E3AD: An Emotion-Aware Vision-Language-Action Model for Human-Centric End-to-End Autonomous Driving

Supplementary Material

A. Data Construction

A.1. VAD Label Construction

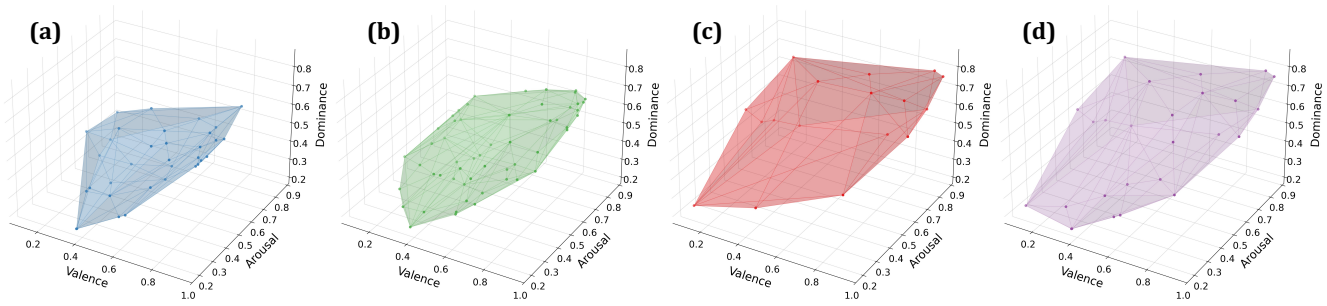


Figure A1. Visualization of the VAD space coverage using 3D convex hulls. **(a)** The hull of the Original commands (Blue) is concentrated in the neutral region, showing limited emotional diversity. **(b)** The Augmented commands (Green) significantly expand the volume, introducing more varied emotional states. **(c)** The hull of the discrete GoEmotions labels (Red) outlines the semantic boundaries of standard emotion categories. **(d)** The final Union (Purple) of our dataset covers a comprehensive volume of the affective space, ensuring the model is trained on a continuous spectrum of Valence, Arousal, and Dominance.

To capture the subtle emotional nuances in natural language commands, we construct continuous Valence-Arousal-Dominance (VAD) labels using a hybrid fusion strategy that combines sentence-level sentiment analysis with word-level lexical grounding. As implemented in our data processing pipeline, the final VAD vector e is computed as a weighted average of two sources:

Sentence-Level Inference (GoEmotions) We utilize a RoBERTa-based classifier fine-tuned on the GoEmotions dataset (`SamLowe/roberta-base-go_emotions`)[1] to predict probability scores for 28 discrete emotion categories (e.g., joy, anxiety, neutrality). We map these discrete labels to the continuous VAD space using the NRC-VAD Lexicon [6]. For a given command, the sentence-level VAD vector e_{goe} is calculated as the weighted average of the VAD values of the predicted labels, weighted by their confidence scores.

Word-Level Inference (TF-IDF + Heuristics) To capture keyword-specific emotional intensity, we tokenize the command and filter stop words. We compute a word-level VAD vector e_{words} by averaging the NRC-VAD[2] values of the remaining tokens, weighted by their TF-IDF scores calculated over the entire corpus. Additionally, we apply a heuristic "Exclamation Boost" to the Arousal dimension. If the command contains exclamation marks, the Arousal score is incrementally increased to reflect heightened urgency.

Fusion and Coverage The final label is obtained via linear fusion: $e_{\text{final}} = \alpha \times e_{\text{goe}} + (1 - \alpha) \times e_{\text{words}}$, with $\alpha = 0.5$. To validate the effectiveness of our emotion-aware data augmentation, we visualize the convex hulls of the resulting VAD vectors in Figure A1. As shown in **(a)**, the original Talk2Car commands (Blue) occupy a narrow, mostly neutral region of the affective space. **(b)** Our augmentation strategy (Green) significantly expands this coverage. **(c)** The GoEmotions prior (Red) represents the theoretical space spanned by discrete emotion categories. **(d)** The final union of data (Purple) demonstrates that our approach successfully covers a broad, continuous spectrum of the VAD space, enabling the model to learn diverse emotional representations.

[1] https://huggingface.co/SamLowe/roberta-base-go_emotions
[2] https://saifmohammad.com/WebPages/nrc-vad.html

A.2. Spatial Reasoning Data Construction

To support the multi-task instruction tuning of E3AD, we processed the raw annotations from the Talk2Car-Trajectory dataset into standardized supervision signals. In the following, we specifically detail the data construction strategies for tasks that necessitated specialized algorithmic generation or heuristic filtering, as opposed to direct annotation usage.

Coordinate System Standardization To ensure consistency across egocentric and allocentric reasoning tasks, we transformed all spatial annotations into a unified Ego-Centric Cartesian Coordinate System. The ego vehicle's centroid is defined as the origin $(0, 0)$. Following standard autonomous driving conventions, the positive X-axis points to the vehicle's forward direction, and the Y-axis represents the lateral dimension. We converted the raw pixel coordinates from the Bird's-Eye View (BEV) maps into metric coordinates using the dataset's resolution (10 pixels/meter). All target locations and trajectory waypoints were transformed into this local frame relative to the ego vehicle's position at the current timestamp.

Egocentric Spatial Grounding We utilized the 2D bounding box annotations provided by the dataset for the referred object in the frontal camera view. The bounding box coordinates $(x_{min}, y_{min}, x_{max}, y_{max})$ were normalized relative to the image dimensions to serve as the regression target for the visual grounding token <ego>.

Egocentric Spatial Relation Since the dataset does not provide explicit categorical spatial labels, we algorithmically generated these labels to train the model's spatial reasoning. To acquire robust 3D positional information from the 2D frontal images, we utilized the standard 3D object detection results (FCOS3D [7]) provided by the Talk2Car benchmark. We first filtered these detections based on a high confidence threshold to ensure reliability. Furthermore, to eliminate referential ambiguity, where multiple objects of the same class (e.g., multiple "cars") co-exist, making the spatial relation ill-defined without complex descriptors, we excluded samples containing duplicate object classes. For the remaining unambiguous targets, we computed the geometric angle θ of the target's centroid relative to the ego vehicle in the metric allocentric space and discretized it into five distinct classes (Directly Ahead, Front-Left, Front-Right, Left, Right).

Egocentric Depth Estimation To supervise 3D perception recovery, we employed the same detection-based data construction strategy. Using the high-confidence (> 0.3), non-ambiguous 3D detections described above, we computed the ground truth depth as the Euclidean distance between the ego vehicle's centroid and the target object's predicted 3D centroid. This scalar value serves as the regression target, ensuring the model learns to extract depth cues for specific, unambiguous objects from the 2D frontal view.

Allocentric Target Location Estimation For the allocentric localization task, we generated specialized visual inputs by rendering the geometric annotations onto the raw BEV semantic maps. Specifically, we superimposed the polygons of the ego vehicle (colored red) and the referred target object (colored yellow) onto the map image using their ground-truth annotations. This visual augmentation provides explicit cues for the model to learn the mapping between visual map features and metric locations. The supervision target is defined as the metric coordinates (x, y) of the target object's centroid relative to the ego vehicle, transformed into the unified coordinate system described above.

B. Experimental Setups

B.1. Datasets

We evaluate E3AD across four challenging real-world benchmarks: Talk2Car [2], DrivePilot [5], MoCAD [4], and Talk2Car-Trajectory [3].

Talk2Car. Talk2Car is built on top of nuScenes [1] and provides 11,959 natural-language commands over 9,217 urban driving images captured in Singapore and Boston. Commands average 11 words and frequently express multi-step, relational reasoning. A linguistic breakdown shows rich syntactic variability (avg. 2.32 nouns, 2.29 verbs, 0.62 adjectives per command), and each video segment is paired with $\sim$14 commands, providing strong multimodal grounding supervision. We use the official split: 8,349 training, 1,163 validation, and 2,447 testing commands.

DrivePilot. DrivePilot is a new dataset that supports open-domain grounding in complex scenes. It leverages Qwen2-VL under regularized prompts to generate structured semantic annotations describing weather, scene context, traffic participants, emotional cues, and interaction patterns. Each sample contains a natural-language instruction, front-view and BEV images,

LLM-generated descriptions, and precise target-object localization. DrivePilot is designed to challenge fine-grained referent disambiguation and command comprehension in realistic, congested urban scenes.

MoCAD. MoCAD originates from Macau's Level-4 autonomous bus deployment and covers over 300 hours of real-world driving. It includes data from a 5 km campus route, an extended 25 km urban corridor, and dense traffic scenarios under varying weather and lighting conditions. The dataset contains ∼13k images and ∼40k annotated objects with commands averaging 12.5 words. Its right-hand–driving configuration differs from many datasets, making MoCAD a valuable benchmark for domain shift, layout adaptation, and cross-cultural grounding robustness.

Talk2Car-Trajectory. Talk2Car-Trajectory augments Talk2Car with multiple human-annotated feasible trajectories for each command. Following the cropping protocol in [3], samples whose referred object falls outside the 120×80 m BEV region are removed. The final Talk2Car-Trajectory splits contain 8,301 training commands (mean path length 28.37 m, with 99.55% of samples annotated with three trajectories), 1,149 validation commands (mean path length 27.98 m, 99.31% with three trajectories), and 2,439 test commands (mean path length 28.41 m, 98.81% with three trajectories). Each command is paired with multiple feasible human-annotated paths, enabling evaluation under diverse ground-truth futures. We adopt this dataset to evaluate end-to-end grounding and navigation quality under multiple ground-truth futures.

B.2. Evaluation Metrics

This section provides the formal definitions of the metrics used in OD-E2E AD, including visual grounding, spatial reasoning, and trajectory planning. Specifically:

Intersection-over-Union (IoU). For the visual grounding evaluation, we evaluate grounding using Intersection-over-Union (IoU). Given the predicted bounding box $\hat{b}$ and the ground-truth box b, the IoU can be formally defined as:

$$\mathrm{IoU}_{50}(\hat{b}, b) = \frac{\mathrm{Area}\left(\hat{b} \cap b\right)}{\mathrm{Area}\left(\hat{b} \cup b\right)}. \tag{1}$$

A prediction is considered correct if $\mathrm{IoU} > 0.5$, following the ECCV Commands 4 Autonomous Vehicles protocol.

For spatial reasoning, we report Mean Absolute Error (MAE) and IoU for spatial reasoning tasks. Formally,

Target Location MAE. Let $p \in \mathbb{R}^2$ denote the ground-truth target location and $\hat{p}$ denotes the estimated target location, and the target location MAE is defined as follows:

$$\mathrm{MAE}_{\text{loc}} = \|\hat{p} - p\|_2. \tag{2}$$

Depth Estimation MAE. Given predicted depth $\hat{d}$ and ground truth d, the depth estimation MAE are defined as:

$$\mathrm{MAE}_{\text{depth}} = |\hat{d} - d|. \tag{3}$$

Accuracy. The accuracy within g meters (PA_g) is defined:

$$\mathrm{PA}_g = \mathbb{I}\left(\|\hat{p} - p\|_2 \leq g\right). \tag{4}$$

For trajectory planning, let the predicted trajectory be $\hat{\tau} = \{\hat{p}_1, \ldots, \hat{p}_{N_p}\}$ and the ground-truth trajectory be $\tau = \{p_1, \ldots, p_{N_p}\}$, the planning metrics are defined as follows:

Average Displacement Error (ADE).

$$\mathrm{ADE}(\hat{\tau}, \tau) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\hat{p}_i - p_i\|_2 . \tag{5}$$

Final Displacement Error (FDE).

$$\mathrm{FDE}(\hat{\tau}, \tau) = \left\|\hat{p}_{N_p} - p_{N_p}\right\|_2 . \tag{6}$$

Discrete Fréchet Distance. Let $\hat{\tau} = [\hat{p}_1, \ldots, \hat{p}_{N_p}]$ and $\tau = [p_1, \ldots, p_{N_p}]$, the discrete Fréchet distance is defined:

$$\mathrm{Fréchet}(\hat{\tau}, \tau) = \min_{\alpha,\beta} \max_{t} \left\|\hat{p}_{\alpha(t)} - p_{\beta(t)}\right\|_2 , \tag{7}$$

where α, β are monotone reparameterizations.

Dynamic Time Warping (DTW). The DTW is defined as:

$$\mathrm{DTW}(\hat{\tau}, \tau) = \min_{\pi} \sum_{(i,j)\in\pi} \|\hat{p}_i - p_j\|_2 , \tag{8}$$

where π is a warping path preserving temporal order.

Symmetric Segment-Path Distance (SSPD). For path-to-segment distance $\mathrm{SPD}(a, b)$:

$$\mathrm{SSPD}(\hat{\tau}, \tau) = \frac{1}{2} \left[\mathrm{SPD}(\hat{\tau}, \tau) + \mathrm{SPD}(\tau, \hat{\tau})\right] . \tag{9}$$

B.3. Decoder Setups

For the decoder, we select the best-performing baseline, PTPC, as the trajectory generator. We first replace the native object detection module of PTPC with the visual grounding results generated by our model, utilizing our predicted bounding box as the definitive target prior. Furthermore, we incorporate our predicted coarse trajectory as a spatial guidance signal; specifically, we encode the coarse waypoints into an additional layout tensor layer, which is concatenated with the original environmental layout features during training. This integration provides a robust spatial prior that constrains the search space for fine-grained physical refinement. Notably, this output format, comprising explicit 2D waypoints and grounded coordinates, is geometric and model-agnostic, ensuring that our framework is not limited to PTPC but can be seamlessly adapted to various trajectory decoders that accept spatial priors or goal-conditioned inputs.

C. Supplementary Results

C.1. Qualitative Results

We provide additional qualitative results in Figure A2 to further demonstrate the robustness of E3AD across diverse driving scenarios, ranging from lane changing and turning to parking and car following. In each case, the model successfully grounds the target object referenced in the command and generates a feasible trajectory. Crucially, the visualized `<EmoThink>` and `<Feedback>` outputs illustrate the model's ability to interpret fine-grained emotional cues, such as urgency or caution, from the natural language command and generate human-centric responses that are tonally aligned with the passenger's state.

C.2. Effects of Emotion-Action Alignments

Training Dynamics and Convergence. Before evaluating the generated trajectories, we first verify the optimization stability of the DPO process. As illustrated in Figure A3, the training exhibits two critical trends. First, the NLL Loss (**red curve**) rapidly decreases and stabilizes, indicating that the policy is successfully modeling the probability distribution of the ground-truth trajectories without mode collapse. Second, and crucially, the Reward Margin (**blue curve**), defined as the log-probability gap $\log p_\theta(\tau^{(i)}|C^{(i)}) - \log p_\theta(\widetilde{\tau}_{k^-}^{(i)}|C^{(i)})$, shows a consistent upward trend. This increasing margin confirms that the model is effectively learning to distinguish between the optimal ground-truth $\tau^{(i)}$ and the emotion-shifted negative samples $\widetilde{\tau}_{k^-}^{(i)}$ (constructed in Eq. 4), assigning higher likelihoods to the alignment-consistent trajectories. The simultaneous convergence of both metrics validates that the optimization objective in Eq. 5 has been effectively minimized.

To quantitatively evaluate whether the generated trajectories reflect the intended emotional semantics and physical realism, we analyze the geometric properties of the trajectories. Let a generated trajectory be denoted as $\hat{\tau} = \{\hat{p}_1, \ldots, \hat{p}_{N_p}\}$, where $\hat{p}_i \in \mathbb{R}^2$ represents the spatial coordinates at step i, and N_p is the horizon length.

C.2.1. Metric Definitions

We adopt four geometric metrics to characterize the motion patterns. While *Straightness* measures global efficiency, we introduce a deviation-based *Sinuosity* to explicitly capture local instability (jitter), which is critical for evaluating trajectory generation.

Straightness: Defined as the ratio of the Euclidean distance between the start and end points to the total path length. It indicates global transport efficiency:

$$\mathcal{M}_{\text{str}} = \frac{\|\hat{p}_{N_p} - \hat{p}_1\|_2}{\sum_{i=1}^{N_p-1} \|\hat{p}_{i+1} - \hat{p}_i\|_2} \tag{10}$$

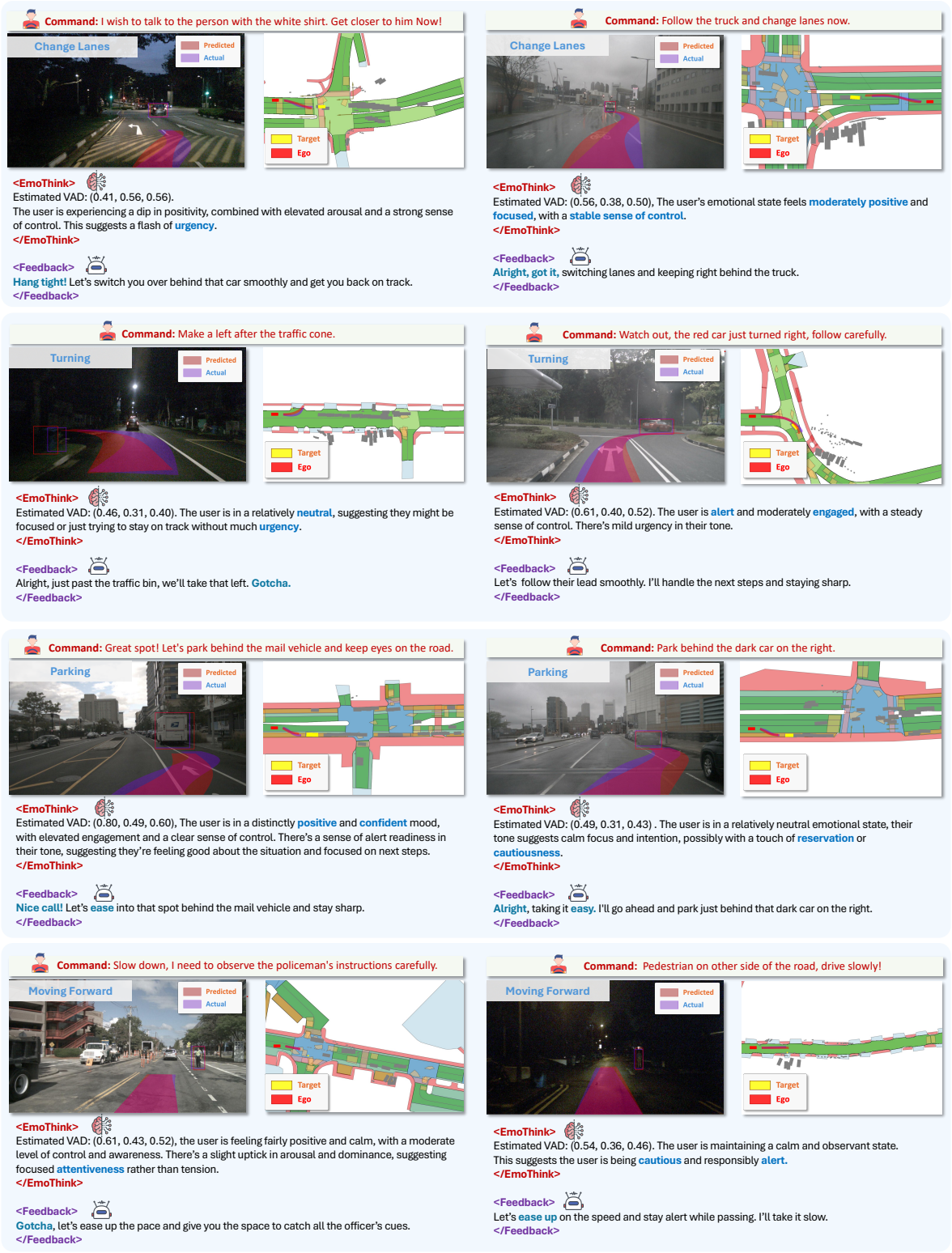


Figure A2. Additional qualitative examples of E3AD across different driving maneuvers (Change Lanes, Turning, Parking, Moving Forward). For each sample, we display the input command, the egocentric view with visual grounding, and the allocentric map with the predicted trajectory. The outputs include the estimated VAD scores, the chain-of-thought emotional analysis (<EmoThink>), and the generated verbal response (<Feedback>), highlighting how the system adapts its interaction style to the detected emotional intent (e.g., responding with "Hang tight!" for urgent commands versus "Taking it easy" for cautious ones).

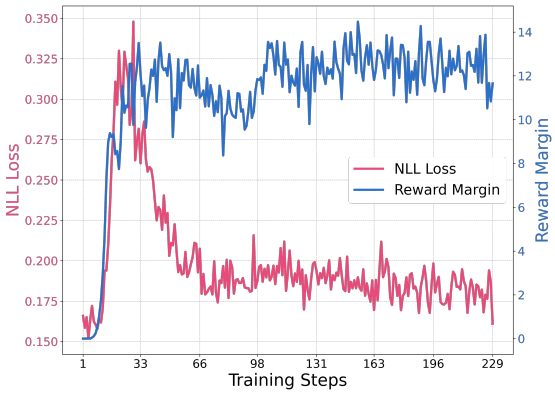


Figure A3. **DPO Training Dynamics.** The **red curve** denotes the **NLL Loss**, reflecting policy optimization, while the **blue curve** tracks the **Reward Margin** (the log-probability gap between preferred and rejected trajectories). The decrease in loss and increase in margin demonstrate successful convergence and preference learning.

Table A1. **Quantitative Evaluation of DPO Efficacy. Panel A** assesses *emotion alignment*, showing that DPO recovers the correct arousal-response patterns (physical laws) observed in Ground Truth. **Panel B** evaluates *physical structural alignment*, demonstrating that DPO significantly enhances the geometric fidelity of the generated trajectories relative to Ground Truth.

Metric	**Straightness**	**Mean Turn**	**Angle Var.**	**Sinuosity**
Panel A: Arousal Alignment Error *(Target: Match GT)*				
GT Correlation (Ref)	0.161	−0.163	−0.155	−0.158
w/o DPO	−0.029	0.023	0.030	0.033
Δ Abs. Error ↓	*0.190*	*0.186*	*0.185*	*0.191*
w/ DPO	**0.167**	**−0.183**	**−0.158**	**−0.166**
Δ Abs. Error ↓	*0.006*	*0.020*	*0.003*	*0.008*
Error Reduction ↑	**96.8%**	**89.2%**	**98.4%**	**95.8%**
Panel B: Physical Alignment Quality *(Target: High Correlation with GT)*				
w/o DPO	0.579	0.476	0.462	0.576
w/ DPO	**0.633**	**0.519**	**0.490**	**0.629**
Improvement ↑	**+9.3%**	**+9.0%**	**+6.1%**	**+9.2%**

Mean Turn: The average absolute change in heading angle, reflecting the tortuosity of the path. Let θ_i be the heading angle at step i, then:

$$\mathcal{M}_{\text{turn}} = \frac{1}{N_p - 2} \sum_{i=1}^{N_p-2} |\theta_{i+1} - \theta_i| \tag{11}$$

Angle Variance: Measures the smoothness and consistency of directional changes. Let δ_i be the turning angle at step i, defined as the angular difference between consecutive heading vectors (adjusted to $(-\pi, \pi]$). The variance is computed as:

$$\mathcal{M}_{\text{var}} = \frac{1}{N_p - 2} \sum_{i=1}^{N_p-2} (\delta_i - \bar{\delta})^2 \tag{12}$$

where $\bar{\delta}$ is the mean turning angle. High variance indicates a trajectory with erratic or jerky steering patterns.

Sinuosity (Lateral Deviation): Unlike the standard length-based definition (which is the reciprocal of straightness), we define Sinuosity as the **Mean Lateral Deviation** to capture path oscillation. It is calculated as the average perpendicular distance of all points from the ideal straight line connecting the start ($\hat{p}_1$) and end ($\hat{p}_{N_p}$):

$$\mathcal{M}_{\text{sin}} = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{|(\hat{p}_{N_p} - \hat{p}_1) \times (\hat{p}_1 - \hat{p}_i)|}{\|\hat{p}_{N_p} - \hat{p}_1\|_2} \tag{13}$$

where $\times$ denotes the 2D cross-product (determinant). This metric is sensitive to the high-frequency jitter often observed in unaligned generative models.

C.2.2. Quantitative Analysis

We evaluate the efficacy of Direct Preference Optimization (DPO) from two perspectives: *Semantic Alignment* (consistency with emotional laws) and *Physical Alignment* (geometric fidelity to ground truth). The results are summarized in Table A1 (Panels A and B).

Emotion Alignment (Panel A). We compute the Spearman rank correlation (ρ) between the input *Arousal* value and the extracted trajectory features. In the Ground Truth (GT) dataset, Arousal is negatively correlated with turning behaviors (e.g., $\rho_{\text{turn}} = -0.163$), implying that high-arousal movement tends to be more direct. As shown in Panel A, the baseline model (w/o DPO) exhibits a *sign error*, incorrectly predicting a positive correlation ($\rho = 0.023$) and failing to capture the underlying physical law. In contrast, the DPO-aligned model successfully recovers the correct negative correlation ($\rho = -0.183$), reducing the alignment error by 89.2% to 98.4% across all metrics. This confirms that DPO effectively injects the correct emotion-action constraints into the policy.

Physical Structural Alignment (Panel B). We further calculate the Spearman correlation between the geometric features of the generated trajectories $\hat{\tau}$ and the paired ground truth τ. Higher correlation indicates better preservation of the realistic geometric structure. As shown in Panel B, DPO consistently improves the structural fidelity. Notably, for the *Sinuosity* metric defined in Eq. (12), DPO achieves a 9.2% improvement ($0.576 \rightarrow 0.629$). This significant gain suggests that the DPO-trained model effectively suppresses unrealistic trajectory jitter and oscillation, resulting in smoother and more human-like motion paths compared to the baseline.

D. Prompts Design

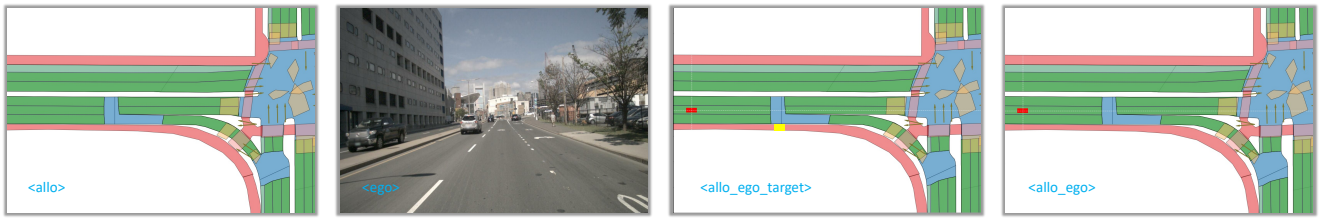
Modality Pretraining During the initial pretraining stage, we utilize discrete, task-specific prompts to equip the model with foundational spatial and emotional capabilities. As illustrated in the left and top-right panels of Figure A4, these include:

- **Egocentric Spatial Grounding:** The model is prompted with the frontal view token `<ego>` to output the 2D bounding box of the target. This trains the model's ability to visually align linguistic semantics with the immediate first-person perceptual field, which is essential for short-horizon interaction.
- **Egocentric Spatial Relation:** To mimic human-like spatial cognition, this prompt asks the model to classify the target's relative position (e.g., "Front-Left") rather than just regressing coordinates. This facilitates a more robust understanding of directionality and relative bearings in the ego-coordinate system.
- **Egocentric Depth Estimation:** This task prompts the model to regress the Euclidean distance to the target. It explicitly supervises the vision encoder to recover 3D metric depth from 2D frontal images, a critical capability for maintaining safety margins and longitudinal control.
- **Allocentric Target Location Estimation:** Using the allocentric token `<allo_ego_target>`, the model estimates target coordinates in the map frame. This task aligns the visual representation with a world-centered "cognitive map," enabling the agent to resolve occlusions and understand global scene topology.
- **Waypoint Planning:** We employ separate prompts for both Egocentric and Allocentric views to predict the future waypoints. By enforcing the model to derive the same trajectory from distinct visual perspectives, we ensure that the planning policy is grounded in both immediate perceptual cues (Egocentric) and global structural constraints (Allocentric), fostering robust spatial representations that are consistent across views.
- **Emotion Modeling:** This text-only prompt instructs the model to map the command into the continuous Valence, Arousal, and Dominance (VAD) space. This acts as a cognitive prior, allowing the system to distinguish the urgency and tone of the request before planning.

Joint Fine-tuning In the second stage, we employ a unified prompt template to enable Emotion-aware Chain-of-Thought (CoT) reasoning. As shown in the "Joint Fine-tuning" panel of Figure A4, the input sequence concatenates the synchronized visual tokens (`<allo>` and `<ego>`) with the user command. The prompt explicitly instructs the model to perform three tasks in sequence: (1) analyze the VAD values, (2) visually ground the target in the frontal view, and (3) predict the future waypoints. This autoregressive structure ensures that the final trajectory planning is conditioned on both the emotional interpretation and spatial grounding results.

Emotion-Action Alignment To align the model's behavior with human emotional intent, we utilize the Direct Preference Optimization (DPO) prompt template. As depicted in the "Emotion-Action Alignment" panel, this prompt presents the visual context and command, soliciting a trajectory prediction. During training, this is structured as a preference pair: the *Preferred* output contains the ground-truth trajectory corresponding to the original command, while the *Rejected* output contains a trajectory generated from an emotion-augmented "negative" command (e.g., a trajectory that ignores urgency or caution). This guides the model to prefer actions that are consistent with the emotional tone of the instruction.

Inference During the inference phase, E3AD utilizes the same prompt structure as the Joint Fine-tuning stage. The system receives the multi-view image tokens and the natural language command as input. It then generates the full response sequence end-to-end: first estimating the VAD emotion state, then locating the target object, and finally planning the waypoints. This unified prompting strategy allows the model to dynamically adapt its planning strategy based on the inferred emotional urgency and spatial context without requiring separate task triggers.



Egocentric Spatial Grounding <ego>

Command: {command}
Locate the target object in the image that the command refers to.

Answer: {"bbox_2d": [x_min, y_min, x_max, y_max]}

Egocentric Spatial Relation <ego>

You are given a Frontal View image of the ego vehicle. The ego vehicle is centered at (0, 0) and faces the positive X-axis (forward).

Command: {command}

Based on the image and the command, determine the spatial position of the target object relative to the ego vehicle. Select the correct option from the following:

A. Directly Ahead B. Front-Left C. Front-Right D. Left Side E. Right Side

Answer: B

Egocentric Depth Estimation <ego>

You are given a Frontal View image of the ego vehicle.

Command: {command}

What is the approximate straight-line (Euclidean) distance in meters from the ego vehicle to the target object referred to by the command?

Answer: 3.6 meters.

Egocentric Waypoint Planning <ego>

You are given a Frontal View image of the ego vehicle:
The ego vehicle is centered at (0, 0) and faces the positive X-axis (forward).
The positive Y-axis points to the left of the ego vehicle, and the negative Y-axis to the right.

Command: {command}

Based on the Frontal View image, predict {N_WPs} future waypoints (X, Y) in meters that fulfill the command.

Answer: {"waypoints": [[x1, y1], [x2, y2], ...]}

Allocentric Waypoint Planning <allo_ego_target>

You are given a Bird's-Eye View (BEV) image showing the ego vehicle (red) and the target object (yellow):

Bird's-Eye View (BEV): covers 120 m ahead (X-axis) and 80 m laterally (Y-axis).
The ego vehicle is centered at (0, 0) and faces the positive X-axis (forward).
The positive Y-axis points to the left of the ego vehicle, and the negative Y-axis to the right.

Command: {command}

Based on the BEV image, predict {N_WPs} future waypoints (X, Y) in meters that fulfill the command.

Answer: {"waypoints": [[x1, y1], [x2, y2], ...]}

Emotion Modeling

Analyze the valence, arousal, and dominance (VAD) of the following driving command.
Command: {command}
Output a JSON strictly adhering to this format: {{"valence": v, "arousal": a, "dominance": d}}

Answer: {"valence": v, "arousal": a, "dominance": d}

Joint Fine-tuning <allo> <ego>

You are given two synchronized perception views:

Bird's-Eye View (BEV): covers 120 m ahead (X-axis) and 80 m laterally (Y-axis). The ego vehicle is centered at (0, 0) and faces the positive X-axis (forward). The positive Y-axis points to the left of the ego vehicle, and the negative Y-axis to the right.

Frontal View: shares the same orientation as the BEV — the ego vehicle faces the positive X-axis. X denotes forward distance (m), and Y denotes lateral offset (m). The ego vehicle is located at (0, 0).

Command: {command}

Based on the images and command, perform three tasks in sequence:
1. Analyze the VAD (Valence, Arousal, Dominance) of the command.
2. Locate the target object referred to by the command in the Frontal View.
3. Predict {N_WPs} future waypoints (X, Y) in meters that fulfill the command.

Answer:
The VAD (Valence, Arousal, Dominance) of the command is: {"valence": v, "arousal": a, "dominance": d}
The target object in the frontal view is located at: {"bbox_2d": [x1, y1, x2, y2]}
The predicted future waypoints to fulfill the command are: {"waypoints": [[x1, y1], ...]}

Emotion-Action Alignment <allo> <ego>

You are given two synchronized perception views:

Bird's-Eye View (BEV): covers 120 m ahead (X-axis) and 80 m laterally (Y-axis). The ego vehicle is centered at (0, 0) and faces the positive X-axis (forward). The positive Y-axis points to the left of the ego vehicle, and the negative Y-axis to the right.

Frontal View: shares the same orientation as the BEV — the ego vehicle faces the positive X-axis. X denotes forward distance (m), and Y denotes lateral offset (m). The ego vehicle is located at (0, 0).

Command: {command}

Based on the images and command, Predict {N_WPs} future waypoints (X, Y) in meters that fulfill the command.

Preferred: {"p_waypoints": [[x1, y1], [x2, y2], ...]}
Rejected: {"r_waypoints": [[x1, y1], [x2, y2], ...]}

Allocentric Target Location Estimation <allo_ego_target>

You are given a Bird's-Eye View (BEV) image showing the ego vehicle (red, at origin 0,0) and a target object (yellow).

Bird's-Eye View (BEV): covers 120 m ahead (X-axis) and 80 m laterally (Y-axis).
The ego vehicle is centered at (0, 0) and faces the positive X-axis (forward).
The positive Y-axis points to the left of the ego vehicle, and the negative Y-axis to the right.

Estimate the coordinates of the target object in the image.

Answer: {"target_bev_position": [x_meters, y_meters]}

Figure A4. Overview of the prompt templates utilized in the E3AD framework across different training stages. The figure details the specific instructions and input/output formats for tasks including spatial grounding, depth estimation, emotion modeling, and trajectory planning. Note that the text elements marked in blue (i.e., <ego>, <allo>, <allo_ego_target>, <allo_ego>) represent the special visual tokens corresponding to the egocentric view, allocentric view, and the allocentric view with the target object highlighted, respectively.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF CVPR*, pages 11621–11631, 2020. 2

[2] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie Francine Moens. Talk2car: Taking control of your self-driving car. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 2088–2098, 2019. 2

[3] Thierry Deruyttere, Dusan Grujicic, Matthew B Blaschko, and Marie-Francine Moens. Talk2car: Predicting physical trajectories for natural language commands. *IEEE Access*, 10:123809–123834, 2022. 2, 3

[4] Haicheng Liao, Huanming Shen, Zhenning Li, Chengyue Wang, Guofa Li, Yiming Bie, and Chengzhong Xu. Gpt-4 enhanced multimodal grounding for autonomous driving: Leveraging cross-modal attention with large language models. *Communications in Transportation Research*, 4:100116, 2024. 2

[5] Haicheng Liao, Huanming Shen, Bonan Wang, Yongkang Li, Yihong Tang, Chengyue Wang, Dingyi Zhuang, Kehua Chen, Hai Yang, Chengzhong Xu, et al. Think before you drive: World model-inspired multimodal grounding for autonomous vehicles. *arXiv preprint arXiv:2512.03454*, 2025. 2

[6] Saif M. Mohammad. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia, 2018. 1

[7] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF ICCV*, pages 913–922, 2021. 2