average reduction in noise $\mathrm{SNR}\left(\Delta \log \mathrm{SNR}_{\mathrm{t},\tau}\right)$ for waypoints $\tau \in P_i$. This allows us to compute a per-step advantage by aggregating future patch rewards:

$$A_t = \sum_{i=1}^{N} w_{t,i} \widehat{R_\iota}$$

For a group of (G) rollouts, we then compute a per-step, leave-one-out, group-relative advantage $\hat{A}_{t,\mathrm{rel}}$ by subtracting the group-mean advantage (excluding the current sample) from A_t and normalizing. The final objective is:

$$L_{\mathrm{GRPO}}(\phi, \psi) = -E\left[\sum_{t=1}^{T} \hat{A}_{t,\mathrm{rel}}\left(\log \pi_\phi\left(s_t|C,t\right) + \log \pi_\psi\left(\Delta s_t|x_t,t,C\right)\right)\right] + \lambda_\phi \mathrm{KL}\left(\pi_\phi|\pi_\phi^{\mathrm{ref}}\right)$$
$$+ \lambda_\psi \mathrm{KL}\left(\pi_\psi|\pi_\psi^{\mathrm{ref}}\right)$$

where $\left(\pi_\phi^{\mathrm{ref}} = \mathcal{N}(1, \sigma_{\mathrm{ref}}^2)\right)$ and $\left(\pi_\psi^{\mathrm{ref}} = \mathcal{N}(0, I)\right)$; $\left(\lambda_\phi, \lambda_\psi > 0\right)$ control conservativeness and are distinct from diffusion variances (β_t).

3.4 TSDP Inference Process

A trajectory is generated via an iterative procedure combining the frozen backbone with the two learned policies.

Initialization. Compute anisotropy $r_\tau = \sigma\left(f_\rho(S_{\mathrm{stat}}[\tau])\right)$ and form$\left(\beta_{t,\tau} = \beta_t^{\mathrm{base}} r_\tau\right)$. For the schedule policy, obtain the whole sequence $\{s_t\}_{t=1}^{T}$ with $s_t = \mu_\phi(z_t)$ (or sample once per step); clip eachs_t to $[s_{min}, s_{max}]$ set $\beta'_{t,\tau} = \beta_{t,\tau} \cdot s_t$. Sample $x_T \sim \mathcal{N}(0, I)$.

Iterative denoising (for $t = T, \ldots, 1$).

(a) Predict $\epsilon_\theta(x_t, t, C)$ and compute the VP score approximation

$$s_\theta(x_t, t, C) \approx -\frac{\epsilon_\theta(x_t, t, C)}{\sigma'_{t,\tau}} \quad \text{(element-wise in } \tau)$$

where $\sigma'_{t,\tau} = \sqrt{1 - \overline{\alpha'_{t,\tau}}}$ and

$$\overline{\alpha'_{t,\tau}} = \prod_{u=1}^{t}\left(1 - \beta'_{u,\tau}\right)$$

(b) Obtain the score-correction action $\Delta s_t = \mu_\psi(x_t, t, C)$ (or sample once), and form

{\tilde{s}}_{\mathrm{\theta}} = s_{\mathrm{\theta}} + \Delta s_t ($clip|\Delta s_t|_2$ if needed).

(c) Take one DPM-Solver++ (VP) step using $\tilde{s}_\theta$ and the schedule $\{\beta'_{u,\tau}\}_{u=1}^{t}$ to obtain x_{t-1}.

Output. The final denoised sample x_0 is returned as the planned trajectory. (No gradients are back-propagated through the solver; policies are trained via policy gradients only.)

3.5 Implementation Details

All components are implemented in PyTorch. The backbone (ϵ_θ) is a Transformer denoiser with encoders for lanes/agents/map; (T=1000) steps for training and 10–20 DPM-Solver++ steps for inference. The anisotropy network $\left(f_\rho\right)$ is a 2-layer MLP; bounds ($r_{min} = 0.5, r_{max} = 1.5$) regularizers ($\lambda_{\text{mean}} = 1.0, \lambda_{\text{smooth}} = 0.1$).

Policies $\left(\pi_\phi, \pi_\psi\right)$ are lightweight Gaussian MLPs; we use clips ($s_t \in [0.9,1.1]$) and ($|\Delta s_t|_2 \leq 2.0$). Training follows two phases: Phase-1 ($\epsilon - prediction$ + $regularization$; AdamW, lr (5×10^{-4}), batch 128, ~500 epochs) and Phase-2 (GRPO; AdamW, lr (1×10^{-4}), group size ($G = 8/16$),$\left(\lambda_\phi = 0.1, \lambda_\psi = 0.05\right)$, ~100 epochs). Experiments are conducted in closed-loop simulation on nuPlan.

4. Research Project Plan

This chapter documents the dataset, simulator, implementation details, baselines, and evaluation protocol used to assess the Time-Series Diffusion Planner (TSDP). All symbols and procedures are consistent with Chapter 3. We emphasize reproducibility: we fix seeds, publish scenario token lists and filtering configs, and specify all preprocessing and coordinate conventions.

4.1 Dataset and Simulation Environment

We use the nuPlan devkit and its closed-loop simulator for all experiments. Scenarios are rolled out with reactive agents, and metrics are computed from the executed ego trajectory rather than open-loop replays. This design matches the planning problem's causal nature and is critical for evaluating control policies learned in Phase 2.

To enable multi-seed runs and extensive ablations on commodity hardware, we adopt nuPlan-mini. It retains the data format, simulator, and metric APIs of full nuPlan while substantially reducing storage and I/O overhead, which is especially important when training the GRPO policies. We construct scenarios with the devkit's ScenarioBuilder using default filters and a 70/15/15 split at the scenario level (no token overlap across splits). The exact counts may vary slightly by installation; we release the full token lists and filter configs with our code so others can reproduce our splits exactly.

Raw logs are resampled to 10 Hz using cubic-spline interpolation for positions and linear interpolation for velocities/accelerations. We then recompute jerk from the resampled signals to avoid aliasing. Each scenario provides 2 s of history and an 8 s planning horizon (i.e., (H=80) waypoints) for closed-loop rollout.

4.2 Implementation Details

Hardware & software.

Unless stated otherwise: 1 (\times) high-end GPU (e.g., A100-80 GB or RTX 4090-24 GB), 32 CPU cores, 256 GB RAM, Python 3.10, PyTorch 2.x + CUDA 12.x, and the nuPlan devkit (v1.1). Mixed precision (fp16) and gradient clipping are enabled.

Phase 1 (diffusion pretraining).

Backbone: Transformer denoiser ϵ_θ with lane/agent/map encoders (Chapter 3).

Forward schedule (cosine VP): we use the cosine cumulative $\overline{\alpha}(\tau)$ with offset ($s = 0.008$) and ($T = 1000$) training steps,

$$\overline{\alpha}(\tau) = \frac{\cos^2\left(\frac{\tau/T + s}{1+s} \cdot \frac{\pi}{2}\right)}{\cos^2\left(\frac{s}{1+s} \cdot \frac{\pi}{2}\right)}, \quad \beta_t^{\text{base}} = \text{clip}\left(1 - \frac{\overline{\alpha}(t)}{\overline{\alpha}(t-1)}, 0, \beta_{max}\right), \quad \beta_{max} = 0.999$$

Time-series anisotropy: $r_\tau = \sigma\left(f_\rho(S_{\text{stat}}[\tau])\right) \in [0.5,1.5]$, with regularizers $\lambda_{\text{mean}} = 1.0, \lambda_{\text{smooth}} = 0.1$ and gradient clipping on ∇_{r_τ}.

Loss & optimizer: $\left(L_{\text{PhaseA}} = E[|\epsilon - \epsilon_\theta|^2] + L_{\text{reg}}\right)$; AdamW (lr ($5 \times 10^{-4}$), weight decay ($10^{-4}$), cosine decay), batch 128, ~500 epochs.

Coordinate frame & normalization: all states are in an ego-centric frame (heading-aligned at ($t = 0$)); positions/velocities/accelerations are z-scored on the training set.

Phase 2 (GRPO fine-tuning).

Policies.

1. Schedule scaling $\left(\pi_\phi(s_t|C,t) = \mathcal{N}\left(\mu_\phi(z_t), \Sigma_\phi(z_t)\right)\right)$ with ($z_t = [\text{PE}(t/T), \text{global}(C)]$).
2. Score correction $\left(\pi_\psi(\Delta s_t|x_t,t,C) = \mathcal{N}(\mu_\psi, \Sigma_\psi)\right)$.

   global(C) features (32-D):

   (i) history summary (mean/var of speed, abs yaw-rate, accel over last 1 s; past-stop flag);

   (ii) route geometry (mean/var of route curvature over $[0, \bar{v}H\Delta t]$, turn-type one-hot);

   (iii) occupancy (neighbor counts in front/side/rear sectors for radii $R \in \{10, 20, 30\}\, m$);

   (iv) traffic context (upcoming signal states within 50/100 m); (v) speed-limit cues (normalized limit, limit-excess indicator, slow-zone flag). All features are z-scored; no future information is used.

Action constraints: $s_t \in [0.9,1.1]$ (hard clip), $|\Delta s_t|_2 \leq 2.0$.

Patch construction & rewards.

The horizon is split into $N = 8$ contiguous patches of equal duration. For patch P_i, we compute safety R_{safe} via signed distance with a speed-dependent threshold $d_0 = d_{\text{base}} + k_v v_{\text{ego}}$, comfort via jerk ($l_1$ on third finite differences), and progress $\Delta s_i / \Delta t_i$. Each component is batch-z-scored and combined with weights $(w_{\text{safe}}, w_{\text{comf}}, w_{\text{prog}})$.

Step-wise credit assignment.

We use ΔlogSNR step-to-patch attribution (Chapter 3) to form normalized weights (w_{t,i}). The per-step advantage includes the same temporal discount γ as the trajectory return:

$$A_t = \sum_{i=1}^{N} w_{t,i} \gamma^{i-1} \hat{R}_i \,, \quad \gamma = 0.98$$

With a leave-one-out group baseline, the GRPO loss is

$$L_{\text{GRPO}}(\phi, \psi) = -E_{C,\text{rlot},t}\left[\hat{A}_{t,\text{rel}}\left(\log \pi_\phi\,(s_t|C,t) + \log \pi_\psi\,(\Delta s_t|x_t,t,C)\right)\right] + \lambda_\phi \mathrm{KL}\left(\pi_\phi|\pi_\phi^{\text{ref}}\right) + \lambda_\psi \mathrm{KL}\left(\pi_\psi|\pi_\psi^{\text{ref}}\right)$$

with references $\pi_\phi^{\text{ref}} = \mathcal{N}(1, \sigma_{\text{ref}}^2)$ and $\pi_\psi^{\text{ref}} = \mathcal{N}(0, I)$. We do not backpropagate through the ODE solver; policies are trained via policy gradients only.

Optimization.

AdamW lr (1×10^{-4}), group size $G \in \{8, 16\}$; $\lambda_\phi = 0.1, \lambda_\psi = 0.05$; ~100 epochs. Inference uses DPM-Solver++ (VP) with 16 steps (10/20 in ablations). Deployment uses policy means for determinism; stochastic runs sample once per step.

4.3 Baselines

We evaluate against strong planners under identical simulator settings, inputs, frames, and scenario splits:

Diffusion Planner (DP). Transformer-based diffusion planner with flexible classifier/energy guidance; we keep its fixed schedule (no time-series anisotropy) and DPM-Solver++ inference.

PlanTF. A carefully engineered imitation-learning (IL) planner; we align its horizon/frequency and use public defaults elsewhere.

PLUTO. An IL planner with query-based architecture and auxiliary objectives; adapted to our horizon/frequency.

UrbanDriver. A widely used IL baseline; we port it to nuPlan-mini for complementary comparison.

Fairness controls. We do not tune baselines beyond matching horizon/frequency and frame conventions. All use the same history length and map layers.

4.4 Evaluation Metrics and Protocol

1. Closed-loop composite score.

   The nuPlan composite aggregates scenario-level metrics; we report it as the primary number.

2. Collision rate (at-fault).

   Percentage of scenarios with at-fault collisions per nuPlan rules.

3. Comfort (jerk).

   Mean and 95th percentile of l_1 jerk magnitude over the executed trajectory (recomputed from 10 Hz resamples).

4. Progress.

   Absolute (m) and relative (% of expert) progress along the route centerline.

5. Additional diagnostics.

   Lane-boundary violations, red-light compliance, and lateral oscillation (yaw-rate variance) to contextualize safety/comfort trade-offs.

6. Protocol.

   10 Hz, 8 s horizon; three seeds $\{0, 1, 2\}$; we report mean $\pm$ 95% CI. Inference uses DPM-Solver++ (order-2/3), 16 steps (10/20 in ablations).

7. A4 (single-policy) ablation.

   We implement a joint Gaussian π_ω with a shared trunk and two heads outputting $(s_t, \Delta s_t)$; the covariance is block-diagonal $\mathrm{diag}\left(\Sigma^{(s)}, \Sigma^{(\Delta s)}\right)$. The GRPO loss replaces $\log \pi_\phi + \log \pi_\psi$ by $\log \pi_\omega$, and the KL regularizer uses a block-diagonal reference $\pi_\omega^{\mathrm{ref}} = \mathcal{N}\left([1,0], \mathrm{diag}(\sigma_{\mathrm{ref}}^2, I)\right)$. All other settings remain unchanged.