking (MOT), the emphasis shifts toward maintaining multiple target identities over time. SORT (Bewley et al., 2016) combines Kalman filtering with detection association, DeepSORT (Wojke et al., 2017) further incorporates appearance embeddings for identity preservation, and MOTR (Zeng et al., 2022) formulates temporal association through learnable track queries in an end-to-end Transformer.

These approaches have substantially advanced appearance matching, temporal association, and target localization. However, passive trackers primarily estimate image-space target states and do not determine how an embodied agent should move to maintain observation. As a result, tracking perception and physical control remain separated, motivating active tracking methods that jointly reason about target motion and agent actions.

### 2.2 Reinforcement-Learning-based Active Tracking

Visual active tracking extends passive perception by requiring an agent to actively control its motion and viewpoint to keep a moving target observable. Early work by Luo et al. (Luo et al., 2018) formulated active object tracking as an end-to-end reinforcement learning problem, directly mapping visual observations to camera-control actions. AD-VAT (Zhong et al., 2019) further introduced an asymmetric adversarial learning scheme in which a learnable target and tracker compete during training, improving robustness to previously unseen target motion. More recently, Zhong et al. (Zhong et al., 2024) combined visual foundation models with offline reinforcement learning, using foundation-model-derived visual representations to improve sample efficiency and generalization in embodied visual tracking.

Active tracking has also been extended from ground agents to aerial platforms. D-VAT (Dionigi et al., 2024) learns an end-to-end policy for micro aerial vehicles that maps monocular visual observations directly to thrust and angular-velocity commands, enabling closed-loop aerial target following. These reinforcement-learning approaches establish the crucial perception–action coupling absent from passive tracking. Nevertheless, target semantics and action generation are typically not jointly modeled through a pretrained vision-language backbone, and spatial knowledge is mainly acquired implicitly from interaction rewards rather than explicitly represented and supervised. This

becomes increasingly limiting when an agent must follow a language-specified target while reasoning about target-relative geometry, occlusion, and surrounding obstacles.

### 2.3 Vision-Language-Action Tracking and Spatial Reasoning

Recent Vision-Language-Action (VLA) models provide a unified framework for grounding language-conditioned perception into embodied actions. Uni-NaVid (Zhang et al., 2025) unifies several navigation tasks, including language-conditioned human following, within a video-based VLA policy. TrackVLA (Wang et al., 2025) further specializes this paradigm for embodied visual tracking by jointly modeling target recognition and trajectory planning with a shared language backbone. These works demonstrate that pretrained vision-language representations provide strong semantic priors for identifying and following language-specified targets.

A growing body of research, however, has shown that semantic representations inherited from vision-language models are insufficient for precise physical interaction without explicit spatial grounding. 3D-VLA (Zhen et al., 2024) incorporates 3D perception and generative world modeling to connect language reasoning with physical scene structure. SpatialVLA (Qu et al., 2025) introduces egocentric 3D position encoding and spatially structured action representations, demonstrating the importance of explicitly modeling geometry for VLA policies. Spatial Forcing (Li et al., 2026a) takes a representation-level approach by aligning intermediate VLA visual features with geometric representations from pretrained 3D foundation models. Depth-based geometry provides another direct source of spatial information. DepthVLA (Yuan et al., 2026) incorporates a pretrained depth module into the VLA architecture, while QDepth-VLA (Li et al., 2026b) introduces quantized depth prediction as an auxiliary objective, encouraging the shared model to learn depth-aware representations for action generation.

Explicit spatial reasoning is also becoming increasingly important for embodied tracking. UAV-Track VLA (Zhang et al., 2026) introduces a spatial-aware auxiliary grounding head alongside continuous action prediction for aerial tracking. TrackVLA++ (Liu et al., 2026) explicitly predicts target-relative spatial relations through Polar-CoT and combines them with target identification memory. ReferTrack (Ye et al., 2026) grounds target selection in image-space bounding boxes and propagates their geometric information through temporal observations. CosFly-VLA (Ren et al., 2026) further incorporates depth, distance, and 3D spatial reasoning during continued pretraining and jointly predicts target grounding, visibility, and flight actions.

Together, these studies highlight spatial grounding as a central component of reliable VLA control. However, most depth- and geometry-enhanced VLAs are developed for robotic manipulation, while recent tracking-oriented approaches primarily introduce explicit grounding, spatial reasoning tokens, or large-scale spatial pretraining. This motivates a concrete design question: how can dense depth observations be aligned with pretrained visual-semantic features and connected to target-centric action prediction in a compact VLA? To solve this question, we first align a lightweight depth branch with the frozen RGB visual-semantic representation before joint VLA training, and further supervise the shared backbone with target-centric spatial predictions, including target localization, relative orientation, and visibility. This design integrates semantic-aligned depth features and target-centric spatial supervision within a compact tracking VLA.

## 3 Method

AlignTrack addresses the two design questions raised in the introduction: how to place depth features in the representation space used by a pretrained visual pathway, and how to make the resulting geometry useful for target-centric action prediction. Figure 1 summarizes the model. We first align the depth branch to spatially corresponding RGB tokens. We then jointly train the multimodal backbone with a target-state query and a trajectory query. This section introduces the prediction task and the information flow; Appendix A gives the individual loss terms and decoder details.

### 3.1 Task and model overview

At time $t$, the agent receives a language instruction $I$, an $H$-frame RGB observation sequence $\mathcal{O}_t = (x_{t-H+1}, \ldots, x_t)$, and the current depth observation $d_t$. The policy predicts $K$ future agent-centric

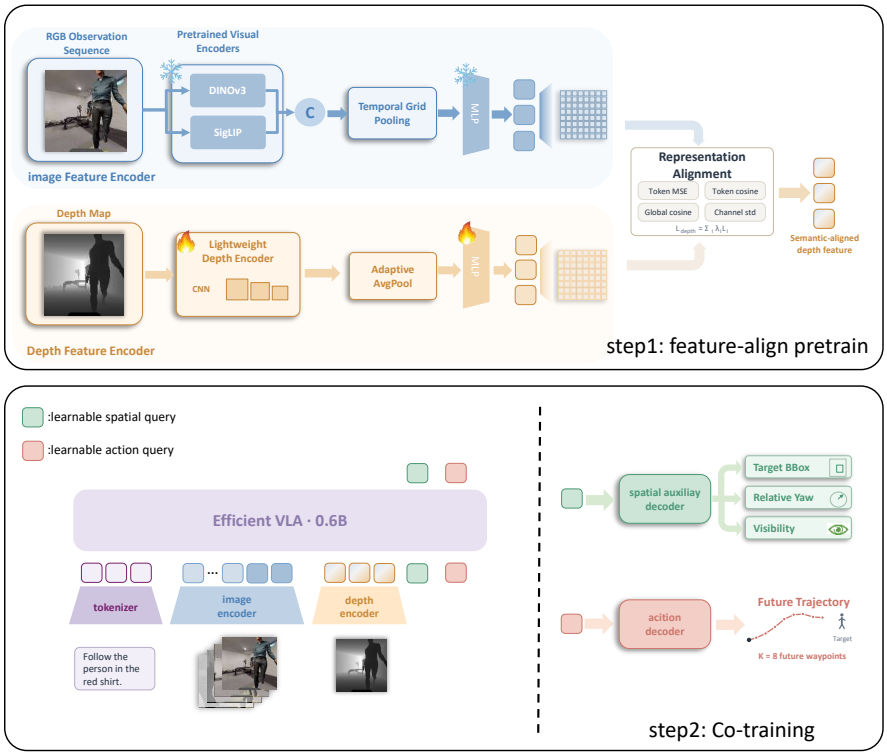


Figure 1: Caption

waypoints,

$$\hat{\mathcal{Q}}_t = (\hat{q}_{t,1}, \ldots, \hat{q}_{t,K}), \qquad \hat{q}_{t,k} = (\hat{x}_{t,k}, \hat{y}_{t,k}, \hat{\theta}_{t,k}). \tag{1}$$

Here $(x, y)$ denotes planar displacement and $\theta$ denotes the heading coordinate of a future waypoint. The policy uses the target specified by $I$ to predict waypoints that maintain target observation during closed-loop tracking. At each decision step, the depth branch encodes one 64-token grid from the current depth frame.

The RGB pathway supplies visual context for the language-specified target, while a separately projected depth pathway supplies spatial structure. A spatial query predicts the target bounding box, relative yaw, and visibility. A subsequent action query predicts the waypoint sequence. Under causal attention, the action query can attend to the spatial query and all preceding observation tokens. The target-state predictions provide auxiliary supervision, and the action query can condition on the spatial query's hidden representation.

### 3.2 Visual and Depth Tokens

For each RGB frame $x_i$, frozen DINOv3 and SigLIP encoders produce spatially indexed features. We concatenate corresponding patch features along the channel dimension to obtain $R_i \in \mathbb{R}^{N \times C}$, where $N = 256$ and $C$ is the combined feature width. We retain an $8 \times 8$ grid for the current frame and use a $2 \times 2$ grid for each historical frame:

$$R_t^{\text{cur}} = \text{Pool}_{8\times8}(R_t), \qquad R_i^{\text{hist}} = \text{Pool}_{2\times2}(R_i) \quad (i < t), \tag{2}$$

$$E_t^R = P_R\big([R_{t-H+1}^{\text{hist}}; \ldots; R_{t-1}^{\text{hist}}; R_t^{\text{cur}}]\big). \tag{3}$$

Thus the current RGB observation contributes 64 tokens, and each historical observation contributes four. $P_R$ maps these tokens into the language-backbone embedding space.

We clip metric depth at 10 m and normalize it to obtain $\bar{d}_t$. A lightweight convolutional encoder $f_D$ extracts depth features; adaptive pooling produces an $8 \times 8$ grid, and an independent projector $P_D$

maps its 64 tokens into the same embedding dimension $D$ as the RGB tokens:

$$E_t^D = P_D\big(\mathrm{Flatten}\big(\mathrm{Pool}_{8\times 8}(f_D(\bar{d}_t))\big)\big) \in \mathbb{R}^{64\times D}. \tag{4}$$

The two projectors retain modality-specific mappings. The representation-alignment stage in Section 3.4 initializes the depth pathway before joint training.

### 3.3 Target-state and action prediction

Let $\mathrm{Emb}(I)$ be the instruction tokens. We feed the backbone the sequence

$$S_t = [\mathrm{Emb}(I); E_t^R; E_t^D; p^{\mathrm{sp}}; p^{\mathrm{act}}]. \tag{5}$$

Here $p^{\mathrm{sp}}$ and $p^{\mathrm{act}}$ are learnable spatial and action query tokens. Denote their output hidden states by $h_t^{\mathrm{sp}}$ and $h_t^{\mathrm{act}}$, respectively. The spatial query is decoded into a normalized target bounding box $\hat{b}_t$, a target-relative yaw $\hat{\psi}_t$, and a visibility logit $\hat{s}_t$. The action query is decoded into all $K$ future waypoints in parallel:

$$(\hat{b}_t, \hat{\psi}_t, \hat{s}_t) = g_{\mathrm{sp}}(h_t^{\mathrm{sp}}), \qquad \hat{\mathcal{Q}}_t = g_{\mathrm{act}}(h_t^{\mathrm{act}}). \tag{6}$$

Here $g_{\mathrm{sp}}$ denotes a shared spatial MLP with task-specific output heads, and $g_{\mathrm{act}}$ includes the trajectory MLP, output activation, and waypoint scaling specified in Appendix A. We use $\psi$ for the target-relative yaw and $\theta$ for the agent waypoint heading.

### 3.4 Depth-to-RGB representation alignment

The depth branch must produce tokens that can be fused with the pretrained RGB pathway. For paired RGB and depth observations, we form a reference grid $E_t^{R,\mathrm{cur}} = P_R(R_t^{\mathrm{cur}}) \in \mathbb{R}^{64\times D}$ and compare it with $E_t^D \in \mathbb{R}^{64\times D}$. Each token pair refers to the same pooled image location. The RGB encoders and RGB projector are fixed during this stage; gradients update the depth encoder and depth projector only. The alignment loss combines token values, token directions, a pooled global direction, and channel-wise variation:

$$\mathcal{L}_{\mathrm{align}} = \lambda_{\mathrm{mse}}\mathcal{L}_{\mathrm{mse}} + \lambda_{\mathrm{cos}}\mathcal{L}_{\mathrm{cos}} + \lambda_{\mathrm{global}}\mathcal{L}_{\mathrm{global}} + \lambda_{\mathrm{std}}\mathcal{L}_{\mathrm{std}}. \tag{7}$$

The terms and their normalization are defined in Appendix A. Token-level terms preserve spatial correspondence; the global term matches the mean feature direction; and the standard-deviation term matches variation across spatial tokens. The resulting depth parameters initialize the depth pathway for joint training. This alignment is an initialization: the RGB projector and depth pathway may both adapt in the subsequent stage.

### 3.5 Joint spatial and trajectory learning

Joint training optimizes the backbone, both projectors, the depth encoder, and both prediction heads, while keeping DINOv3 and SigLIP frozen. The trajectory objective compares predicted waypoints with valid future waypoints after coordinate-wise scaling. The spatial objective supervises the target bounding box, relative yaw, and visibility. We optimize

$$\mathcal{L}_{\mathrm{joint}} = \beta_{\mathrm{nav}}\mathcal{L}_{\mathrm{nav}} + \beta_{\mathrm{sp}}\big(\lambda_b\mathcal{L}_{\mathrm{bbox}} + \lambda_\psi\mathcal{L}_{\mathrm{yaw}} + \lambda_v\mathcal{L}_{\mathrm{vis}}\big). \tag{8}$$

The alignment loss is not applied during this stage. The auxiliary target-state losses make target location, orientation, and visibility explicit training signals for the shared representation, while the trajectory loss trains the action query to use that representation for control. Detailed losses and valid-label handling appear in Appendix A.

## 4 Experiment

### 4.1 Experimental Setup

**Model Configuration.** We implement AlignTrack using Qwen3-0.6B as the language backbone. The RGB observations are encoded by the pretrained DINOv3-ViT-S/16 and SigLIP SO400M vision towers. Both vision towers are kept frozen, and their features are extracted offline at a spatial

resolution of $384 \times 384$. The resulting DINOv3 and SigLIP features have dimensions 384 and 1152, respectively, and are concatenated into a 1536-dimensional RGB representation.

For each training sample, we use a temporal observation horizon of $H = 32$. Each historical RGB frame is represented by four coarse tokens, while the current frame is represented by 64 fine-grained tokens. The depth encoder produces 64 tokens with an intermediate feature dimension of 256. The action decoder predicts $K = 8$ future waypoints. The maximum instruction length is set to 128 tokens. Since all experiments use a single egocentric camera, the optional angle-based temporal-view embedding is disabled.

Depth values are clipped to a maximum range of 10 meters and normalized to $[0, 1]$. During depth pretraining, the input depth maps are resized to $384 \times 384$. The RGB token features are stored in half precision after offline extraction, while the trainable projectors and prediction heads use the language-model embedding space.

**Depth Encoder Pretraining.** The depth encoder is pretrained for five epochs using the frozen RGB teacher representation. We use AdamW with a learning rate of $1 \times 10^{-4}$, a weight decay of $0.01$, and a batch size of 32. The gradient norm is clipped to 1.0. The alignment-loss weights are set to

$$\lambda_{\text{mse}} = 1, \qquad \lambda_{\cos} = 1, \qquad \lambda_{\text{global}} = 0.1, \qquad \lambda_{\text{std}} = 0.1. \tag{9}$$

The resulting depth encoder is then used to initialize the depth branch of the full model.

**Joint Fine-tuning.** The complete AlignTrack model is jointly fine-tuned with the navigation and spatial auxiliary objectives. All trainable components, including the Qwen3-0.6B backbone, RGB projector, depth encoder, action decoder, spatial query, and spatial prediction head, are updated during this stage, while the pretrained DINOv3 and SigLIP vision encoders remain frozen.

For the reported experiments, we use an effective global batch size of 64, a learning rate of $2 \times 10^{-5}$, and a weight decay of $0.01$, and jointly fine-tune the model for five epochs. We adopt AdamW as the optimizer, clip the gradient norm to $1.0$, and enable bfloat16 mixed-precision training. The loss weights are set to

$$\beta_{\text{nav}} = 10, \qquad \beta_{\text{spatial}} = 1, \tag{10}$$

with equal weights for the three spatial auxiliary terms:

$$\lambda_b = \lambda_\theta = \lambda_v = 1. \tag{11}$$

All reported models are trained using distributed data parallelism on eight NVIDIA GeForce RTX 5090 GPUs. Under this configuration, joint fine-tuning converges within approximately 12 hours.

**Data and Evaluation Protocol.** We follow the data-processing and evaluation protocol of TrackVLA. RGB features are precomputed using the frozen DINOv3 and SigLIP encoders, while depth observations and target spatial annotations are loaded together with each tracking sample. The model is evaluated in unseen environments using the DT, STT, and AT settings. We report success rate (SR), strict success rate, tracking rate (TR), and collision rate (CR), where higher SR and TR and lower CR indicate better performance.

### 4.2 Quantitative Comparison

We compare AlignTrack with several representative visual tracking and embodied navigation methods under the STT, DT, and AT settings. Following the evaluation protocol of TrackVLA, we report the success rate (SR), tracking rate (TR), and collision rate (CR). Higher SR and TR indicate more reliable target following, while a lower CR indicates safer navigation.

Table 1 shows that AlignTrack improves tracking rate over TrackVLA across STT, DT, and AT with a 0.6B language backbone. The gains are most pronounced on AT, where SR increases from 50.2% to 64.93% and TR increases from 63.7% to 79.68%, corresponding to improvements of 14.73 and 15.98 percentage points. CR decreases from 17.1% to 8.91%, an 8.19-point reduction. AlignTrack thus improves all three reported metrics over TrackVLA in this setting.

The tracking-rate gains extend to STT and DT. On STT, AlignTrack raises TR from 78.6% to 88.14%, with SR at 84.93% compared with TrackVLA's 85.1%. On DT, TR increases from 63.2% to 71.48%. Across the three settings, the TR improvements are 9.54, 8.28, and 15.98 percentage points, respectively.

Table 1: Quantitative comparison on the STT, DT, and AT tracking settings. Each entry reports SR/TR/CR.

| Methods | Backbone | *STT* | *DT* | *AT* |
|---|---|---|---|---|
| | | SR↑/TR↑/CR↓ | SR↑/TR↑/CR↓ | SR↑/TR↑/CR↓ |
| IBVS† | – | 42.9/56.2/3.75 | 10.6/28.4/6.14 | 15.2/39.5/**4.90** |
| PoliFormer† | – | 4.67/15.5/40.1 | 2.62/13.2/44.5 | 3.04/15.4/41.5 |
| EVT | – | 24.4/39.1/42.5 | 3.23/11.2/47.9 | 17.4/21.1/45.6 |
| EVT‡ | – | 32.5/49.9/40.5 | 15.7/35.7/53.3 | 18.3/21.0/44.9 |
| Uni-NaVid | – | 25.7/39.5/41.9 | 11.3/27.4/43.5 | 8.26/28.6/43.7 |
| TrackVLA | 7B | **85.1**/78.6/**1.65** | **57.6**/63.2/**5.80** | 50.2/63.7/17.1 |
| **AlignTrack (Ours)** | **0.6B** | 84.93/**88.14**/4.13 | 53.70/**71.48**/16.30 | **64.93**/**79.68**/8.91 |

Table 2: Ablation results on EVT-Benchmark. Each entry reports SR/TR/CR (%). "w/o Depth" removes the depth branch and its alignment pretraining; "w/o Depth & Spatial Sup." additionally removes the spatial auxiliary predictions and losses. Higher SR and TR and lower CR are better.

| Variant | *STT* | *DT* | *AT* |
|---|---|---|---|
| | SR↑/TR↑/CR↓ | SR↑/TR↑/CR↓ | SR↑/TR↑/CR↓ |
| w/o Depth & Spatial Sup. | 82.17/85.37/8.71 | 41.96/65.81/22.25 | 58.04/76.57/16.88 |
| w/o Depth | 82.97/87.78/6.96 | **56.52**/**73.44**/20.91 | **66.64**/**81.63**/10.01 |
| **AlignTrack (full)** | **84.93**/**88.14**/**4.13** | 53.70/71.48/**16.30** | 64.93/79.68/**8.91** |

These results establish consistent tracking-rate gains with a language backbone approximately one-eleventh the size of TrackVLA's. The simultaneous improvements in SR, TR, and CR on AT provide the strongest evidence of AlignTrack's benchmark performance.

### 4.3 Ablation Analysis

Table 2 compares full AlignTrack with two nested ablations. The "w/o Depth" variant retains target-centric spatial supervision but removes the aligned depth branch. The "w/o Depth & Spatial Sup." variant additionally removes the spatial auxiliary predictions and losses, leaving the Qwen3-0.6B action-prediction pathway.

Spatial supervision improves all three reported metrics across STT, DT, and AT. Compared with "w/o Depth & Spatial Sup.", the "w/o Depth" variant increases DT success rate from 41.96% to 56.52% and tracking rate from 65.81% to 73.44%. On AT, it improves success rate by 8.60 percentage points and tracking rate by 5.06 points, while reducing collision rate from 16.88% to 9.83%. These results show that explicit target-state supervision strengthens embodied tracking across different evaluation settings.

The aligned depth branch provides a complementary improvement in collision avoidance. Relative to "w/o Depth", full AlignTrack reduces collision rate from 6.96% to 4.13% on STT, from 20.91% to 16.30% on DT, and from 9.83% to 8.91% on AT. It also achieves the strongest STT results on all three metrics. Compared with the variant without either component, full AlignTrack improves success rate and tracking rate in every setting and reduces collision rate by 4.58, 5.95, and 7.97 percentage points on STT, DT, and AT, respectively. The two components thus contribute distinct, complementary benefits to the final tracking policy.

## 5 Conclusion

We presented AlignTrack, a compact vision-language-action framework that connects semantics, geometry, and embodied actions through semantic-aligned depth representations and target-centric

spatial supervision. With a 0.6B language backbone, AlignTrack improves tracking rate over the 7B TrackVLA across all three EVT-Benchmark settings. On AT, it simultaneously improves success rate by 14.73 percentage points and tracking rate by 15.98 points while reducing collision rate by 8.19 points. These results highlight explicit semantic–geometric–action alignment as a promising design direction for compact embodied visual tracking.

## AI USE STATEMENT

In this work, we used generative AI tools, including ChatGPT and Codex, to assist with drafting and revising portions of the manuscript, improving writing clarity and organization, searching and summarizing relevant literature, formatting references, and developing, debugging, and refactoring research code. Generative AI tools were also used to assist with the implementation and discussion of components of the proposed method and experimental pipeline.

All AI-assisted text, code, references, and technical content were reviewed and verified by the authors. The authors independently checked cited literature, tested AI-assisted code and experimental implementations for correctness, and verified the reported results against the actual experimental outputs. All research decisions, experimental conclusions, and final claims were made by the authors. The authors take full responsibility for the final content of this work.

## REFERENCES

Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision Workshops*, pp. 850–865, 2016. doi: 10.1007/978-3-319-48881-3_56. 2

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing*, pp. 3464–3468, 2016. doi: 10.1109/ICIP.2016.7533003. 2

Alberto Dionigi, Simone Felicioni, Mirko Leomanni, and Gabriele Costante. D-VAT: End-to-end visual active tracking for micro aerial vehicles. *IEEE Robotics and Automation Letters*, 2024. doi: 10.1109/LRA.2024.3385700. 2

Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, 2018. 2

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, 2026a. 3

Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. Qdepth-vla: Quantized depth prediction as auxiliary supervision for vision-language-action models. In *International Conference on Autonomous Agents and Multiagent Systems*, 2026b. 3

Jiahang Liu, Yunpeng Qi, Jiazhao Zhang, Minghan Li, Shaoan Wang, Kui Wu, Hanjing Ye, Hong Zhang, Zhibo Chen, Fangwei Zhong, Zhizheng Zhang, and He Wang. TrackVLA++: Unleashing reasoning and memory capabilities in vla models for embodied visual tracking. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. 1, 3

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3286–3295. PMLR, 2018. 2

V. Nivashini and G. Rajesh. YOLO-SR and GeoTrack for distress target detection and tracking in UAV maritime search and rescue. *Scientific Reports*, 2026. doi: 10.1038/s41598-026-68146-w. URL https://doi.org/10.1038/s41598-026-68146-w. 1

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action models. In *Robotics: Science and Systems*, 2025. 3

Ruilong Ren, Songsheng Cheng, Yunpeng Zhou, Hanxuan Chen, Xiangyue Wang, Tianle Zeng, Shuai Yuan, Binbo Li, Hanzhong Guo, Ji Pei, Da Zhang, and Kangli Wang. Cosfly-vla: A spatially aware vision-language-action model for uav tracking. *arXiv preprint arXiv:2607.15004*, 2026. 3

Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. Trackvla: Embodied visual tracking in the wild. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4139–4164. PMLR, 2025. 1, 3

Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing*, pp. 3645–3649, 2017. doi: 10.1109/ICIP.2017.8296962. 2

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*, pp. 341–357, 2022. doi: 10.1007/978-3-031-20047-2_20. 2

Hanjing Ye, Jieting Zhao, Yaling Pan, Weinan Chen, Li He, and Hong Zhang. Robot person following under partial occlusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7591–7597. IEEE, 2023. doi: 10.1109/ICRA48891.2023.10160738. 1

Hanjing Ye, Tianle Zeng, Jiazhao Zhang, Shaoan Wang, Zibo Zhang, Weixi Situ, Yuchen Zhou, Yonggen Ling, and Hong Zhang. Refertrack: Referring then tracking for embodied visual tracking. *arXiv preprint arXiv:2607.20061*, 2026. 3

Cherng Liin Yong, Ban Hoe Kwan, Danny Wee-Kiat Ng, and Hong Seng Sim. Human tracking and following using machine vision on a mobile service robot. In *2022 IEEE 10th Conference on Systems, Process & Control (ICSPC)*, pp. 274–279. IEEE, 2022. doi: 10.1109/ICSPC55597.2022.10001803. 1

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. In *IEEE International Conference on Robotics and Automation*, 2026. 3

Hendrik Zender, Patric Jensfelt, and Geert-Jan M. Kruijff. Human- and situation-aware people following. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1131–1136. IEEE, 2007. doi: 10.1109/ROMAN.2007.4415250. 1

Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, 2022. 2

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.013. 1, 3

Qiyao Zhang, Shuhua Zheng, Jianli Sun, Chengxiang Li, Xianke Wu, Zihan Song, Zhiyong Cui, Yisheng Lv, and Yonglin Tian. UAV-Track VLA: Embodied aerial tracking via vision-language-action models. *arXiv preprint arXiv:2604.02241*, 2026. 1, 3

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3d vision-language-action generative world model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 61229–61245. PMLR, 2024. 3

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. AD-VAT: An asymmetric dueling mechanism for learning visual active tracking. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkgYmhR9KX. 2

Fangwei Zhong, Kui Wu, Hai Ci, Churan Wang, and Hao Chen. Empowering embodied visual tracking with visual foundation models and offline rl. In *European Conference on Computer Vision*, pp. 139–155. Springer, 2024. doi: 10.1007/978-3-031-73464-9_9. 2

# A Decoder and Objective Details

**Spatial decoder.** Let $D = 1024$ denote the hidden width of Qwen3-0.6B. The spatial-query state is first processed by a two-layer MLP,

$$\mathbf{z}_t = \mathrm{GELU}(\mathbf{W}_2\,\mathrm{GELU}(\mathbf{W}_1\,\mathrm{LN}(\mathbf{h}_t^{\mathrm{sp}}))), \tag{12}$$

where both hidden layers have width $D/2 = 512$. Three independent linear heads then produce

$$\widehat{\mathbf{b}}_t = \sigma(\mathbf{W}_b\mathbf{z}_t + \mathbf{c}_b), \qquad \widehat{\psi}_t = \pi\tanh(\mathbf{W}_\psi\mathbf{z}_t + \mathbf{c}_\psi), \qquad \widehat{s}_t = \mathbf{W}_v\mathbf{z}_t + \mathbf{c}_v. \tag{13}$$

The box $\widehat{\mathbf{b}}_t = (\widehat{c}_x, \widehat{c}_y, \widehat{w}, \widehat{h})$ uses normalized image coordinates, $\widehat{\psi}_t \in [-\pi, \pi]$ is the bearing of the target position in the robot frame (not the target person's body orientation), and $\widehat{s}_t$ is an unnormalized visibility logit.

**Trajectory decoder and output scaling.** The action-query state is decoded by a three-layer MLP with two hidden layers of width $2D = 2048$ and GELU activations. For the checkpoint used in our main experiments, the output is unbounded:

$$\mathbf{u}_t = f_{\mathrm{act}}\big(\mathrm{LN}(\mathbf{h}_t^{\mathrm{act}})\big) \in \mathbb{R}^{3K}, \qquad \widehat{\mathbf{Q}}_t = \mathrm{Reshape}_{K\times 3}(\mathbf{u}_t) \odot \boldsymbol{\alpha}, \quad \boldsymbol{\alpha} = (2, 2, 1). \tag{14}$$

Thus only the two translational coordinates are scaled; the heading coordinate is left unchanged. The implementation also supports a bounded variant that applies $\tanh$ before the multiplication in Eq. equation 14, but that option was disabled for all numbers reported in the main paper.

**Depth-to-RGB alignment.** Write $\mathbf{E}^D = [\mathbf{e}_1^D; \ldots; \mathbf{e}_{N_d}^D]$ and $\mathbf{E}^R = [\mathbf{e}_1^R; \ldots; \mathbf{e}_{N_d}^R]$, where $N_d = 64$ and each token has dimension $D$. The minibatch-averaged alignment loss is a weighted sum of

$$\mathcal{L}_{\mathrm{mse}} = \frac{1}{N_d D}\|\mathbf{E}^D - \mathbf{E}^R\|_F^2, \tag{15}$$

$$\mathcal{L}_{\cos} = 1 - \frac{1}{N_d}\sum_{j=1}^{N_d}\frac{(\mathbf{e}_j^D)^\top\mathbf{e}_j^R}{\|\mathbf{e}_j^D\|_2\|\mathbf{e}_j^R\|_2 + \epsilon}, \tag{16}$$

$$\mathcal{L}_{\mathrm{global}} = 1 - \frac{(\overline{\mathbf{e}}^D)^\top\overline{\mathbf{e}}^R}{\|\overline{\mathbf{e}}^D\|_2\|\overline{\mathbf{e}}^R\|_2 + \epsilon}, \tag{17}$$

$$\mathcal{L}_{\mathrm{std}} = \frac{1}{D}\left\|\mathrm{Std}_j(\mathbf{E}^D) - \mathrm{Std}_j(\mathbf{E}^R)\right\|_2^2. \tag{18}$$

Here $\overline{\mathbf{e}} = N_d^{-1}\sum_j \mathbf{e}_j$. The channel-wise standard deviation uses the population estimate (i.e., no Bessel correction). All normalizations use $\epsilon = 10^{-6}$. The final objective is $\mathcal{L}_{\mathrm{align}} = \mathcal{L}_{\mathrm{mse}} + \mathcal{L}_{\cos} + 0.1\mathcal{L}_{\mathrm{global}} + 0.1\mathcal{L}_{\mathrm{std}}$.

**Navigation and target-state losses.** Let $m_{i,k} \in \{0, 1\}$ indicate whether waypoint $k$ of sample $i$ is valid, and define $\widetilde{\mathbf{q}} = \mathbf{q} \oslash \boldsymbol{\alpha}$. The implementation averages the squared error over every valid coordinate:

$$\mathcal{L}_{\mathrm{nav}} = \frac{\sum_{i=1}^{B}\sum_{k=1}^{K} m_{i,k}\,\|\widehat{\mathbf{q}}_{i,k} \oslash \boldsymbol{\alpha} - \mathbf{q}_{i,k} \oslash \boldsymbol{\alpha}\|_2^2}{3\sum_{i=1}^{B}\sum_{k=1}^{K} m_{i,k} + \epsilon}. \tag{19}$$

For spatial supervision, $v_i \in \{0, 1\}$ is the target visibility label:

$$\mathcal{L}_{\mathrm{bbox}} = \frac{\sum_i v_i\ \mathrm{SmoothL1}(\widehat{\mathbf{b}}_i, \mathbf{b}_i)}{\sum_i v_i + \epsilon}, \tag{20}$$

$$\delta_i = \mathrm{atan2}\Big(\sin(\widehat{\psi}_i - \psi_i), \cos(\widehat{\psi}_i - \psi_i)\Big), \qquad \mathcal{L}_{\mathrm{yaw}} = \frac{1}{B}\sum_i \delta_i^2, \tag{21}$$

$$\mathcal{L}_{\mathrm{vis}} = \frac{1}{B}\sum_i \mathrm{BCEWithLogits}(\widehat{s}_i, v_i). \tag{22}$$

SmoothL1 is averaged over the four box coordinates. Bounding-box regression is masked for invisible targets, whereas yaw remains supervised because the simulator provides the target-relative bearing even during occlusion. Joint training uses

$$\mathcal{L}_{\mathrm{joint}} = 10\mathcal{L}_{\mathrm{nav}} + \mathcal{L}_{\mathrm{bbox}} + \mathcal{L}_{\mathrm{yaw}} + \mathcal{L}_{\mathrm{vis}}. \tag{23}$$

The feature-alignment objective is not retained during joint training.

Table 3: Depth-token encoder. Shapes exclude the batch dimension.

| Stage | Kernel/stride/pad | Channels | Output shape | Normalization |
|---|---|---|---|---|
| Input | – | 1 | $1 \times 384 \times 384$ | – |
| Conv + GELU | $7/4/3$ | 64 | $64 \times 96 \times 96$ | GN(8) |
| Conv + GELU | $5/4/2$ | 128 | $128 \times 24 \times 24$ | GN(16) |
| Conv + GELU | $3/3/1$ | 256 | $256 \times 8 \times 8$ | GN(1) |
| Adaptive average pool | – | 256 | $256 \times 8 \times 8$ | – |
| Flatten | – | 256 | $64 \times 256$ | – |
| Depth projector | – | 1024 | $64 \times 1024$ | LayerNorm |

# B Detailed Architecture and Token Flow

## B.1 Vision preprocessing and projectors

Both RGB towers receive bicubically resized $384 \times 384$ images. We discard the DINOv3 class and register tokens, spatially resize the SigLIP token grid to the DINOv3 grid, and concatenate the corresponding 384-dimensional DINOv3 and 1152-dimensional SigLIP patch features. Adaptive average pooling produces a $2 \times 2$ grid for every preceding frame and an $8 \times 8$ grid for the current frame. The shared RGB projector is

$$P_R(\mathbf{x}) = \mathbf{W}_2 \, \mathrm{GELU}(\mathbf{W}_1 \, \mathrm{LN}(\mathbf{x}) + \mathbf{b}_1) + \mathbf{b}_2, \tag{24}$$

with dimensions $1536 \to 1024 \to 1024$.

Metric depth is sanitized before encoding: NaN and positive infinity are mapped to the 10-m far plane, negative infinity is mapped to zero, and all values are clipped to $[0, 10]$ m and divided by 10. Table 3 specifies the complete depth branch. Group normalization makes this branch independent of the local minibatch size. Its modality-specific projector has the same LayerNorm–Linear–GELU–Linear form as $P_R$, but maps $256 \to 1024 \to 1024$; the RGB projector is never reused on depth tokens.

## B.2 Temporal tokens and exact sequence layout

We use 31 preceding RGB frames in addition to the current RGB observation. Each preceding frame contributes four pooled features and one learned temporal token; the current frame contributes 64 pooled features and one temporal token. For time index $r$ and modality kind $k \in \{\mathrm{hist}, \mathrm{cur}\}$, the temporal token is

$$\mathbf{e}^{\mathrm{time}}_{r,k} = \mathbf{e}^{\mathrm{index}}_{r} + \mathbf{e}^{\mathrm{view}}_{0} + \mathbf{e}^{\mathrm{kind}}_{k}. \tag{25}$$

The optional camera-angle embedding is disabled because all experiments use one egocentric camera. If fewer than 31 preceding frames are available at the beginning of an episode, the sequence is left-padded by repeating its earliest available frame. At the first observation, the current coarse feature is used as this padding value, matching training and closed-loop evaluation.

Let $\mathcal{H}_t$ be the 31 interleaved history blocks, $\mathbf{E}^D_t$ the 64 depth tokens, and $\mathcal{C}_t$ the interleaved current-frame block. The exact backbone input order is

$$\mathbf{S}_t = [\mathrm{Emb}(I); \mathcal{H}_t; \mathbf{E}^D_t; \mathcal{C}_t; \mathbf{p}_{\mathrm{sp}}; \mathbf{p}_{\mathrm{act}}]. \tag{26}$$

The instruction is truncated to at most 128 tokens. Excluding instruction tokens, Eq. equation 26 contains $31(4+1) + 64 + (64+1) + 2 = 286$ tokens, giving a maximum sequence length of 414.

## B.3 Structured causal attention

The backbone uses causal attention, with two additional restrictions designed to preserve modality roles. First, a depth-token query cannot attend to the instruction or historical RGB blocks. It can attend only to preceding depth tokens, so the depth representation remains independent of language and visual history before fusion. The subsequent current-RGB block can attend to the instruction, history, and all depth tokens. Second, the spatial-query row can attend only to valid instruction tokens, current depth, current RGB, and itself; historical RGB and the following action query are

masked. Since the action query is last, it can attend to every preceding non-padding token, including the spatial-query representation. This ordering implements the intended flow *current target grounding → action prediction* while retaining the temporal context needed by the action query.

## C Label Construction and Closed-Loop Control

**Target-centric labels.** The visibility label is the binary simulator annotation at the current frame. The normalized box is stored as $(c_x, c_y, w, h)$. The relative-yaw label is the bearing of the target position in the robot coordinate frame; it should not be interpreted as the target humanoid's body heading.

**Future trajectory labels.** Trajectory targets are computed from recorded robot poses rather than by integrating commanded velocities. Let $(X_t, Z_t)$ and $\phi_t$ be the current world-plane position and yaw. For a future pose $(X_{t+k}, Z_{t+k}, \phi_{t+k})$, we use

$$x_{t,k} = \Delta X_k \cos\phi_t + \Delta Z_k \sin\phi_t, \tag{27}$$

$$y_{t,k} = \Delta X_k \sin\phi_t - \Delta Z_k \cos\phi_t, \tag{28}$$

$$\theta_{t,k} = \mathrm{wrap}(\phi_t - \phi_{t+k}), \tag{29}$$

where $\Delta X_k = X_{t+k} - X_t$ and $\Delta Z_k = Z_{t+k} - Z_t$. Data generation requires sufficient future poses, constructs nine future points, and retains the first $K = 8$ points during loading. The loader additionally supports a per-waypoint validity mask; absent an explicit mask, all eight points are valid.

**Receding-horizon execution.** Although the policy predicts eight waypoints in parallel, closed-loop evaluation executes only the first waypoint and replans after the next observation. If $\widehat{\mathbf{q}}_{t,1} = (\widehat{x}, \widehat{y}, \widehat{\theta})$, the normalized Habitat base-velocity command is

$$\mathbf{a}_t = \mathrm{clip}_{[-1,1]}\left(\left[\frac{\widehat{x}}{v_x/f_c}, \frac{\widehat{y}}{v_y/f_c}, \frac{\widehat{\theta}}{\omega/f_c}\right]\right), \tag{30}$$

where $(v_x, v_y, \omega) = (15, 10, 6.28)$ and the simulator control frequency is $f_c = 40$ Hz. Hence one unit normalized action corresponds to maximum local changes of $(0.375\,\mathrm{m}, 0.25\,\mathrm{m}, 0.157\,\mathrm{rad})$ per control step. The current coarse RGB feature is appended to the history only after the action is computed, preventing the current image from appearing in both the history and current-frame blocks at the same decision step.

## D Reproducibility Details

**Feature-alignment stage.** The alignment teacher consists of cached frozen DINOv3–SigLIP current-frame features followed by a frozen RGB projector copied from a five-epoch RGB–spatial policy checkpoint. Only the 1.82M-parameter depth branch is optimized. Training uses 2,241,154 paired observations for five epochs on eight GPUs, with a per-GPU batch size of 32 (global batch size 256), AdamW, learning rate $10^{-4}$, weight decay $0.01$, gradient-norm clipping at 1.0, and bfloat16 automatic mixed precision. This corresponds to 8,755 optimizer steps per epoch and 43,775 steps in total. The random seed is zero.

**Joint stage.** The joint dataset contains 1,986,693 samples: 674,124 AT, 712,430 DT, and 600,139 STT samples. We train for five epochs on eight GPUs using a per-GPU batch size of 8 (global batch size 64), AdamW with a constant learning rate of $2 \times 10^{-5}$, weight decay $0.01$, gradient-norm clipping at 1.0, and bfloat16 automatic mixed precision. No warmup, learning-rate scheduler, or gradient accumulation is used. The five epochs contain 155,215 optimizer steps. DINOv3 and SigLIP remain frozen and their half-precision features are loaded from the offline cache. Qwen3-0.6B, both modality projectors, the depth encoder, temporal embeddings, both query tokens, and both decoders are updated. The unused angle-projection parameters are frozen.

Table 4: Policy parameter accounting. Frozen DINOv3 and SigLIP vision towers are excluded because the table reports the AlignTrack policy checkpoint.

| Component | Parameters |
|---|---|
| Qwen3-0.6B backbone | 596.050M |
| RGB projector | 2.627M |
| Depth encoder and projector | 1.817M |
| Temporal/view/kind/angle module | 4.200M |
| Trajectory decoder | 6.347M |
| Spatial decoder | 0.793M |
| Spatial and action query tokens | 0.002M |
| Total policy parameters | 611.836M |
| Trainable in the reported setting | 611.833M |

**Initialization.** The depth encoder is initialized from the final alignment artifact. The Qwen backbone is initialized from Qwen3-0.6B. The reported joint run starts a new optimizer and does not resume a complete policy checkpoint. This distinction is important: the RGB projector used to define the alignment teacher comes from the RGB–spatial checkpoint, whereas the RGB projector in the joint run is newly initialized and then learned jointly with all other policy components.

## E Ablation Protocols

To isolate the two proposed components without introducing unrelated topology changes, we use the following controls. *Without spatial supervision* removes $\mathcal{L}_{\mathrm{bbox}}$, $\mathcal{L}_{\mathrm{yaw}}$, and $\mathcal{L}_{\mathrm{vis}}$ and freezes the unused spatial prediction MLP, but retains the spatial query and the same attention topology. Consequently, this control measures the effect of explicit target-state supervision rather than changing the action network's sequence length. *Without aligned depth initialization* retains the complete depth branch but initializes it randomly before joint training. *Without depth* removes the depth block while retaining the RGB pathway, spatial query, and target-state losses. All controls use the same data, five-epoch schedule, global batch size, and evaluation protocol as the full model.