# Counterfactual VLA: Self-Reflective Vision-Language-Action Model with Adaptive Reasoning

## Supplementary Material

This supplementary material is organized as follows.

- Sec. 6 presents additional ablation studies on dataset mixture, token-level loss weighting, and the effect of decoding temperature on adaptive counterfactual reasoning.
- Sec. 7 details the optimization setup, batching strategy, loss weighting, and sequence configuration used for all CF-VLA variants.
- Sec. 8 presents the full instruction prompts given to the VLA model during training and inference and to the VLM expert labeller for generating counterfactual reasoning traces.
- Sec. 9 provides extended qualitative visualizations, including both representative success cases and failure modes of CF-VLA's counterfactual reasoning.

## 6. Additional Ablation Studies

**Importance of counterfactual data filtering.** The counterfactual pipeline deliberately applies CF supervision only to scenes where trajectory quality is bottlenecked by meta-actions, identified by the disagreement between free-generation and pre-filled meta-action rollouts. A straightforward question is ***what if we train the model to generate counterfactual reasoning for all data?*** Fig. 6 compares two `CF-VLA` models that differ only in this selection step, and Tab. 4 reports their final metrics. In the *Whole Dataset* variant, counterfactual reasoning traces are generated for *all* meta-action–labeled samples and used as CF data. In the *Filtered Data* variant, CF traces are generated only for the subset that satisfies the trajectory-disagreement criterion (Sec. 3.3), i.e., cases where improved meta-actions are expected to yield significant trajectory gains. Despite using strictly more labeled CF data, the Whole Dataset model converges more slowly and plateaus at a higher validation minADE. The filtered model attains lower error across almost the entire training horizon and achieves the best final minADE.

When counterfactual reasoning traces are generated only on the filtered subset, the model achieves better minADE and minFDE and a lower corner distance, despite emitting much shorter responses and a lower think rate (average length 125.7 vs. 191.1 tokens; think rate 0.22 vs. 0.67). Generating CF traces on the entire dataset leads to substantially more and longer "Thinking:" segments but does not improve, and in fact slightly degrades, key planning metrics. These results indicate that counterfactual supervision must be targeted: simply adding more CF labels and forcing the model to "think" on every scene introduces redundant or noisy reasoning signals, diluting the impact of informative counterfactual examples and ultimately harming the performance. The rollout–filter–label stage is therefore not merely a data-efficiency optimization, but a crucial component for extracting reliable self-reflection signals.

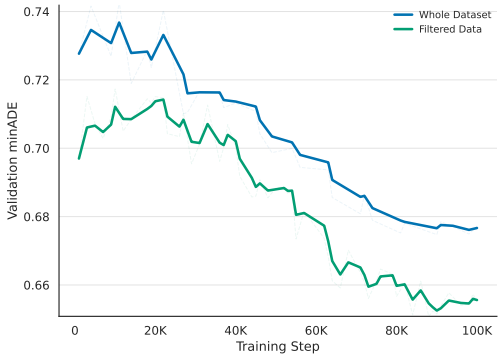


Figure 6. **Effect of counterfactual data filtering.** Validation minADE of CF-VLA on $\mathcal{D}_{\text{val}}$ when counterfactual (CF) labels are generated on the entire meta-action dataset (**Whole Dataset**) versus only on the high-value subset selected by the rollout–filter–label pipeline (**Filtered Data**). Training on CF labels for all scenes slows convergence and converges to a noticeably worse validation error, while restricting CF supervision to the filtered subset yields lower minADE throughout training and a better final model.

**The impact of data mixture.** As shown in Fig. 7(B), removing the large trajectory-only dataset $\mathcal{D}_{\text{traj}}$ and training only on meta-action and CF data (blue curve, $\mathcal{D}_{\text{meta}}^{\times 1} \cup \mathcal{D}_{\text{CF}}^{\times 1}$) causes the model to overfit rapidly and generalize poorly. Even when $\mathcal{D}_{\text{traj}}$ is included, aggressively repeating the smaller meta-action or CF datasets (*e.g.* $\mathcal{D}_{\text{traj}}^{\times 1} \cup \mathcal{D}_{\text{meta}}^{\times 2} \cup \mathcal{D}_{\text{CF}}^{\times 10}$, $\mathcal{D}_{\text{traj}}^{\times 1} \cup \mathcal{D}_{\text{meta}}^{\times 10} \cup \mathcal{D}_{\text{CF}}^{\times 10}$, and $\mathcal{D}_{\text{traj}}^{\times 1} \cup \mathcal{D}_{\text{meta}}^{\times 10} \cup \mathcal{D}_{\text{CF}}^{\times 100}$) eventually hurts validation minADE: the model degrades as duplicated meta-actions / CF traces dominate the training distribution. The best configuration is the natural mixture of each dataset (green, $\mathcal{D}_{\text{traj}}^{\times 1} \cup \mathcal{D}_{\text{meta}}^{\times 1} \cup \mathcal{D}_{\text{CF}}^{\times 1}$), where the data is sampled from the concatenated pool of all samples. This yields both the lowest validation error and the most stable convergence. These results indicate that the large, diverse base trajectory dataset $\mathcal{D}_{\text{traj}}$ is essential for grounding the planner in realistic driving statistics, while a moderate amount of meta-action and counterfactual data is sufficient to teach the model how to verbalize and revise its plans; oversampling these

Table 4. **Effect of our proposed data filtering pipeline.** We use models with route information and train them with the counterfactual reasoning traces on the filtered or the whole training dataset.

| Model | MinADE↓ | AvgADE↓ | MinFDE↓ | AvgFDE↓ | MinIOU↑ (init → edited) | Corner Dist.↓ | Output Length | Think Rate |
|---|---|---|---|---|---|---|---|---|
| `CF-VLA` (w/ route, filtered ds) | 0.6712 | 1.4574 | 1.7988 | 3.9466 | 0.9207→0.9231 | 0.6010 | 125.67 | 0.2190 |
| `CF-VLA` (w/ route, whole ds) | 0.6811 | 1.4185 | 1.8296 | 3.8344 | 0.9207→0.9231 | 0.6128 | 191.14 | 0.6677 |

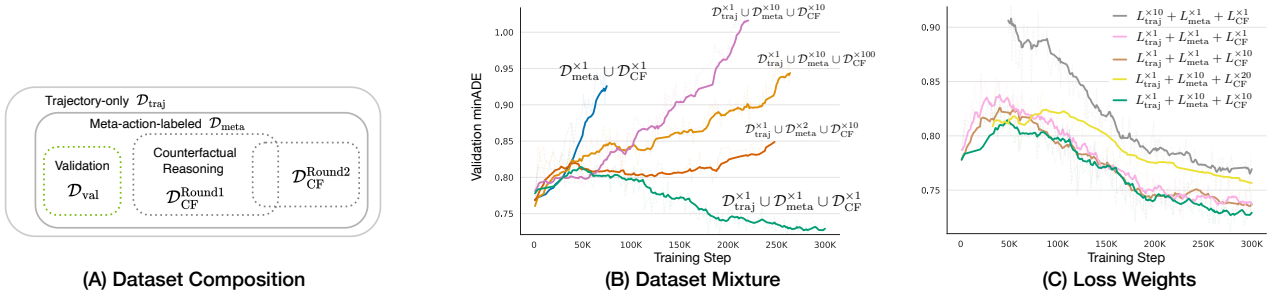


Figure 7. **(A) Dataset composition.** A subset of the meta-action-labeled dataset $\mathcal{D}_{\text{meta}}$ is held out as the validation set $\mathcal{D}_{\text{val}}$. **(B) Ablation on dataset mixture.** $\mathcal{D}^{\times x}$ denotes repeating the dataset $x$ times. For simplicity, $\mathcal{D}_{\text{meta}}$ denotes the training subset of meta-action-labeled data. Training with the natural combination of three tasks (`traj-only`, `meta-traj`, and `CF`) avoids overfitting and yields the best validation performance. **(C) Ablation on loss weights.** $L^{\times y}$ applies a multiplier $y$ to the cross-entropy loss of the corresponding token group. Emphasizing meta-action and CF reasoning tokens with $L_{\text{traj}}^{\times 1} + L_{\text{meta}}^{\times 10} + L_{\text{CF}}^{\times 10}$ stabilizes training and improves trajectory accuracy, whereas putting a large weight only on trajectory tokens degrades performance (the gray line). Further increasing the CF weight to $L_{\text{traj}}^{\times 1} + L_{\text{meta}}^{\times 10} + L_{\text{CF}}^{\times 20}$ induces more thinking (think rate 0.2338 vs. 0.1478 for the $1:10:10$ model) but harms trajectory accuracy, indicating that a higher think rate alone is not a reliable proxy for better control performance.

specialized datasets reduces effective scene diversity and leads to worse trajectory accuracy due to overfitting.

**The impact of loss mixture.** Token-level loss weighting is used to preserve the desired balance across sections (meta-actions, reasoning, and trajectory) in the model's response. Denote the losses on trajectory, meta-action, and counterfactual reasoning tokens as $L_{\text{traj}}$, $L_{\text{meta}}$, and $L_{\text{CF}}$. A configuration such as $L_{\text{traj}}^{\times 1} + L_{\text{meta}}^{\times 10} + L_{\text{CF}}^{\times 10}$ (Fig. 7(C)) applies a $10\times$ multiplier to the meta-action and CF reasoning tokens relative to trajectory tokens and corresponds to the best-performing model. As shown in Fig. 7(C), increasing the weights on meta-action and CF tokens ($L_{\text{meta}}^{\times 10}, L_{\text{CF}}^{\times 10}$) stabilizes training and improves reasoning consistency, whereas emphasizing trajectory tokens alone ($L_{\text{traj}}^{\times 10}$) deteriorates validation error. However, pushing the CF weight further to $L_{\text{traj}}^{\times 1} + L_{\text{meta}}^{\times 10} + L_{\text{CF}}^{\times 20}$ induces more thinking (raising the think rate from $0.1478$ to $0.2338$) but slightly harms trajectory accuracy. This shows that think rate is not monotonically correlated with performance: overly rewarding CF tokens encourages longer and more frequent reasoning traces even when they do not translate into better meta-actions or trajectories. Balanced supervision between reasoning and control, rather than simply "thinking more", is the key to strong performance.

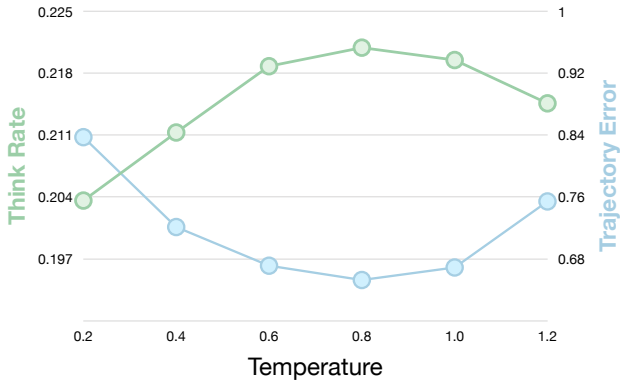


Figure 8. **Effect of decoding temperature on adaptive counterfactual reasoning.** Think rate (left axis) and trajectory error (minADE, right axis) of CF-VLA on the validation set under different sampling temperatures are plotted. Both curves exhibit a strong inverse correlation: temperatures that induce higher think rates generally correspond to lower trajectory error, up to a moderate range around 0.8, while very low or very high temperatures either under-utilize counterfactual reasoning or introduce noisy generations that harm planning accuracy.

**Effect of decoding temperature.** To study how decoding stochasticity interacts with adaptive counterfactual reasoning, this ablation varies the sampling temperature at infer-

Table 5. **Ablation study on the training datasets.** We train `CF-VLA` (w/ route, round 2, 3 ds) with 3 datasets: $\mathcal{D}_{\text{traj}}$,$\mathcal{D}_{\text{meta}}$,$\mathcal{D}_{\text{CF}}^{\text{Round2}}$ (3 ds). We train `CF-VLA` (w/ route, round 2, 4 ds) with 4 datasets: $\mathcal{D}_{\text{traj}}$,$\mathcal{D}_{\text{meta}}$,$\mathcal{D}_{\text{CF}}^{\text{Round1}}$,$\mathcal{D}_{\text{CF}}^{\text{Round2}}$ (4 ds). ↓ lower is better, ↑ higher is better.

| Model | ADE↓ Min (Avg) | FDE↓ Min (Avg) | Corner Dist.↓ | Collision↓ | Off-road↓ | IOU↑ init→edited | Output Len. (Think Rate) |
|---|---|---|---|---|---|---|---|
| `meta-act` (w/ route) | 0.7263 (1.4612) | 1.9561 (3.9269) | 0.6600 | 0.0196 | 0.0619 | 0.9236 | 87.20 (–) |
| `CF-VLA` (w/ route, round1) | **0.6712** (1.4574) | **1.7988** (3.9466) | 0.6010 | 0.0177 | 0.0593 | 0.9207→0.9231 | 125.67 (0.219) |
| `CF-VLA` (w/ r., round2, 3 ds) | 0.6813 (**1.3898**) | 1.8291 (**3.7474**) | 0.6168 | **0.0174** | **0.0585** | **0.9238→0.9276** | 109.36 (0.123) |
| `CF-VLA` (w/ r., round2, 4 ds) | 0.6776 (1.4405) | 1.8108 (3.9017) | **0.6083** | 0.0176 | 0.0588 | 0.9186→0.9241 | 140.23 (0.299) |

ence while keeping all other settings fixed. As shown in Fig. 8, trajectory error monotonically decreases when increasing the temperature from 0.2 to 0.8, but starts to rise again at 1.0 and 1.2, forming a U-shaped curve. The think rate changes more mildly and peaks around 0.8. Across temperatures, think rate and trajectory error are strongly and inversely correlated: configurations that elicit more counterfactual reasoning tend to produce more accurate trajectories. Very low temperatures make the model overly deterministic—once it settles on emitting "Action:" for many scenes, it under-explores the "Thinking:" branch, limiting the benefit of self-reflection and leading to suboptimal trajectories. In contrast, overly high temperatures inject excessive randomness into meta-actions and reasoning traces, which degrades trajectory quality despite a similar average think rate. A moderate temperature therefore provides the best trade-off, preserving stable planning while allowing CF-VLA to invoke counterfactual reasoning where it is most beneficial.

**Training dataset composition.** As shown in Tab. 5, comparing `CF-VLA` (w/ route, round 2, 3 datasets) and `CF-VLA` (w/ route, round 2, 4 datasets) reveals an important efficiency trade-off. Including the first-round CF data (4 datasets) slightly improves geometric metrics, but degrades average errors, IOU, and safety metrics, and nearly doubles the think rate and output length. This suggests simply adding more CF traces (especially when they are not on-policy rolled out) is not always beneficial, and `CF-VLA` (w/ route, round2, 3 datasets) offers a better balance between accuracy, safety, and test-time efficiency.

## 7. Training Hyper-parameters

All CF-VLA variants are trained from the `Qwen2.5-VL-3B-Instruct` backbone using the optimization and data settings listed below.

**Optimization.** Models are optimized with AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$ and weight decay $0.01$. The initial learning rate is $1 \times 10^{-5}$ with a cosine decay schedule and $2{,}000$ warm-up steps. For `CF-VLA` models, we set the learning rate to $5 \times 10^{-6}$. Training runs for $300{,}000$ optimization steps with gradient norm clipping at $1.0$.

**Batching and precision.** The per-device training batch size is $1$ with gradient accumulation set to $1$ (global batch size equal to the number of GPUs). Training uses bfloat16 without gradient checkpointing and activation checkpointing. Distributed training uses fully sharded data parallelism. The VLM side uses FlashAttention-2 for all attention layers with bfloat16 activations. We use $64$ NVIDIA A100 GPUs and thus batch size is $64$ for each experiment.

**Loss weighting and data mixture.** Training minimizes token-level cross-entropy over the assistant outputs only. Different token groups are weighted as

$$w_{\text{act}} : w_{\text{meta}} : w_{\text{CF}} = 1 : 10 : 10,$$

for trajectory tokens (`act`), meta-action tokens (`meta`), and counterfactual reasoning tokens (`cot`), respectively. For counterfactual samples, the loss on the first (uncorrected) meta-action block is masked out. Unless otherwise specified, the `meta-act` models are trained on the mixture $D_{\text{traj}} \cup D_{\text{meta}}$ and CF-VLA models are trained on the mixture $D_{\text{traj}} \cup D_{\text{meta}} \cup D_{\text{CF}}$ without oversampling (repetition) of any dataset.

**Sequence configuration.** Each training example uses $4$ frames per camera over a 2s history window (2Hz), with resolution set to $448 \times 796$. The model inputs $16$ history waypoints and predicts $64$ future waypoints.

## 8. Instruction Template

The following box shows the full instruction prompt (system and user messages) used to query the VLA models for producing meta-actions and the optional counterfactual reasoning traces during training and evaluation:

```
<|im_start|>system
You are a helpful autonomous driving assistant.

You will receive videos and a history trajectory as context and the goal is to produce
    future trajectories that are reasonable in the driving scene, safe and consistent
    with the history trajectory.

The videos are captured by cameras mounted on the self-driving car, showing the driving
    scene over the past 2 seconds. Camera camera_front_wide_120fov shows the front of the
     car with a 120 degree field of view. Camera camera_front_tele_30fov shows the front
    of the car with a 30 degree field of view, the zoom-in view of
    camera_front_wide_120fov.

The history trajectory is 1.6 seconds of past motion for the self-driving car, sampled
    at 10 Hz, and includes x, y, z coordinates and heading (all four values are relative
    to the state at the current time). It is enclosed between <|traj_history_start|> and
    <|traj_history_end|>, and is encoded into a continuous latent representation using 1
    token by the history trajectory encoder.

The future trajectory covers 6.4 seconds of predicted motion, also sampled at 10 Hz, and
     includes x, y, z coordinates and heading (relative to the state at the current time)
    . The future trajectory is represented by a set of 6 discrete tokens enclosed between
     <|traj_future_start|> and <|traj_future_end|>. Each token is sampled from the
    vocabulary of the future trajectory tokenizer, which will decode the 6 tokens into a
    6.4-second future trajectory.

You may receive different types of tasks given the context, including:
- Producing a future trajectory given the video and history trajectory.
- Proposing the meta actions in the driving process and generate the future trajectory
    according to the meta actions.
- Reflecting on the correctness of the meta actions according to the information in the
    scene and correcting the meta actions if necessary.

Meta actions are a set of high-level descriptions of the vehicle's behavior. Meta
    actions are grouped into three categories:
- **longitudinal**: ["Accelerate", "Decelerate", "Keep Speed", "Wait", "Reverse"]
- **lateral**: ["Straight", "Left Turn", "Right Turn"]
- **lane**: ["Keep Lane", "Left Lane Change", "Right Lane Change"]

The meta actions must partition the 6.4-second planning horizon (from 0.0s to 6.4s)
    without gaps or overlaps *within each group*. Groups are optional; omit any group not
     relevant to the current scene. List the meta actions in each group in bullet-point
    format, like the example below:

- longitudinal
  - 0.0s-0.9s: Keep Speed
  - 0.9s-6.4s: Accelerate
- lateral
  - 0.0s-6.4s: Straight
- lane
  - 0.0s-4.4s: Left Lane Change
  - 4.4s-6.4s: Keep Lane
```

```
Each time interval (in seconds) is specified as `start-end`, followed by a colon and an
    action within the group.

If you are tasked to propose the meta actions, you should first generate the meta
    actions before generating the future trajectory. If you think the first proposed meta
     actions are **incorrect**, generate a "Thinking:" section that briefly reflects on
    your meta-actions and explains the problem. Then provide a second "Meta Actions:"
    section with a corrected meta-action plan, and finally generate the future trajectory
     in the "Action:" section. If you are confident that the first meta actions are **
    valid**, you may skip the reflection section and go directly to the trajectory
    generation.

Respond appropriately based on the given input and task.<|im_end|>
<|im_start|>user
The video for camera 1 (camera_front_wide_120fov) is: <|vision_start|><|video_pad|>...<|
    vision_end|>The video for camera 6 (camera_front_tele_30fov) is: <|vision_start|><|
    video_pad|>...<|vision_end|>The trajectory history is: <|traj_history_start|><|
    traj_history|><|traj_history_end|>First, list the meta actions in the driving process
    . Then, generate a possible future trajectory. If the meta-action is unsafe or
    incorrect, reflect on it and provide a corrected one. Use the output format below.

Meta Actions:
[meta actions]

Thinking: (optional)
[text]

Meta Actions: (optional)
[corrected meta actions]

Action:
<|traj_future_start|>[6 tokens]<|traj_future_end|><|im_end|>
<|im_start|>assistant
```

The following box shows the full instruction prompt (system and user messages) given to the VLM expert labeller (`Qwen2.5-VL-72B-Instruct`) for generating counterfactual reasoning traces and refining meta-action annotations:

```
<|im_start|>system
You are an autonomous-driving labeling assistant. Your job is to **diagnose and correct
    ** prediction errors **using visual cues** in the provided frames/videos. You **must
    not use** or reveal any privileged knowledge beyond what is visually observable. Be
    concrete, timestamped, and concise.<|im_end|>
<|im_start|>user
You will be presented with a driving scenario where you need to choose the correct meta-
    actions. Your task is to analyze the situation and provide reasoning about why the
    previous meta-actions are less preferable to the expert meta-actions.

## Observation

You will receive: the PREDICTED meta-actions (longitudinal, lateral, lane) with time
    intervals as well as the EXPERT meta-actions. Meta-actions are a set of high-level
    descriptions of the vehicle's behavior. Meta-actions are grouped into three
    categories with these options in each category:

- **longitudinal**: ["Accelerate", "Decelerate", "Keep Speed", "Wait", "Reverse"]
```

```
- **lateral**: ["Straight", "Left Turn", "Right Turn"]
- **lane**: ["Keep Lane", "Left Lane Change", "Right Lane Change"]

The meta-actions must partition the 6.4-second planning horizon (from 0.0s to 6.4s).
    Groups are optional - omit any group not relevant to the current scene. The meta-
    actions are in bullet-point format. Each time interval (in seconds) is specified as `
    start-end`, followed by a colon and an action within the group. Like the example
    below (note that this example is not related to the scene you are watching):

- longitudinal
  - 0.0s-0.9s: Keep Speed
  - 0.9s-6.4s: Accelerate
- lateral
  - 0.0s-6.4s: Straight
- lane
  - 0.0s-4.4s: Left Lane Change
  - 4.4s-6.4s: Keep Lane

You will receive: VIDEOS/FRAMES from two cameras. The images are captured by cameras
    mounted on the self-driving car, showing the driving scene at this moment. Camera `
    camera_front_wide_120fov` shows the front of the car with a 120 degree field of view.
     Camera `camera_front_tele_30fov` shows the front of the car with a 30 degree field
    of view, the zoom-in view of `camera_front_wide_120fov`.

## Task

Your task is to provide a detailed counterfactual reasoning trace as an internal self-
    reflection that demonstrates your reasoning process for the current situation. Your
    reasoning should:

1. Start with analyzing the driving scenario and the goal. Highlight any relevant visual
     clues, constraints, or consequences from the scenario. For example, the lane
    markings, the traffic lights, the vehicles, the pedestrians, the road conditions, etc
    .
2. Discuss the predicted meta-actions, explain why they may be less optimal, think about
     the expected consequences. If the predicted meta-actions are already close to the
    expert meta-actions, it's OK to simply acknowledge that the meta-actions are already
    good and explain why.
3. Justify why the expert action may be more preferable, while not indicating you have
    access to the expert action. State the possible changes to the predicted meta-actions
     to make it closer to the expert action.


Hard rules:

- **Never mention** ground truth, labels, "expert", "GT", "dataset", or phrases implying
     access to future or privileged information. The ground truth meta-actions are only
    used to help you understand the scene and the predicted meta-actions. Do not discuss
    the ground truth meta-actions in the reasoning to avoid information leakage. The
    ultimate goal is to insert the reasoning after the predicted meta-actions and before
    the GT meta-actions so I can build the dataset that have 1) wrong meta-actions, 2)
    the reflection, and 3) the corrected meta-actions. Therefore, it's strictly forbidden
     to imply that you have access to the ground truth meta-actions in the reasoning. Do
    not discuss anything like "according to the images", "the trajectory suggests" or "
    the GT meta-actions suggest", "the vehicles seem like [some future actions]", etc.
- **Anchor** every claimed correction with one or more specific cues. Always ground your
     decisions based on the provided images. Discuss the reasoning in the context
```

```
    according to the given images. Do not simply say "which meta action is wrong and I
    should change it to something".
- Use **one short paragraph** for `reasoning` ( ≤ 80 words, no line breaks). Your
    reasoning should be concise and to the point. Do not list the final (correct) meta-
    actions. Do not use markdown or other formatting in the reasoning. Avoid meta-
    commentary about being an AI. Use natural, step-by-step reasoning. Focus on logical
    decision-making. Directly write the self-reflection reasoning, no extra headings,
    disclaimers, or external notes.
- If uncertain or the predicted meta-actions are similar to the ground truth meta-
    actions, you can simplify the reasoning.
- It's OK to find that the predicted meta-actions are already close to the expert meta-
    actions, and you don't need to change too much.

## Scenario Data
The video captured by camera_front_wide_120fov: <|vision_start|><|video_pad|>...<|
    vision_end|>

The video captured by camera_front_tele_30fov: <|vision_start|><|video_pad|>...<|
    vision_end|>

The expert meta-actions are:
- longitudinal
  - 0.0s-0.5s: Keep Speed
  - 0.5s-5.0s: Decelerate
  - 5.0s-5.9s: Keep Speed
  - 5.9s-6.4s: Accelerate
- lateral
  - 0.0s-5.0s: Left Turn
  - 5.0s-6.4s: Straight
- lane
  - 0.0s-5.5s: Keep Lane
  - 5.5s-6.4s: Left Lane Change

The predicted meta actions are:
- longitudinal
  - 0.0s-6.4s: Keep Speed
- lateral
  - 0.0s-3.4s: Left Turn
  - 3.4s-6.4s: Straight
- lane
  - 0.0s-6.4s: Keep Lane

Please propose counterfactual reasoning on the predicted meta-actions.<|im_end|>
<|im_start|>assistant
```

## 9. Additional Visualization

In this section, we provide more visualization results for model CF-VLA (w/ route, round 2, 3 datasets).

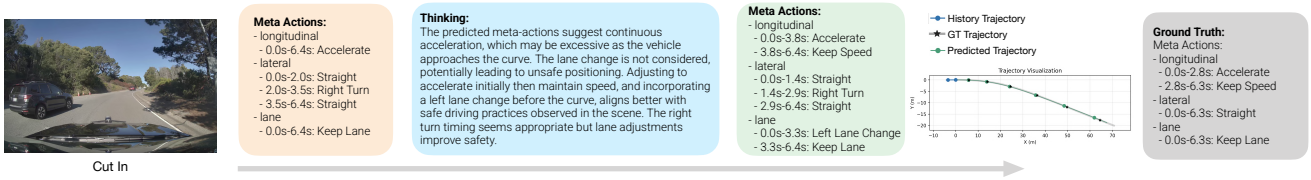


Figure 9. **Counterfactual reasoning for a cut-in near a curve.** The scene shows the ego vehicle following traffic on a two-lane road where a right-hand curve and potential cut-in are ahead. The initial meta-actions (orange) prescribe continuous acceleration through the entire horizon and keep the ego in its current lane while executing the right turn, which could build up unnecessary speed as the vehicle enters the curve and leaves no lateral margin to surrounding traffic. Conditioned on these meta-actions and the video, the counterfactual reasoning step (blue) recommends accelerating only in the early phase and then maintaining speed, while also introducing a left lane change before the curve to obtain a safer positioning. The revised meta-actions (green) thus shorten the acceleration period, add a brief left lane change followed by lane keeping, and retain a similar right-turn timing; the resulting trajectory remains very close to the ground truth (gray) in space while adopting a more explicitly safety-oriented lane strategy. This example highlights how CF-VLA can simultaneously reshape longitudinal and lane-level intent around curves and cut-ins, yielding a conservative yet efficient plan that stays consistent with the scene.

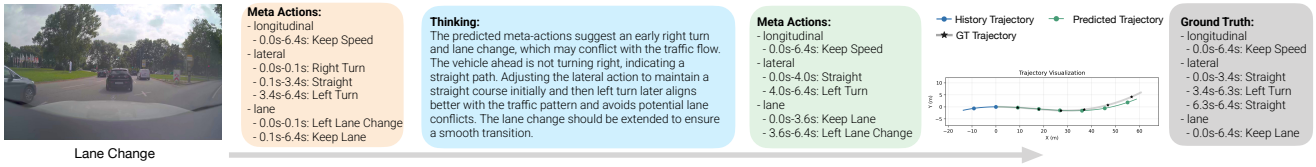


Figure 10. **Self-correction in a lane-change scenario.** The scene shows the ego vehicle following a lead car on an urban road while approaching a junction where the traffic ahead continues straight and then bends left. The initial meta-actions (orange) incorrectly introduce a brief right turn and lane change at the beginning of the horizon (around 0–0.1 s), followed by a later left turn, creating an unnecessarily complex maneuver that could conflict with the surrounding traffic pattern. Conditioned on these meta-actions and the video frames, the counterfactual reasoning step (blue) notes that the lead vehicle is not turning right and recommends maintaining a straight lateral course for longer before initiating a smoother left turn and extended lane change. The revised meta-actions (green) thus remove the spurious early right turn, delay the left-turn phase to around 4.0 s, and stretch the lane-change interval, yielding a predicted trajectory that closely follows the ground-truth path (gray) while preserving similar speed. This example illustrates how CF-VLA can use visually grounded counterfactual reasoning to correct an initially misaligned turning intent and produce a more coherent, traffic-consistent plan.

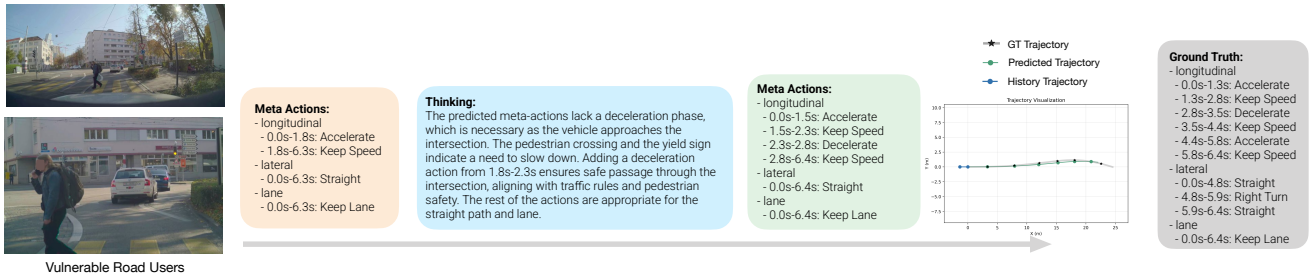


Figure 11. **Counterfactual reasoning near vulnerable road users and queuing traffic.** The scene shows a pedestrian finishing a crossing at an intersection while a queue of vehicles waits ahead. The initial meta-actions (orange) instruct the ego to accelerate from standstill and then maintain speed from 1.8–6.3 s, lacking any explicit deceleration phase before reaching the intersection and the queue. Conditioned on these meta-actions and the telephoto views, the reasoning step (blue) notes the pedestrian crossing and yield signage and recommends inserting a short deceleration interval, yielding revised meta-actions (green) that accelerate from 0–1.5 s, briefly cruise, then decelerate between 2.3–2.8 s before keeping speed while joining the line of cars. The resulting trajectory closely follows the longitudinal profile, where the human driver also accelerates once the pedestrian has cleared the crosswalk and then slows to queue at the intersection. This example illustrates that CF-VLA's self-reflection can enrich an initially over-optimistic plan with an appropriate queuing deceleration, achieving behavior that is both pedestrian-aware and consistent with surrounding traffic.

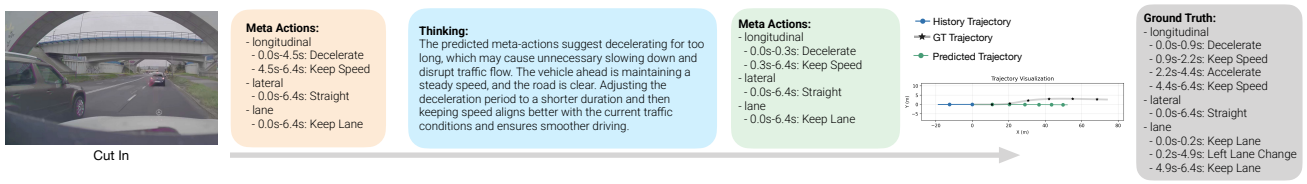


Figure 12. **Conservative counterfactual reasoning in a cut-in scenario.** The scene shows highway driving with a lead vehicle in the ego lane and another vehicle overtaking in the adjacent left lane. The initial meta-actions (orange) propose a long deceleration from 0–4.5 s followed by keeping speed, even though the lead vehicle is already maintaining a steady pace and the road ahead is largely clear. Conditioned on these meta-actions and the videos, the reasoning step (blue) correctly identifies this as overly cautious and recommends shortening the deceleration period, switching to a plan that briefly decelerates from 0–0.3 s and then keeps speed (green), yielding a smoother and less disruptive longitudinal profile that better matches the traffic flow. However, both the initial and revised meta-actions retain a "Keep Lane" decision for the full horizon, whereas the human driver (gray) performs a left lane change and subsequent acceleration to pass slower traffic. As a result, the predicted trajectory remains safe but more conservative and less efficient than the human expert, likely prioritizing collision avoidance with the adjacent overtaking vehicle over an aggressive lane change.

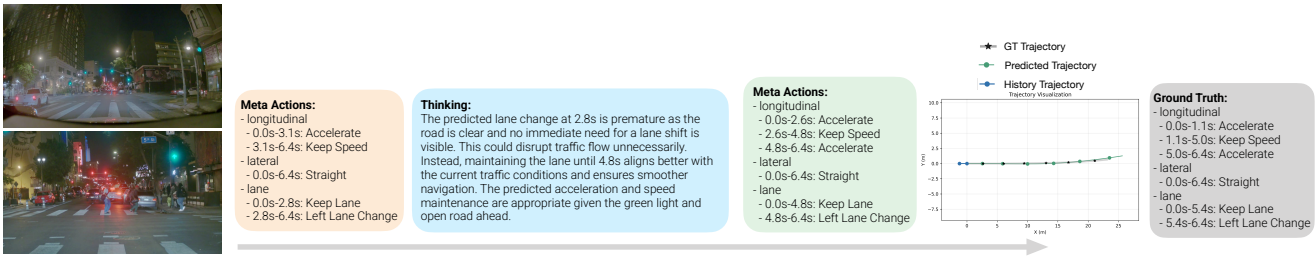


Figure 13. **Refining lane-change timing with counterfactual reasoning at night.** The scene shows nighttime urban driving through a signalized intersection with pedestrians near the crosswalk and open road beyond. The initial meta-actions (orange) plan to accelerate through the green light and initiate a left lane change at 2.8 s, causing the ego vehicle to start shifting lanes shortly after entering the intersection. Conditioned on these meta-actions and the video input, the counterfactual reasoning step (blue) judges this maneuver as premature given the clear lane ahead and potential disruption to traffic, and recommends extending the keep-lane phase before moving left. The revised meta-actions (green) therefore maintain lane until about 4.8 s, then perform a later left lane change while preserving a smooth accelerate–cruise–accelerate longitudinal pattern. As shown in the trajectory visualization, this produces a predicted trajectory whose lane-change timing closely matches the human driver (gray), who also stays in the original lane until well past the intersection before merging left. This example highlights how CF-VLA can use counterfactual self-reflection to delay unnecessary early lane changes, leading to behavior that is both smoother and more consistent with human driving.

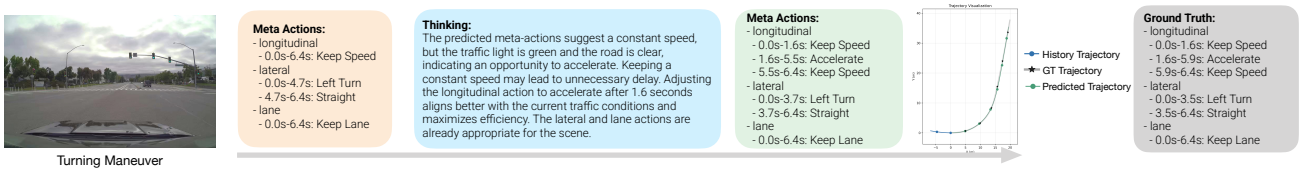


Figure 14. **Refining acceleration timing in a turning maneuver with counterfactual reasoning.** The scene shows the ego vehicle approaching a large signalized intersection with a green light and no visible cross-traffic. The initial meta-actions (orange) plan to keep a constant speed for the entire 6.4 s horizon while executing a left turn followed by a straight segment, which is safe but under-utilizes the open intersection and may lead to unnecessary delay. Conditioned on these meta-actions and the video input, the counterfactual reasoning step (blue) correctly notes that the road is clear under green and recommends introducing an acceleration phase after roughly 1.6 s, while leaving the lateral and lane decisions unchanged. The revised meta-actions (green) therefore maintain speed initially, then accelerate through the middle of the intersection before returning to a cruising speed, producing a predicted trajectory that closely matches the human driver's motion (gray) in both lateral path and longitudinal profile. This example illustrates how CF-VLA can inject a well-timed mid-intersection acceleration through self-reflection, improving efficiency without sacrificing safety.

We also show the failure cases:

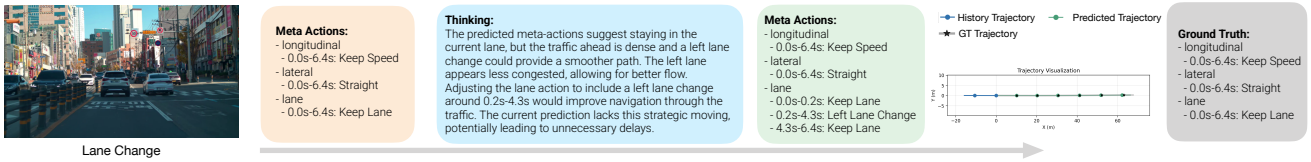


Figure 15. **Failure case of CF-VLA's counterfactual reasoning at a straight lane.** The scene shows the ego vehicle on a straight multi-lane urban road with moderate traffic ahead. The initial meta-actions (orange) propose a reasonable plan that keeps speed and lane throughout the horizon. However, conditioned on these meta-actions and the video input, the counterfactual reasoning step (blue) argues that the left lane "appears less congested" and recommends inserting a left lane change between 0.2 s and 4.3 s to "improve navigation through the traffic." The revised meta-actions (green) follow this suggestion and introduce a left lane change, causing the predicted trajectory to drift into the adjacent lane, while the ground-truth driver in fact keeps lane and continues straight, as indicated by the trajectory visualization and ground-truth meta-actions (gray). This example illustrates that CF-VLA can occasionally over-correct a safe plan based on a misinterpretation of visual cues, hallucinating a beneficial lane change where the human driver does not perform one. Such failure cases highlight the need for better calibration between counterfactual reasoning, route-level intent, and true traffic semantics.

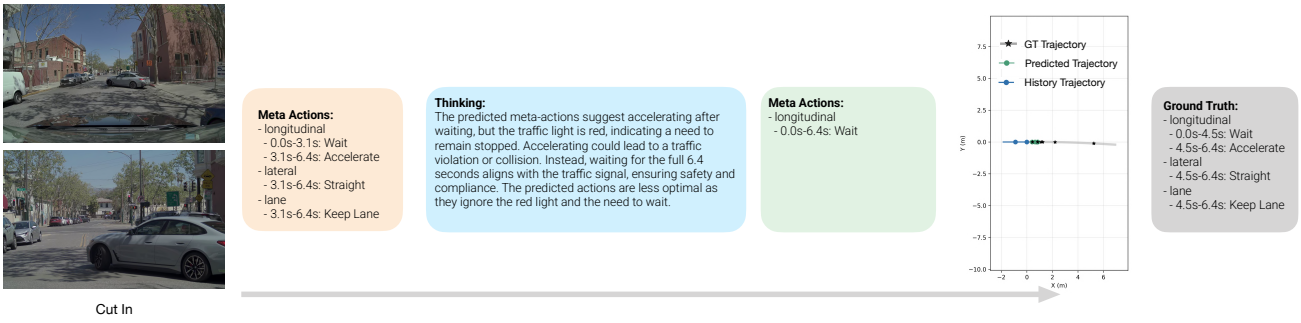


Figure 16. **Failure case of CF-VLA's counterfactual reasoning at a cut-in.** The scene shows the ego vehicle waiting while a nearby vehicle cuts into the lane ahead at an urban intersection. The initial meta-actions (orange) propose to wait until about 3.1 s and then accelerate straight in the current lane from 3.1–6.4 s, which qualitatively resembles the human driver who eventually moves off when the vehicle above leaves. Conditioned on these meta-actions and the video frames, the counterfactual reasoning step (blue) incorrectly concludes that any acceleration would "ignore the red light" and recommends waiting for the full 6.4s to ensure safety and compliance. The revised meta-actions (green) therefore switch to a pure "wait" plan over the entire horizon, producing a predicted trajectory that remains stopped, whereas the ground-truth trajectory (gray) waits only until about 4.5 s before accelerating and clearing the intersection. The model misinterprets the scenario by observing the traffic light and ignores that the major factor causing the waiting behavior is the vehicle above. This example illustrates how CF-VLA's self-reflection can sometimes become overly conservative and miscalibrated with respect to the spatial relationship between objects and traffic lights, degrading performance by overriding a reasonable initial plan.