ENERGY-BASED GENERATIVE PLANNING FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

The central goal of autonomous driving is to combine human-level driving performance with reliable decision-making under multiple safety and behavioral constraints. In this paper, we introduce **EnergyDrive**, an energy-based generative planning framework for end-to-end autonomous driving that supports interpretable planning, flexible sampling, and compositional control. EnergyDrive learns an explicit, scene-conditioned trajectory energy landscape from expert demonstrations, rather than time-dependent transport fields typically learned by diffusion- and flow-based generative planners. This provides a shared representation for trajectory generation and objective composition, with repulsive sampling to explore diverse driving trajectories. At inference time, the learned driving prior and modular energies for target speed, drivable-area compliance, and collision avoidance jointly define the planning objective. EnergyDrive outperforms the compared imitation-learning baselines on the NAVSIM v2 `navhard` and HUGSIM. Qualitative results demonstrate flexible sampling over driving behavior through energy composition. Energy landscape visualizations show the interpretability and reveal how learned preferences and individual constraints shape the resulting plans. Our website is available at https://iclr2027-16699-author.github.io/.

1 INTRODUCTION

Building driving agents that combine human-like adaptability with reliable decision-making has always been a central goal of autonomous driving. End-to-end imitation learning has emerged as a promising approach by learning driving behavior directly from sensor observations and expert demonstrations (Pomerleau, 1988; Chitta et al., 2023; Hu et al., 2023). Generative modeling has further advanced decision-making by learning expressive distributions over actions and trajectories (Janner et al., 2022; Chi et al., 2025). In autonomous driving, diffusion- and flow-based planners capture the multimodality of driving behavior and generate diverse trajectories for the same scene (Zheng et al., 2025; Liu et al., 2026b; Wang et al., 2026b). However, learning to reproduce expert behavior does not ensure that generated plans satisfy the requirements of a new situation. A planner must reconcile safety constraints and driving objectives, such as collision avoidance, traffic rules, comfort, and desired speed, including unfamiliar combinations of these requirements. This motivates generative planners that support flexible search under explicit requirements and make the learned driving behavior easier to interpret.

Recent generative planners incorporate explicit requirements through guidance and refinement. Diffusion Planner (Zheng et al., 2025) uses classifier guidance to shape driving behavior, while GuideFlow (Liu et al., 2026b) combines constrained flow generation with energy-based refinement. Although effective for constraint-aware planning, these methods still encode the driving prior through time-dependent denoising or transport fields, coupling trajectory search to a prescribed generation process at inference time. This coupling makes it less straightforward to steer generation, as flexible sampling and composition with diverse driving constraints rely on guidance or refinement tailored to the underlying dynamics. Moreover, the learned preferences over candidate trajectories remain implicit, limiting the interpretability of the resulting driving behavior.

Energy-based models (EBMs) provide such a representation by expressing a generative model as an energy landscape (Du & Mordatch, 2019). Generation proceeds through search for low-energy solu-

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.

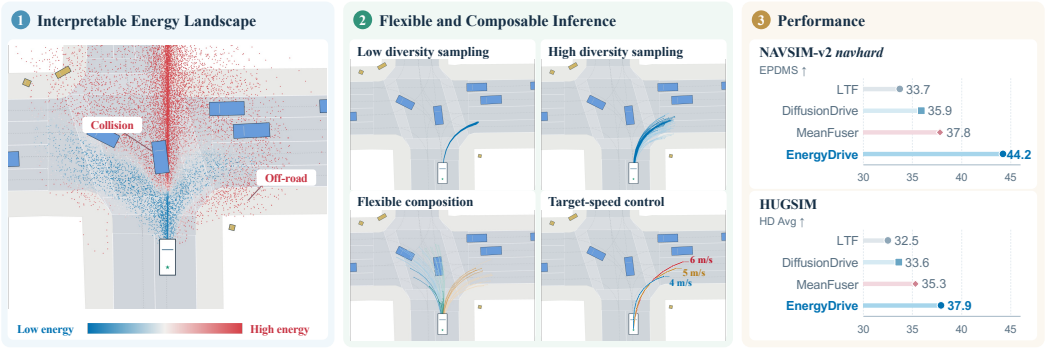

Figure 1: **EnergyDrive overview.** (1) **Interpretable energy landscape.** Trajectory endpoints colored by composed energy values. (2) **Flexible and composable inference.** Repulsion strength controls sample diversity, while objective composition and target-speed control produce different planning behaviors using the same learned model. (3) **Performance.** EnergyDrive outperforms baselines on NAVSIM-v2 navhard and HUGSIM benchmarks.

tions on the EBM's energy landscape, allowing different initializations and optimization strategies. The energy can also be evaluated across candidate solutions to understand the model's preferences, or composed with other energies to incorporate additional requirements. In image generation, these properties have enabled compositional generalization to new combinations of concepts (Liu et al., 2022; Du et al., 2023). More recently, Equilibrium Matching (EqM) has enabled optimization-based generation by learning sampling-time-independent fields, including variants that explicitly parameterize the underlying energy (Wang & Du, 2025). These advances make EBMs a promising foundation for interpretable planning, flexible sampling, and generation under multiple constraints.

In this paper, we introduce EnergyDrive, an energy-based generative framework for end-to-end autonomous driving with a shared representation for trajectory generation, objective composition, and flexible inference. EnergyDrive learns a scene-conditioned trajectory energy independent of sampling time, using human demonstrations and contrastive supervision from negative trajectories (LeCun et al., 2006; Gutmann & Hyvärinen, 2010; Du et al., 2024). Trajectory generation proceeds through gradient-based optimization, with a repulsive sampler that jointly updates multiple trajectories to explore diverse low-energy regions. The learned driving prior and modular energies for target speed, drivable-area compliance, and collision avoidance jointly define the planning objective, and spatial terms use predicted bird's-eye-view semantics. Experiments on NAVSIM v2 (Cao et al., 2025) and HUGSIM demonstrate the resulting planner's effectiveness.

We summarize our contributions as follows: 1) We introduce EnergyDrive, an energy-based generative framework for end-to-end autonomous driving that unifies trajectory generation and constraint composition within an interpretable energy landscape. 2) We develop repulsive sampling and modular energy composition for flexible inference-time planning, together with contrastive learning from negative trajectories. 3) EnergyDrive outperforms the compared imitation-learning baselines on NAVSIM v2 `navhard` and HUGSIM, while qualitative analyses demonstrate sampling flexibility, compositionality, and objective interpretability.

2 Related Work

End-to-End Autonomous Driving. End-to-end learning offers a data-driven alternative to rule-based and modular driving stacks (Bojarski et al., 2016; Codevilla et al., 2018; Wu et al., 2022) where sensor inputs are processed and directly decoded into vehicle trajectories. State-of-the-art trajectory decoders model multimodal trajectory distributions with diffusion and flow matching (Liao et al., 2025; Xing et al., 2025; Jiang et al., 2025b; Yin et al., 2026; Wang et al., 2026b), building on general-purpose generative models (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023) and their use in decision making (Janner et al., 2022; Ajay et al., 2023; Chi et al., 2025). A complementary line of work generates a fixed set of candidate trajectories and selects with learned scorers (Li et al., 2025b; Yao et al., 2026; Li et al., 2025c).

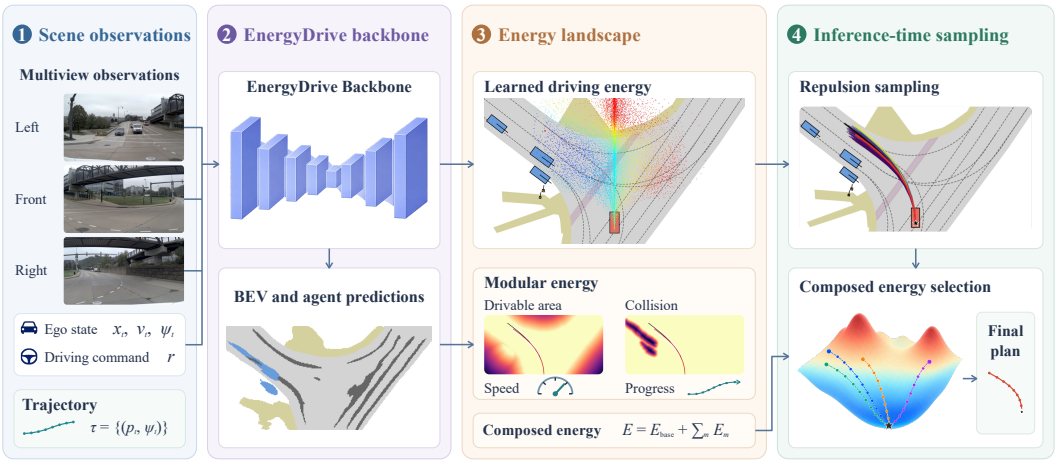

Figure 2: **EnergyDrive framework.** First, EnergyDrive provides learned trajectory energy landscape and auxiliary scene predictions, conditioned by camera observations, ego state, and a driving command. Then, the learned energy landscape is composed with modular energies which encode driving requirements. Repulsive sampling under the learned driving energy generates candidate trajectories; the default planner selects among them using the composed energy.

Driving Objectives at Training and Inference Time. Driving policies must also respect constraints on safety, comfort, and other behavioral preferences. One approach incorporates these during training, for example through hard-negative trajectory harvesting (Bansal et al., 2018; Wang et al., 2026a), or through reinforcement-learning fine-tuning (Li et al., 2025a; Song et al., 2026; Ding et al., 2025). These methods require retraining whenever objectives change, which is often infeasible in practice. Inference-time alternatives steer generative planners toward explicit objectives through diffusion or flow guidance (Dhariwal & Nichol, 2021; Chung et al., 2022; Zheng et al., 2025; Liu et al., 2026b; Yu et al., 2026). For example, Zheng et al. (2025) compose energy functions as denoising guidance, while FlowDrive (Jiang et al., 2025a) conditions a diffusion planner on potential-field-enhanced BEV features. Search-based methods, including Diffusion-ES (Yang et al., 2024) and TOAD (Xu et al., 2026), optimize black-box trajectory objectives at test time.

Energy-Based Models and Decision Making. Energy-based models have been explored for high-dimensional generation and compositional visual synthesis (Du & Mordatch, 2019; Nijkamp et al., 2020; Du et al., 2020). Explicit energies support additive composition, whereas standard diffusion sampling with summed scores need not recover the intended composed distribution (Salimans & Ho, 2021; Liu et al., 2022; Du et al., 2023). Energy-based formulations span reinforcement learning (Haarnoja et al., 2017), policy learning (Florence et al., 2022; Singh et al., 2023; Davies et al., 2025), and model-based planning (Boney et al., 2020), with energy composition explored in reactive control and diffusion-based motion planning (Urain et al., 2023; Luo et al., 2024). Driving applications include learned planning costs, structured motion energies, and energy-based steering (Zeng et al., 2019; 2020; Balesni et al., 2023). More recently, Equilibrium Matching (EqM) (Wang & Du, 2025) enables generation through optimization over a sampling-time-independent energy landscape, while EqM-style and time-unconditional decoders have been explored for robotic control (Zhang et al., 2026; Liu et al., 2026a).

3 EnergyDrive: Energy-Based Generative Planning

In this section, we introduce EnergyDrive, an energy-based framework that unifies trajectory generation, driving-objective composition, and inference-time search. We first learn an observation-conditioned driving energy from expert demonstrations, then compose it with modular objectives for safety and behavior control. Repulsive sampling explores diverse driving alternatives, and composed-energy evaluation selects the final plan. Finally, contrastive learning incorporates model-generated failures to shape the learned energy using both desirable and unsuccessful driving behaviors.

3.1 SCENE-CONDITIONED DRIVING ENERGY

Scene encoding and prediction heads. Let o denote the current camera observations, ego kinematics, and driving command. We represent a future trajectory as $\tau = \{(p_t, \psi_t)\}_{t=1}^{T}$, where $p_t \in \mathbb{R}^2$ and ψ_t are the ego-frame position and heading at the t-th future pose, with consecutive poses separated by time step Δt.

Following TransFuser (Chitta et al., 2023), we construct a camera-based scene encoder with three heads for trajectory-energy modeling, BEV semantic prediction, and agent detection. Our trajectory-energy head is a conditional temporal U-Net (Ronneberger et al., 2015): the global scene query modulates its features through channel-wise affine transformations, and cross-attention to BEV tokens and agent queries supplies spatial and object-level context.

Explicit trajectory energy. Building on explicit-energy Equilibrium Matching (Wang & Du, 2025), we parameterize the driving energy using the squared norm of the U-Net output:

$$E_\theta(\tau; o) = \frac{1}{2}\|f_\theta(\tau; o)\|_2^2, \qquad g_\theta(\tau; o) = -\nabla_\tau E_\theta(\tau; o). \tag{1}$$

Here, f_θ is the network output and g_θ is its induced descent field. Since the energy is independent of sampling time, it defines a scene-conditioned landscape over trajectories, and inference can be formulated as optimization over this landscape.

To learn the driving field from demonstrations, we draw an expert trajectory τ, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and an interpolation coefficient $\gamma \sim \mathcal{U}(0, 1)$. We construct the noised trajectory $\tau_\gamma = \gamma\tau + (1-\gamma)\epsilon$ and optimize

$$\mathcal{L}_{\text{EqM}} = \mathbb{E}_{o,\tau,\epsilon,\gamma}\left\|g_\theta(\tau_\gamma; o) - c(\gamma)(\tau - \epsilon)\right\|_F^2, \tag{2}$$

where $c(\gamma)$ remains constant away from the data endpoint and decreases to 0 as γ approaches 1.

Base energy model training. Gradient matching constrains the descent field but does not directly fix the energy level at demonstrations. We therefore anchor the energy level of the trajectories to an anchor constant b:

$$\mathcal{L}_{\text{anc}} = \mathbb{E}_{o,\tau}\left[\left(E_\theta(\tau; o) - b\right)^2\right]. \tag{3}$$

This term regularizes the explicit energy throughout base training. We also retain BEV semantic cross-entropy and agent classification and box-regression losses, denoted collectively by $\mathcal{L}_{\text{aux}}$. In total, the base learning objective is

$$\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{EqM}} + \beta_{\text{anc}}\mathcal{L}_{\text{anc}} + \beta_{\text{aux}}\mathcal{L}_{\text{aux}}. \tag{4}$$

3.2 MODULAR ENERGY FUNCTIONS AND COMPOSITION

BEV semantic prediction. The auxiliary BEV head predicts a seven-class semantic raster over a $32\,\text{m} \times 64\,\text{m}$ region at $0.25\,\text{m}$ resolution, giving a 128×256 grid. At each grid cell, the seven predicted logits are converted by softmax into class probabilities $P_\ell(q \mid o)$ that sum to one. The drivable-area and collision energies below are constructed from these predicted BEV probabilities at inference time.

Drivable-area energy. To encourage the vehicle to remain within drivable regions, we combine the predicted probabilities of road, centerline, and vehicle classes as $P_{\text{drive}}(q \mid o)$. Let $\{q_j(\tau)\}_{j=1}^{J}$ denote ego-footprint points evaluated along the trajectory. We define

$$E_{\text{DA}}(\tau; o) = \frac{1}{J}\sum_{j=1}^{J}\mathbf{1}\{q_j(\tau) \in \Omega\}\left[1 - P_{\text{drive}}(q_j(\tau) \mid o)\right], \tag{5}$$

where Ω denotes the predicted BEV region. We evaluate four footprint corners at each of 32 interpolated poses, giving $J = 128$ points (Appendix B.1).

Collision avoidance energy. To penalize trajectories that overlap predicted obstacles, we combine the static-object, vehicle, and pedestrian probabilities as $P_{\text{obstacle}}(q \mid o)$. Using the same footprint points, we define the collision avoidance energy $E_{\text{collision}}(\tau; o)$.

Target-speed energy. To control the desired driving speed, we penalize deviations from a target-speed profile. With $p_0 = 0$ and $v_t = \|p_t - p_{t-1}\|_2/\Delta t$, we define

$$E_{\text{speed}}(\tau; o) = \frac{1}{2T}\sum_{t=1}^{T}(v_t - \bar{v}_t)^2, \quad \bar{v}_t = \max\{v^\star, \max(v_{\text{ego}}, v^\star) - (t-1)a_{\text{dec}}\Delta t\}\,. \tag{6}$$

Here $v^\star$ is the requested target speed and a_{dec} is a fixed maximum deceleration used to construct the reference profile. If the current ego speed exceeds $v^\star$, the reference decreases toward the target at this bounded rate.

Progress energy. To balance safety and progress, we use planned travel distance as a progress measure:

$$E_{\text{progress}}(\tau) = \sum_{t=1}^{T}\|p_t - p_{t-1}\|_2. \tag{7}$$

Its signed weight controls the preference for advancing: $w_{\text{prog}} > 0$ favors shorter travel and more cautious plans, while $w_{\text{prog}} < 0$ encourages greater travel and more aggressive progress. Composing this preference provides a controllable safety–progress trade-off.

Together, the learned driving prior and these modules define the planning objective, with explicit road-compliance, obstacle-avoidance, speed, and progress terms whose relative priorities can be adjusted at inference time.

3.3 Flexible Inference with Energy Composition

EnergyDrive's explicit, composable trajectory energy provides a landscape on which inference can use flexible optimization and sampling strategies. This supports different search procedures, diverse parallel exploration, and inference under newly composed driving objectives.

Gradient-based sampling. Starting from Gaussian trajectory noise, we generate a plan by following the learned descent field:

$$\tau^{(k+1)} = \tau^{(k)} + \eta_k g_\theta(\tau^{(k)}; o), \tag{8}$$

where k indexes optimization iterations and η_k is the step size. We can use flexible sampling schedule including fixed stepsize or cosine-decaying schedule.

Repulsive sampling. To obtain a diverse set of feasible proposals while still concentrating samples in low-energy regions, we run K trajectories in parallel and augment the learned descent with a repulsive interaction inspired by Stein variational gradient descent (SVGD) (Liu & Wang, 2016). The learned energy attracts each trajectory toward plausible driving solutions, while the interaction discourages nearby particles from collapsing to the same mode. We denote the repulsion contribution by $\widehat{R}_i^{(k)}$, the parallel update is,

$$\tau_i^{(k+1)} = \tau_i^{(k)} + \eta_k\left[g_\theta(\tau_i^{(k)}; o) + \widehat{R}_i^{(k)}\right]. \tag{9}$$

Details for the repulsive sampling are provided in Appendix B.4.

Energy composition and sampling. EnergyDrive supports multiple forms of energy composition and flexible sampling on the resulting landscapes (Du et al., 2020; 2023). Given a collection of energies $\{E_i(\tau; o)\}_{i\in\mathcal{I}}$, *AND composition* combines their objectives additively, whereas *OR composition* expresses alternatives through a soft minimum:

$$\mathcal{E}_{\text{AND}}(\tau; o) = \sum_{i\in\mathcal{I}} w_i E_i(\tau; o), \quad \mathcal{E}_{\text{OR}}(\tau; o) = -\zeta \log \sum_{i\in\mathcal{I}} \exp[-E_i(\tau; o)/\zeta]\,. \tag{10}$$

Here, w_i controls each objective's contribution, and $\zeta > 0$ controls the softness of OR, approaching a hard minimum as $\zeta \to 0$. The operators can also be nested, allowing alternative maneuvers to be evaluated under shared safety objectives. For any differentiable composed energy $\mathcal{E}_{\text{comp}}$, the corresponding descent field is

$$g_{\text{comp}}(\tau; o) = -\nabla_\tau \mathcal{E}_{\text{comp}}(\tau; o). \tag{11}$$

We use this field for repulsive OR sampling in Figure 3. Appendix B.3 details the procedure.

Composed-energy selection. Unless otherwise stated, our quantitative experiments use repulsive sampling under the learned driving energy alone, together with an independently generated non-repulsive proposal, to form a candidate set $\mathcal{C}(o)$. We then select the candidate with the lowest composed energy,

$$\hat{\tau} = \arg\min_{\tau \in \mathcal{C}(o)} \mathcal{E}_w(\tau; o), \tag{12}$$

where $\mathcal{E}_w$ combines the learned driving energy with the modular terms using the normalizations in Appendix B.2. Thus, composition changes the selected trajectory, not the coordinates of the generated proposals. The same composed objective can select from a supplied trajectory vocabulary without iterative sampling. Appendix B.5 evaluates both modes and demonstrates a speed–performance trade-off.

3.4 Contrastive Learning from Negative Samples

Energy-based models are commonly trained contrastively by assigning lower energy to positive data and higher energy to negative samples (LeCun et al., 2006; Gutmann & Hyvärinen, 2010). We use this principle to incorporate model-generated driving failures as additional supervision.

Mining informative negatives. After training the base model, we apply repulsive sampling to training scenes and evaluate the resulting proposals with collision and drivable-area checks. For each scene, we treat the dataset-provided trajectory as the positive τ^+, and the nearest failing proposal as the negative τ^-, following the negative selection principle of BeyondDrive (Wang et al., 2026a).

Shared-noise contrastive supervision. To compare expert and negative trajectories under matched perturbations, we follow IRED's shared-noise construction (Du et al., 2024). Each pair uses the same Gaussian noise ϵ and interpolation coefficient $\gamma \sim \mathcal{U}(\gamma_{\min}, 1)$:

$$\tau^+_\gamma = \gamma\tau^+ + (1-\gamma)\epsilon, \qquad \tau^-_\gamma = \gamma\tau^- + (1-\gamma)\epsilon. \tag{13}$$

Their difference is therefore $\tau^+_\gamma - \tau^-_\gamma = \gamma(\tau^+ - \tau^-)$, preserving the contrast between the underlying trajectories. We optimize

$$\mathcal{L}_{\text{neg}} = \mathbb{E}\left[\gamma \ \text{softplus}\big(E_\theta(\tau^+_\gamma; o) - E_\theta(\tau^-_\gamma; o)\big)\right]. \tag{14}$$

This objective encourages lower energy for expert trajectories relative to their paired failures, with γ emphasizing comparisons closer to the underlying samples. Combining these terms with the base training objective gives

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \beta_{\text{neg}}\mathcal{L}_{\text{neg}}. \tag{15}$$

In this way, EnergyDrive learns from both expert demonstrations and its own failed proposals: base supervision learns the generative driving field, while contrastive supervision teaches the energy to disfavor unsafe alternatives.

4 Experiments

4.1 Datasets and Benchmarks

Training data. We train our proposed EnergyDrive on NAVSIM `navtrain` demonstrations and SimScale PDM pseudo-expert trajectories (Tian et al., 2026). For contrastive training, we mine failures from a base checkpoint using the protocol in Section 3.4, and train the contrastive-learning model from scratch. Approximately 38% of the training scenes provide a valid positive–negative pair. The base objective is applied to all scenes, and the contrastive term only to paired scenes.

Evaluation. The primary benchmark is the standard NAVSIM-v2 `navhard` (Cao et al., 2025) evaluation, which assesses challenging driving scenarios through a two-stage pseudo-simulation protocol. This evaluates planning in a domain aligned with our NAVSIM-based training setting. We also evaluate on `navtest`, where several metrics are near saturation and overall scores are closely clustered among the compared planners. We therefore focus the main discussion on `navhard` and report `navtest` results in Appendix C.2.

To assess generalization beyond NAVSIM, we additionally evaluate zero-shot transfer on HUGSIM (Zhou et al., 2025), a closed-loop benchmark covering scenes from nuScenes, KITTI-360, PandaSet, and Waymo. We directly deploy the NAVSIM-trained checkpoints of EnergyDrive

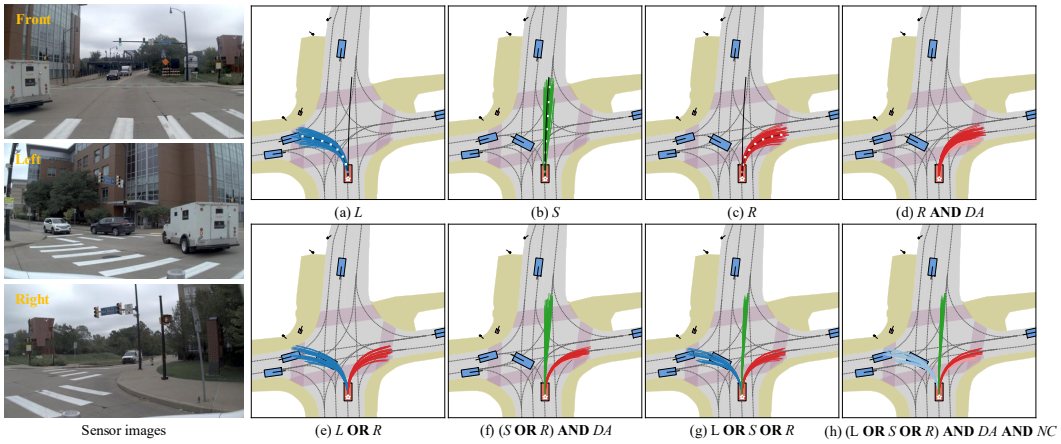

Figure 3: **Energy composition with AND and OR operations.** L, S, and R denote left, straight, and right command conditionings; DA and NC denote drivable-area and collision-avoidance constraints. Panels (a–c) show single-command proposals (black: GT; strong: sample), and (e,g) use repulsive gradient sampling on the level-normalized OR energy. In constraint-annotated panels (d,f,h), lighter trajectories fail the indicated check(s); their coordinates are unchanged.

Table 1: Planning performance on NAVSIM-v2 `navhard`.

Method	Stage	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	**EPDMS↑**
LTF (Chitta et al., 2023)	S1	96.1	85.3	99.4	99.3	84.7	94.7	93.6	97.6	77.3	33.7
	S2	85.5	66.9	91.6	99.1	93.0	81.1	58.3	95.1	42.9	
DiffusionDrive (Liao et al., 2025)	S1	97.9	87.3	99.3	99.3	82.8	96.2	98.0	97.6	63.6	35.9
	S2	87.1	72.1	93.1	98.8	91.9	81.5	61.5	95.5	28.6	
MeanFuser (Wang et al., 2026b)	S1	97.6	91.8	99.3	99.3	84.4	96.7	96.4	97.6	80.4	37.8
	S2	87.2	67.2	94.4	98.3	94.0	79.9	65.9	95.0	42.6	
EnergyDrive (Ours)	S1	97.6	94.9	98.7	100.0	71.8	98.2	93.8	97.8	46.2	**44.2**
	S2	94.7	85.2	94.4	99.3	63.8	93.1	55.2	96.9	48.6	

Evaluation protocol. All methods are evaluated under the standard NAVSIM-v2 `navhard` protocol.

and all baselines. This tests cross-dataset generalization under closed-loop execution, where planning decisions affect subsequent observations. Evaluation details are provided in Appendix C.4.

4.2 Quantitative Results on Benchmarks

Baselines and comparison scope. We compare against LTF, a TransFuser-based regression planner (Chitta et al., 2023); DiffusionDrive, which uses truncated diffusion for multimodal trajectory generation (Liao et al., 2025); and MeanFuser, which uses MeanFlow for one-step multimodal generation and adaptive reconstruction (Wang et al., 2026b). We use the official NAVSIM–SimScale checkpoints for LTF and DiffusionDrive (Tian et al., 2026), and retrain MeanFuser with its official codebase on the same data as EnergyDrive. The comparison excludes separate PDM-score-supervised trajectory scorers. Appendix C.1 details the baselines. Appendix C.5 compares with GuideFlow's published results.

Planning performance. On NAVSIM-v2 navhard (Table 1), EnergyDrive achieves 44.2 EPDMS and outperforms the compared baselines. It achieves the best drivable-area compliance (DAC) in both stages and the best no-at-fault collision (NC) score in Stage 2. The gains are particularly pronounced in Stage 2, where NC and DAC exceed the best baseline on each metric by 7.5 and 13.1 points. These results highlight stronger safety performance under perturbed observations, although accompanied by lower ego progress.

On HUGSIM (Table 2), EnergyDrive achieves the highest aggregate HD score of 37.9, surpasses the compared baselines. It also leads NC and DAC submetrics. These results extend the aggregate performance advantage to closed-loop evaluation without target-domain retraining, supporting the planner's cross-dataset generalization.

Table 2: Planning performance on HUGSIM.

Method	NC↑	DAC↑	TTC↑	COM↑	R_c↑	HD-E↑	HD-M↑	HD-H↑	HD-X↑	**HD Avg↑**
LTF	53.5	96.3	47.0	99.2	46.2	67.2	40.0	18.8	6.9	32.5
DiffusionDrive	55.1	96.8	49.3	98.4	46.3	69.5	44.8	15.4	5.7	33.6
MeanFuser	54.6	96.6	49.0	99.4	48.9	77.5	40.5	17.7	10.9	35.3
EnergyDrive (Ours)	66.9	97.1	63.3	85.2	48.2	72.9	47.4	18.7	13.9	**37.9**

Evaluation protocol. All methods are evaluated under the same HUGSIM protocol with controller convergence to avoid hardware-dependent controller behavior. HD is reported over easy (E), medium (M), hard (H), and extreme (X) scenarios, together with the aggregate HD score.

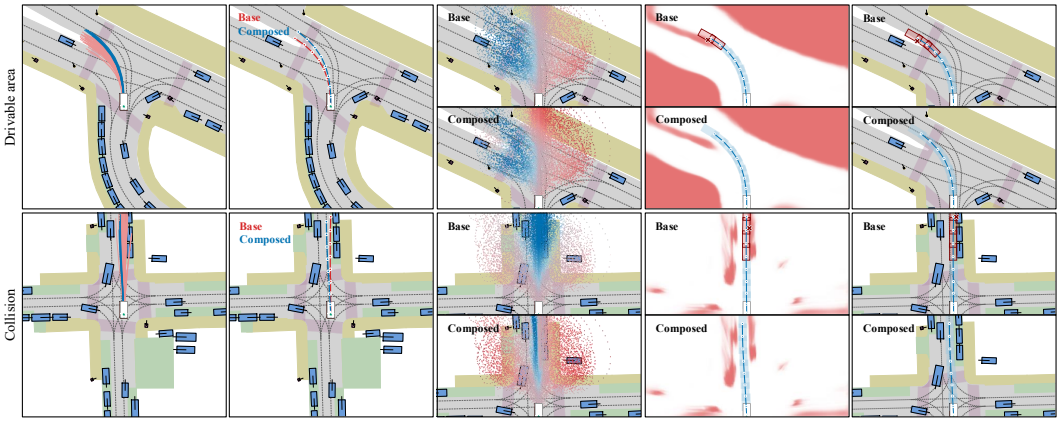

Figure 4: **Trajectory selection with modular energy composition.** Columns show (a) the repulsive proposal pool, (b) candidates selected by the base learned energy (red) and the composed energy (blue), (c) candidate endpoints colored by full-trajectory energy (blue: low energy; red: high energy), (d) selected trajectories over the predicted off-road or obstacle probability layer, and (e) the same trajectories over the scene map.

4.3 Inference-Time Flexibility and Interpretability

Composition on energy landscape. Figure 3 demonstrates composition over the left-, straight-, and right-conditioned energies of a single model. Repulsive sampling follows the gradient of the level-normalized OR energy to explore maneuver alternatives. The constraint-annotated panels identify candidates satisfying shared drivable-area and collision checks without changing their coordinates, illustrating nested requirements such as $(L$ OR S OR $R)$ AND DA AND NC. These examples illustrate how maneuver alternatives and shared safety requirements can be considered together.

Modular energies and interpretability. Figure 4 shows two cases where composition with drivable area and collision energy help the planning performance. Compared with selection using the learned driving energy alone, the composed objective favors candidates that better satisfy the illustrated road and obstacle constraints. In Figure 4 (c), we visualize the energy landscape by a coloring on trajectory endpoints, which shows the interpretability of our method.

Target-speed control. Figure 5 varies the requested speed from 2 to 6 m/s while keeping the scene and trained model fixed. Higher targets produce trajectories that travel farther over the same planning horizon, demonstrating adjustable driving behavior through inference-time composition without retraining. This experiment illustrates behavioral control beyond safety constraints. Target-speed energy is not included in quantitative evaluations, since it is not required by benchmarks.

4.4 Ablation Study

Modular energies and contrastive training. Table 3 examines modular energies on `navhard`. Collision energy alone improves NC but reduces DAC, while drivable-area energy improves DAC with little change in NC. Combining the two achieves 39.3 EPDMS, exceeding either term alone and demonstrating the value of jointly considering complementary constraints. Adding progress energy raises the score to 42.1, with improved Stage 2 NC and DAC, supporting the benefit of adjusting the safety–progress trade-off. Under composed inference, contrastive training provides an additional

Table 3: Ablation study for EnergyDrive.

Method	Components				NAVSIM-v2 `navhard`		
	Contrast.	Coll.	DA	Prog.	NC↑	DAC↑	**EPDMS↑**
Base					96.6 / 85.6	86.7 / 70.3	34.5
+ Collision		✓			98.7 / 88.5	84.9 / 67.4	34.8
+ DA			✓		96.6 / 85.3	92.9 / 75.3	36.7
+ Collision + DA		✓	✓		98.2 / 88.7	91.8 / 73.9	39.3
+ Collision + DA + progress		✓	✓	✓	98.1 / 91.7	93.3 / 80.6	42.1
+ Contrastive + collision + DA	✓	✓	✓		97.8 / 89.4	91.3 / 75.8	40.2
+ Contrastive + all energies	✓	✓	✓	✓	97.6 / 94.7	94.9 / 85.2	**44.2**

Base: model training w/o contrastive learning; Contrast.: contrastive training; Coll.: collision energy; DA: drivable-area energy; Prog.: progress energy. "All energies" refers to collision, drivable-area, and progress energies. Bold marks the best available EPDMS. NC and DAC entries are reported as S1 / S2.

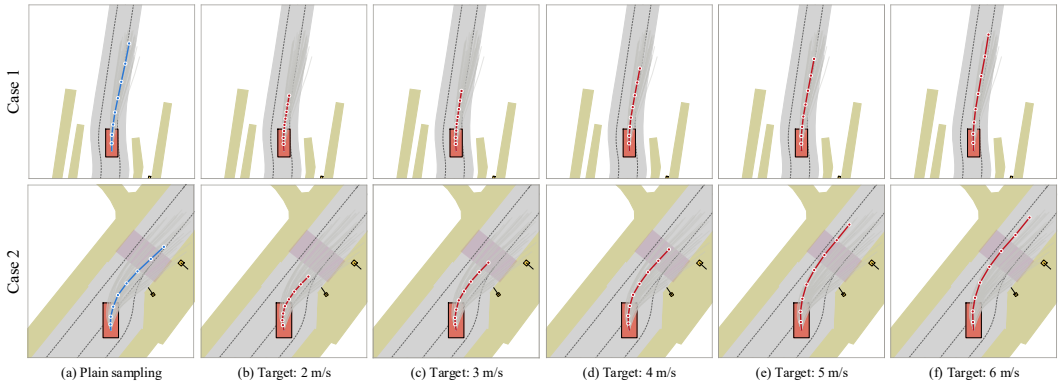

Figure 5: **Inference-time target-speed control.** Each row shows the same scene under plain sampling and composition with different requested speeds. Over the shared planning horizon, higher speed targets produce trajectories that travel farther. Speed labels specify the requested targets. Target-speed energy is not used in the quantitative benchmark evaluations.

0.9 EPDMS without progress energy and 2.1 points with it, bringing the full planner to 44.2. These comparisons support the complementary benefits of negative-sample supervision and inference-time composition.

Hyperparameter sensitivity. Appendix C.3 reports sensitivity to the energy-composition weights, showing a relatively stable range of performance with both safety terms active and a controllable safety–progress trade-off.

5 Conclusion

We have introduced EnergyDrive, an energy-based generative framework that unifies trajectory generation and objective composition through a scene-conditioned energy landscape. The model learns driving behavior from expert demonstrations and incorporates model-generated failures through contrastive supervision. Repulsive sampling explores diverse driving alternatives, while modular energy composition allows safety and behavioral objectives to be combined and adjusted at inference time without retraining. The complete planner leads the compared imitation-learning baselines in NAVSIM v2 `navhard` and HUGSIM benchmarks. Energy visualizations and composition experiments further illustrate how learned preferences and explicit constraints shape the resulting plans. These results support a shared energy representation for effective driving, flexible search, compositional control, and interpretable planning objectives.

A practical limitation is the higher inference latency of iterative sampling on EnergyDrive. The same composed energy can instead rank fixed anchor trajectories in parallel without sampling, offering an efficiency–performance trade-off. Details will be discussed in Appendix B.5.

AI USE STATEMENT

In this work, we used generative AI tools for design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis. We have not used generative AI tools for generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, assist with translation, clean and reformat dataset, and interpret results. And provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, are not applicable to this work. Additionally, we used generative AI tools for create or modify scientific figures or images, summarize or analyse existing literature. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The model architecture, trajectory normalization, training objectives, and negative-mining procedure are described in Section 3 and Appendix A. Appendix B specifies the modular energies, sampling procedure, and benchmark-specific composition weights, and examines vocabulary-based inference. Appendix C documents baseline checkpoints, evaluation protocols, and hyperparameter sensitivity. We use a fixed evaluation seed across methods to support reproducibility.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023. 2

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. 2

Mikita Balesni, Ardi Tampuu, and Tambet Matiisen. Controlling steering with energy-based models. *arXiv preprint arXiv:2301.12264*, 2023. 3

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 3

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 2

Rinu Boney, Juho Kannala, and Alexander Ilin. Regularizing model-based planning with energy-based models. In *Conference on Robot Learning*, pp. 182–191. PMLR, 2020. 3

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning (CoRL)*, 2025. 2, 6, 21, 22

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. 1, 2

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *Pattern Analysis and Machine Intelligence (PAMI)*, 2023. 1, 4, 7, 14, 21, 22

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 3

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 4693–4700. IEEE, 2018. 2

Travis Davies, Yiqi Huang, Alexi Gladstone, Yunxin Liu, Xiang Chen, Heng Ji, Huxian Liu, and Luhui Hu. EBT-Policy: Energy unlocks emergent physical reasoning capabilities. *arXiv preprint arXiv:2510.27545*, 2025. 3

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021. 3

Fan Ding, Xuewen Luo, Hwa Hui Tew, Ruturaj Reddy, Xikun Wang, and Junn Yong Loo. Drive as you like: Strategy-level motion planning based on a multi-head diffusion model. *arXiv preprint arXiv:2508.16947*, 2025. 3

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019. 1, 3

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020. 3, 5

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pp. 8489–8510. PMLR, 2023. 2, 3, 5

Yilun Du, Jiayuan Mao, and Joshua B Tenenbaum. Learning iterative reasoning through energy diffusion. *arXiv preprint arXiv:2406.11179*, 2024. 2, 6, 16

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pp. 158–168. PMLR, 2022. 3

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010. 2, 6

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017. 3

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. 2

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023. 1

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 1, 2

Hao Jiang, Zhipeng Zhang, Yu Gao, Zhigang Sun, Yiru Wang, Yuwen Heng, Shuo Wang, Jinhao Chai, Zhuo Chen, Hao Zhao, Hao Sun, Xi Zhang, Anqing Jiang, and Chuan Hu. FlowDrive: Energy flow field for end-to-end autonomous driving. *arXiv preprint arXiv:2509.14303*, 2025a. 3

Xuefeng Jiang, Yuan Ma, Pengxiang Li, Leimeng Xu, Xin Wen, Kun Zhan, Zhongpu Xia, Peng Jia, Xianpeng Lang, and Sheng Sun. TransDiffuser: Diverse trajectory generation with decorrelated multi-modal representation for end-to-end autonomous driving. *arXiv preprint arXiv:2505.09315*, 2025b. 2

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 2, 6

Derun Li, Jianwei Ren, Yue Wang, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Zhongpu Xia, Peng Jia, Xianpeng Lang, et al. Finetuning generative trajectory model with reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025a. 3

Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M. Alvarez. Hydra-MDP++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025b. 2

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M. Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025c. 2, 21, 24

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047. IEEE, 2025. 2, 7, 21, 22

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. 2

Huanming Liu, Congsheng Xu, Jianmin Ji, and Yao Mu. π_0-EqM: Equilibrium matching for closed-loop vision-language-action control. *arXiv preprint arXiv:2605.23128*, 2026a. 3

Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, JunQiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yadan Luo. Guideflow: Constraint-guided flow matching for planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3719–3728, 2026b. 1, 3, 22, 23

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European conference on computer vision*, pp. 423–439. Springer, 2022. 2, 3

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016. 5, 20

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 2

Yunhao Luo, Chen Sun, Joshua B Tenenbaum, and Yilun Du. Potential based diffusion motion planning. *arXiv preprint arXiv:2407.06169*, 2024. 3

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5272–5280, 2020. 3

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 1

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015. 4

Tim Salimans and Jonathan Ho. Should EBMs model the energy or the score? In *Energy Based Models Workshop, International Conference on Learning Representations*, 2021. 3

Sumeet Singh, Stephen Tu, and Vikas Sindhwani. Revisiting energy based models as policies: Ranking noise contrastive estimation and interpolating energy models. *arXiv preprint arXiv:2309.05803*, 2023. 3

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Diver: Reinforced diffusion breaks imitation bottlenecks in end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. 3

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, et al. Simscale: Learning to drive via real-world simulation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36365–36374, 2026. 6, 7, 15, 21

Julen Urain, Anqi Li, Puze Liu, Carlo D'Eramo, and Jan Peters. Composable energy policies for reactive motion generation and reinforcement learning. *The International Journal of Robotics Research*, 42(10):827–858, 2023. 3

Junli Wang, Zhihua Hua, Xueyi Liu, Zebin Xing, Wei Zhang, Kun Ma, Guang Chen, Hangjun Ye, Long Chen, and Qichao Zhang. Beyond imitation: Learning safe end-to-end autonomous driving from hard negatives. In *European Conference on Computer Vision*, pp. 224–241. Springer, 2026a. 3, 6, 16

Junli Wang, Yinan Zheng, Xueyi Liu, Zebin Xing, Pengfei Li, Kun Ma, Hangjun Ye, Guang Chen, Guang Li, Long Chen, et al. Meanfuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via meanflow for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17884–17893, 2026b. 1, 2, 7, 21, 22

Runqian Wang and Yilun Du. Equilibrium matching: Generative modeling with implicit energy-based models. *arXiv preprint arXiv:2510.02300*, 2025. 2, 3, 4, 16

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022. 2

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611. IEEE, 2025. 2

Yihong Xu, Éloi Zablocki, Yuan Yin, Elias Ramzi, Ellington Kirby, Alexandre Boulch, and Matthieu Cord. Test-time trajectory optimization for autonomous driving. *arXiv preprint arXiv:2606.07170*, 2026. 3

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15342–15353. IEEE, 2024. 3

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026. 2

Liuhan Yin, Runkun Ju, Guodong Guo, and Erkang Cheng. DiffRefiner: Coarse to fine trajectory planning via diffusion refinement with semantic interaction for end to end autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12009–12017, 2026. 2

Hang Yu, Ye Jin, Alessandro Canevaro, Julian Schmidt, Julian Jordan, Peizheng Li, Marc Kaufeld, Silvan Lindner, Johannes Betz, and Wilhelm Stork. G2dp: Diffusion planning with spatio-temporal grid guidance. *arXiv preprint arXiv:2606.26017*, 2026. 3

Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8652–8661. IEEE, 2019. 3

Wenyuan Zeng, Shenlong Wang, Renjie Liao, Yun Chen, Bin Yang, and Raquel Urtasun. Dsdnet: Deep structured self-driving network. In *European conference on computer vision*, pp. 156–172. Springer, 2020. 3

Zunzhe Zhang, Runhan Huang, Yicheng Liu, Shaoting Zhu, Linzhan Mou, and Hang Zhao. Generative control as optimization: Time unconditional flow matching for adaptive and robust robotic control. *arXiv preprint arXiv:2603.17834*, 2026. 3

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. In *International conference on learning representations*, volume 2025, pp. 37207–37227, 2025. 1, 3

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 6, 22

A Model Architecture and Training

A.1 Trajectory Representation and Normalization

EnergyDrive predicts a four-second trajectory with $T = 8$ future poses at $\Delta t = 0.5\,\mathrm{s}$ intervals. Positions are expressed in the current ego rear-axle frame, with x pointing forward and y to the left; headings are relative to the current ego orientation. The current pose is $(p_0, \psi_0) = (0, 0)$ and is not one of the eight predicted poses.

The energy network operates on an affine-normalized trajectory $z = \mathsf{n}(\tau) \in \mathbb{R}^{3\times T}$. For each pose,

$$z_{:,t} = \big((x_t, y_t, \psi_t)^\top - a\big) \oslash s, \qquad a = \begin{bmatrix} 31.8 \\ 2.95 \\ 0.11 \end{bmatrix}, \quad s = \begin{bmatrix} 33.4 \\ 22.6 \\ 1.91 \end{bmatrix}, \tag{A.1}$$

where $\oslash$ denotes element-wise division; positions are in metres and headings in radians before normalization. These constants are fixed across training, negative mining, and inference. We denote the inverse transform by d, so that $\mathsf{d}(z)_t = a + s \odot z_{:,t}$. Gaussian initialization and all sampling updates are performed in normalized coordinates. After sampling, trajectories are converted back to physical coordinates and headings are wrapped using $\mathrm{atan2}(\sin\psi_t, \cos\psi_t)$.

To make the implementation's energy conventions explicit, we distinguish the network-space energy from its evaluation on physical trajectories:

$$\begin{aligned} e_\theta(z; o) &= \frac{1}{2}\|f_\theta(z; o)\|_F^2, \quad E_\theta(\tau; o) = e_\theta(\mathsf{n}(\tau); o), \\ \bar{E}_\theta(\tau; o) &= \frac{E_\theta(\tau; o)}{D}, \qquad\qquad D = 3T = 24. \end{aligned} \tag{A.2}$$

The EqM descent field is $g_\theta^z(z; o) = -\nabla_z e_\theta(z; o)$. The anchor loss, contrastive loss, and default candidate-selection objective use the per-coordinate energy $\bar{E}_\theta$. The sampling field is computed from the unaveraged energy e_θ, so the sampling step sizes below apply to that convention.

A.2 Scene Encoder and Energy Network

Observation processing. We use the current front, front-left, and front-right camera images, without temporal image history. Following the NAVSIM TransFuser (Chitta et al., 2023) preprocessing, all three images are cropped vertically by 28 pixels at each boundary, and each side image is cropped horizontally by 416 pixels at each boundary. The images are concatenated in left–front–right order, resized to 256×1024 pixels, and converted to floating-point RGB values in $[0, 1]$. The ego-status input contains a four-dimensional driving-command encoding, two velocity components, and two acceleration components. Absolute global position is not an input to the energy network.

Table A.1: Training data and coverage of the fixed negative pack. Each covered scene contributes one negative trajectory. Coverage is computed relative to all training scenes, not only the scenes eligible for pairing.

Source	Training scenes	Paired scenes	Coverage
NAVSIM `navtrain`	85,109	33,712	39.61%
SimScale PDM round 0-4	197,088	73,500	37.29%
Total	282,197	107,212	37.99%

Camera-only TransFuser backbone. We adopt the TransFuser backbone with a ResNet-34 image encoder and a learned spatial latent in place of LiDAR measurements. The latent has shape $1 \times 256 \times 256$ and is processed by a second ResNet-34; image and latent features interact through the multiscale fusion transformers. Thus, the retained BEV branch does not require a LiDAR sensor. The image encoder is initialized with the ImageNet-pretrained `resnet34` weights. The remaining components are initialized independently, and the network is trained end to end.

The final 8×8 BEV feature grid is projected to 256 channels. These 64 spatial tokens and one encoded ego-status token form the memory of a three-layer transformer decoder, with eight attention heads and a feed-forward width of 1024. Its 31 learned queries produce one scene query and 30 agent queries. The auxiliary BEV head predicts seven semantic logits per grid cell, while the detection head predicts vehicle presence and five-dimensional boxes (x, y, ψ, l, w) from the agent queries.

Conditional temporal U-Net. The energy head takes $z \in \mathbb{R}^{3\times 8}$ and returns an output of the same shape. Its encoder widths are $(128, 256, 512)$, with temporal resolutions $(8, 4, 2)$, followed by a bottleneck and two upsampling stages. Residual blocks use kernel-size-three temporal convolutions, GroupNorm with eight groups, and Mish activations. A 256-dimensional projection of the scene query provides channel-wise affine conditioning. Cross-attention at each encoder, bottleneck, and decoder resolution uses 128-dimensional features and four heads. Its memory concatenates projected BEV and agent tokens, giving $64 + 30 = 94$ tokens. No sampling-time or noise-level embedding is supplied to the energy network.

A.3 Training Data and Optimization

We train on NAVSIM `navtrain` demonstrations and the five SimScale PDM pseudo-expert rounds, indexed 0–4 (Tian et al., 2026). We do not use the separate recovery-expert rounds. Synthetic logs are included only when their originating real log belongs to the training partition. The resulting datasets are concatenated and shuffled, without additional balancing between real and synthetic samples. Table A.1 lists the sample counts and the scenes with a mined negative. Validation uses 18,179 held-out real samples and does not attach mined negatives.

Both the base model and the full contrastive model are trained for 500 epochs on eight NVIDIA RTX PRO 6000 GPUs. We use distributed data parallelism with 64 samples per GPU, yielding an effective batch size of 512, and Adam with a constant learning rate of 10^{-4}. Training uses BF16 mixed precision and an exponential moving average (EMA) of the model weights with decay 0.9999. We evaluate the final EMA checkpoint. For numerical stability, we apply the gradient norm clip and skip outlier updates when the monitored gradient norm or the loss is above the threshold.

The base model is trained without the contrastive pair term. Its final EMA checkpoint is used to generate the fixed negative dataset for the full model. The full model is then trained independently for 500 epochs with the same initialization recipe. It is not initialized or fine-tuned from the base-model checkpoint.

A.4 Learning Objectives

Equilibrium matching. For a positive trajectory τ^+, let $z^+ = \mathsf{n}(\tau^+)$. We draw $\epsilon \sim \mathcal{N}(0, I)$ and $\gamma \sim \mathcal{U}(\varepsilon, 1-\varepsilon)$, with $\varepsilon = 10^{-3}$, and form $z_\gamma = \gamma z^+ + (1-\gamma)\epsilon$. The implemented EqM

objective (Wang & Du, 2025) averages the squared error over trajectory coordinates:

$$\mathcal{L}_{\mathrm{EqM}} = \mathbb{E}\left[\frac{1}{D}\left\|g_\theta^z(z_\gamma; o) - c(\gamma)(z^+ - \epsilon)\right\|_F^2\right], \qquad c(\gamma) = \min\left\{1, \frac{1-\gamma}{1-0.8}\right\}. \tag{A.3}$$

The field is differentiated through during training, retaining the computation graph for optimization of the network parameters. The interpolation coefficient controls the target but is not an input to f_θ.

Clean-positive energy anchor. We regularize the per-coordinate energy at the unperturbed positive trajectory:

$$\mathcal{L}_{\mathrm{anc}} = \mathbb{E}_{o,\tau^+}\left[\left(\bar{E}_\theta(\tau^+; o) - 1\right)^2\right]. \tag{A.4}$$

This term applies to every training sample, independently of whether it has a mined negative. It is evaluated on clean trajectories, rather than their noised interpolants, and has its own coefficient independent of the contrastive weight.

Shared-noise contrastive loss. For each paired scene, normalize the positive and negative trajectories as z^+ and z^-. Following the shared-noise construction of IRED (Du et al., 2024), draw one $\epsilon \sim \mathcal{N}(0, I)$ and one $\gamma \sim \mathcal{U}(0.5, 1-\varepsilon)$ for the pair:

$$z_\gamma^+ = \gamma z^+ + (1-\gamma)\epsilon, \qquad z_\gamma^- = \gamma z^- + (1-\gamma)\epsilon. \tag{A.5}$$

The pair objective is

$$\mathcal{L}_{\mathrm{neg}} = \mathbb{E}_{\mathrm{paired}}\left[\gamma\ \mathrm{softplus}\left(\frac{e_\theta(z_\gamma^+; o)/D - e_\theta(z_\gamma^-; o)/D}{\vartheta}\right)\right], \qquad \vartheta = 1. \tag{A.6}$$

Gradients propagate through both positive and negative energies. We use the softplus loss, which is averaged over paired samples in each minibatch, and a minibatch without valid pairs contributes zero contrastive loss. The noise and interpolation coefficient are resampled during training even though the underlying negative trajectories remain fixed.

Auxiliary supervision and total loss. The BEV semantic loss $\mathcal{L}_{\mathrm{BEV}}$ is pixel-wise cross-entropy. For agent detection, Hungarian matching uses a weighted combination of presence-classification and positional ℓ_1 costs. The matched classification loss $\mathcal{L}_{\mathrm{cls}}$ is binary cross-entropy, and $\mathcal{L}_{\mathrm{box}}$ sums absolute errors in the five box coordinates over valid matched vehicles, normalized by the number of valid targets. In the camera-only configuration, detection supervision retains vehicles within the forward $\pm 80^\circ$ angular range. The full training objective is

$$\mathcal{L} = 10\mathcal{L}_{\mathrm{EqM}} + \mathcal{L}_{\mathrm{anc}} + 0.5\mathcal{L}_{\mathrm{neg}} + 10\mathcal{L}_{\mathrm{BEV}} + 10\mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{box}}. \tag{A.7}$$

The base model uses the same objective with the contrastive term removed. All terms are enabled from the beginning of the corresponding training run.

A.5 Offline Negative-Sample Mining

We mine negatives using the final EMA checkpoint of the base model on the training scenes in Table A.1. For each scene, we generate one ordinary proposal and a pool of $K = 64$ repulsive proposals. Mining uses the ten-step sampler described in Appendix B.4, with initial repulsion strength 2.0 rather than the default inference value of 1.5. No modular energy is added during proposal generation.

Each proposal is evaluated with the on-road and no-collision checks, similar to the simulator's definition. A hard-failure candidate violates at least one of these checks. Pairing requires a positive target that passes both checks and at least one failing proposal. We retain the failing proposal nearest to the target, following the hard-negative selection principle of BeyondDrive (Wang et al., 2026a). Specifically, the implemented distance is

$$d_{\mathrm{pose}}(\tau, \tau^+) = \frac{1}{T}\sum_{t=1}^{T}\left[(x_t - x_t^+)^2 + (y_t - y_t^+)^2 + (\psi_t - \psi_t^+)^2\right]. \tag{A.8}$$

This distance uses stored poses in physical coordinates and direct heading differences, without angular wrapping or per-channel normalization of the differences. The ordinary proposal is also eligible for selection. Thus,

$$\tau^- = \arg\min_{\tau \in \mathcal{H}(o)} d_{\mathrm{pose}}(\tau, \tau^+), \tag{A.9}$$

where $\mathcal{H}(o)$ is the set of hard-failure proposals for the scene.

The resulting pack contains 107,212 negatives, one for each covered scene, and is held fixed throughout full-model training. These include 32,506 collision failures and 75,073 off-road failures; the categories overlap for 367 trajectories. Scenes without a paired negative still contribute the EqM, anchor, and auxiliary losses. The simulator checks are used for offline mining; the deployed planner instead evaluates its learned energy and the predicted-map energies described below.

B Inference-Time Search and Energy Composition

B.1 BEV Semantics and Modular Energies

Predicted semantic layers. The BEV head predicts a 128×256 grid at $0.25\,\mathrm{m}$ resolution, covering 32 metres forward and 64 metres laterally. Its seven mutually exclusive labels are background, road, walkway, centerline, static object, vehicle, and pedestrian. Training targets rasterize lane and intersection polygons, walkway polygons, lane and lane-connector centerlines, and annotated object boxes in the ego frame. Later semantic labels overwrite earlier labels at shared pixels. At inference time, all spatial energies use softmax probabilities predicted from the camera observations, not these ground-truth rasters.

The drivable probability sums road, centerline, and vehicle probabilities, accounting for road pixels overwritten by centerline and vehicle labels in the semantic target. The obstacle probability sums static-object, vehicle, and pedestrian probabilities. The vehicle class therefore contributes to both the road-support union and the obstacle union: treating a vehicle-covered pixel as underlying road support does not remove its collision penalty.

Temporal interpolation and vehicle footprint. We prepend the current pose to the trajectory and divide each interval into $r = 4$ equal segments. For $t = 1, \dots, T$ and $j = 1, \dots, r$, the interpolated pose is

$$(\widetilde{p}_{t,j}, \widetilde{\psi}_{t,j}) = \left(1 - \frac{j}{r}\right)(p_{t-1}, \psi_{t-1}) + \frac{j}{r}(p_t, \psi_t). \tag{B.1}$$

This gives $M = rT = 32$ poses at $0.125\,\mathrm{s}$ spacing, including the original future poses. At each interpolated pose, we evaluate four corners of the vehicle footprint. The rear-axle-relative offsets, in metres, are

$$\begin{aligned} b_1 &= (4.049, 1.1485), \qquad b_2 = (4.049, -1.1485), \\ b_3 &= (-1.127, 1.1485), \quad b_4 = (-1.127, -1.1485). \end{aligned} \tag{B.2}$$

For an interpolated pose indexed by m, the corner location is $q_{m,j} = \widetilde{p}_m + R(\widetilde{\psi}_m) b_j$, where R is the planar rotation matrix. The spatial energies average over all $4M = 128$ corner locations:

$$E_{\mathrm{DA}}(\tau; o) = \frac{1}{4M} \sum_{m=1}^{M} \sum_{j=1}^{4} \mathbf{1}\{q_{m,j} \in \Omega\} \big[1 - P_{\mathrm{drive}}(q_{m,j} \mid o)\big], \tag{B.3}$$

$$E_{\mathrm{collision}}(\tau; o) = \frac{1}{4M} \sum_{m=1}^{M} \sum_{j=1}^{4} \mathbf{1}\{q_{m,j} \in \Omega\} P_{\mathrm{obstacle}}(q_{m,j} \mid o). \tag{B.4}$$

Here Ω is the predicted raster region. Out-of-raster points contribute zero, and the denominator remains $4M$.

Raster sampling. Probabilities at footprint locations are obtained by bilinear interpolation. For a physical point (x, y), the normalized raster coordinates are $u = y/32$ and $v = x/16 - 1$. We use border padding with `align_corners=False`, followed by the explicit mask $0 \le x \le 32$ and $|y| \le 32$. The default energies use occupancy probabilities directly, without thresholding, distance transforms, or temporal discounting of dynamic-object probabilities. The predicted map is fixed over the trajectory horizon for a planning query and recomputed from the observation at the next query.

Speed and progress. The target-speed energy uses the origin-prepended displacement speeds $v_t = \|p_t - p_{t-1}\|_2 / \Delta t$ and the deceleration-limited reference

$$\bar{v}_t = \max\{v^\star, \max(v_{\mathrm{ego}}, v^\star) - (t-1) a_{\mathrm{dec}} \Delta t\}, \qquad a_{\mathrm{dec}} = 3\,\mathrm{m/s^2}. \tag{B.5}$$

We use the symmetric quadratic energy $E_{\text{speed}} = (2T)^{-1}\sum_t (v_t - \bar{v}_t)^2$. The progress term is the cumulative positional arc length $E_{\text{progress}} = \sum_t \|p_t - p_{t-1}\|_2$, including the segment from the origin to the first waypoint. Target-speed energy is used for controllable-generation examples and is not included in the default benchmark objective.

B.2 Default Planner: Proposal Generation and Selection

The default planner separates proposal generation from composed-energy selection. It generates one ordinary trajectory and $K = 64$ interacting trajectories under the learned driving energy, yielding a set $\mathcal{C}(o)$ of 65 candidates. Modular energies do not alter their coordinates during proposal generation. Each candidate is then scored by

$$\begin{aligned}\mathcal{E}_w(\tau; o) = \bar{E}_\theta(\tau; o) + w_{\text{DA}} E_{\text{DA}}(\tau; o) + w_{\text{col}} E_{\text{collision}}(\tau; o) \\ + w_{\text{prog}} \frac{E_{\text{progress}}(\tau)}{X}, \qquad X = 32\,\text{m},\end{aligned} \tag{B.6}$$

and the planner returns $\widehat{\tau} = \arg\min_{\tau \in \mathcal{C}(o)} \mathcal{E}_w(\tau; o)$. The learned energy is averaged over its $D = 24$ coordinates, while travel distance is normalized by the forward extent of the BEV raster. These fixed normalizers make the weighting convention explicit. The default NAVSIM weights are $(w_{\text{DA}}, w_{\text{col}}, w_{\text{prog}}) = (1, 1, 0.1)$. The positive progress coefficient expresses a preference for shorter travel when balancing the learned driving energy against safety terms.

Selection uses the complete pool without hard occupancy gates, a fixed trajectory vocabulary, or a subsequent coordinate-refinement stage. The ordinary proposal participates in the same comparison as the repulsive proposals. Composition therefore changes which candidate is returned, rather than modifying candidate coordinates. The separate vocabulary-based variant is described in Appendix B.5.

B.3 Command Alternatives and Gradient-Based Composition

The qualitative OR examples use gradients of a composed learned energy, whereas the default quantitative planner uses learned-energy proposals followed by composed-energy selection (Appendix B.2). We distinguish these inference procedures below.

Level-normalized OR composition. For an allowed command set $\mathcal{A} \subseteq \{\text{L}, \text{S}, \text{R}\}$, let $o^{(c)}$ replace only the driving command in the observation and define $E_c(\tau; o) = E_\theta(\tau; o^{(c)})$. The same network is evaluated under each command. Since command-conditioned energies may have different baseline levels, we subtract a scene-specific reference level before OR composition:

$$\widetilde{E}_c(\tau; o) = E_c(\tau; o) - \ell_c(o), \qquad \ell_c(o) = E_c(\tau_c^{\text{ref}}; o), \tag{B.7}$$

where τ_c^{ref} is an ordinary proposal under command c. The reference levels are held fixed throughout the subsequent search. This is an alignment of branch energy levels, distinct from the per-coordinate averaging in Eq. A.2.

The OR energy is the soft minimum of the aligned branch energies:

$$E_\vee(\tau; o, \mathcal{A}) = -\zeta \log \sum_{c \in \mathcal{A}} \exp\left[-\frac{\widetilde{E}_c(\tau; o)}{\zeta}\right], \qquad \zeta > 0. \tag{B.8}$$

Its normalized-coordinate descent field is

$$\begin{aligned} g_\vee^z(z; o) &= \sum_{c \in \mathcal{A}} \alpha_c(\mathsf{d}(z); o)\, g_\theta^z(z; o^{(c)}), \\ \alpha_c(\tau; o) &= \frac{\exp[-\widetilde{E}_c(\tau; o)/\zeta]}{\sum_{d \in \mathcal{A}} \exp[-\widetilde{E}_d(\tau; o)/\zeta]}. \end{aligned} \tag{B.9}$$

The weights α_c vary with the trajectory, allowing different particles to follow different command-conditioned branches. Shared requirements can be added to this alternative-maneuver energy, for example $E_\vee + w_{\text{DA}} E_{\text{DA}} + w_{\text{col}} E_{\text{collision}}$. Neither operation requires retraining the network.

Sampling used in the OR examples. In Figure 3, particles are updated using the composed field $g^z_\vee$ in Eq. B.9, together with repulsion. OR composition therefore acts on trajectory coordinates during generation, rather than only selecting among independently generated command-specific proposals. The fixed reference levels affect the branch weights α_c, but do not change the individual branch gradients. In the constraint-annotated panels (d,f,h), shared drivable-area and collision checks mark the generated candidates without changing their coordinates; these annotations are distinct from gradient-based optimization of the safety terms.

Direct optimization of composed energies. For any differentiable composed objective, define $F(z; o) = E_{\text{comp}}(\mathsf{d}(z); o)$. Direct sampling or refinement uses

$$z^{(k+1)} = z^{(k)} - \eta_k \nabla_z F(z^{(k)}; o). \tag{B.10}$$

Gaussian initialization supports trajectory generation, while initialization from a normalized existing plan supports refinement. The affine transform must be included in the gradient: for a modular energy defined in physical coordinates, the corresponding gradient in normalized coordinates is multiplied channel-wise by s. This coordinate-changing optimization is distinct from the default selection procedure in Appendix B.2.

Why the default planner uses modular-energy selection. The learned driving field is trained by EqM on perturbed trajectories, whereas the BEV-derived safety terms evaluate footprint locations on a predicted probability raster (Appendix B.1). These provide complementary information, but need not have equally useful gradients for trajectory generation. Under bilinear raster sampling, spatial derivatives depend on local probability differences: a locally uniform patch can provide a nonzero penalty but little directional information. Thus, differentiability alone does not ensure a useful correction direction for a coherent trajectory. The default planner uses repulsion to explore the learned driving landscape and compares the resulting candidates using composed energy values, without requiring the spatial modules to provide a generative field. Only this selection step is value-based; proposal generation still differentiates the learned energy. In contrast, OR composition combines learned generative energies and uses their composed gradients directly, illustrating a complementary inference procedure within the same representation.

B.4 Repulsive Sampling

We implement the interacting sampler in normalized trajectory coordinates. All quantities below are recomputed at each sampling iteration for the K particles of a single scene; particles from different scenes do not interact. Let $z_i \in \mathbb{R}^{3\times T}$ and $G_i = g^z_\theta(z_i; o)$ denote particle i and its learned descent field.

Low-frequency descriptor and kernel interaction. Define a unit-norm temporal ramp $u_t = t/(\sum_{q=1}^{T} q^2)^{1/2}$ and project the position channels onto it:

$$\phi_i = z_{i,xy} u \in \mathbb{R}^2, \qquad v_i = G_{i,xy} u \in \mathbb{R}^2. \tag{B.11}$$

This descriptor summarizes a smooth positional component across the horizon. Pairwise interactions use an RBF kernel with a scene-adaptive bandwidth:

$$\begin{aligned} h^2 &= \frac{\text{median}_{i<j} \|\phi_i - \phi_j\|_2^2}{\log K}, \qquad \kappa_{ij} = \exp\left(-\frac{\|\phi_i - \phi_j\|_2^2}{2h^2}\right), \\ a_i &= \frac{1}{K-1} \sum_{j\neq i} \kappa_{ij}(\phi_i - \phi_j). \end{aligned} \tag{B.12}$$

A positive numerical floor is applied to h^2. Self-interactions are excluded. With a single particle, the interaction is zero.

Projection and lifting. We remove the component of a_i along the learned field's descriptor:

$$a_i^\perp = \begin{cases} a_i - \dfrac{\langle a_i, v_i\rangle}{\|v_i\|_2^2} v_i, & \|v_i\|_2 > 0, \\ a_i, & \|v_i\|_2 = 0. \end{cases} \qquad R_i^{\text{raw}} = \begin{bmatrix} a_i^\perp u^\top \\ 0_{1\times T} \end{bmatrix}. \tag{B.13}$$

The lifted interaction is smooth over time and has no heading component. Moreover, $\langle G_i, R_i^{\text{raw}}\rangle_F = \langle v_i, a_i^\perp\rangle = 0$, so it is orthogonal to the full learned field in normalized trajectory space.

Algorithm B.1 EnergyDrive default inference

Require: Observation o; trained energy; $K = 64$; $N = 10$; weights w
Ensure: One four-second trajectory $\widehat{\tau}$
1: Obtain scene conditioning and predicted BEV probabilities from o.
2: Initialize $z_0, z_1, \ldots, z_K$ independently from $\mathcal{N}(0, I)$.
3: **for** $k = 0, \ldots, N-1$ **do**
4: Update the ordinary proposal: $z_0 \leftarrow z_0 + \eta_k g_\theta^z(z_0; o)$.
5: Evaluate $G_i = g_\theta^z(z_i; o)$ for $i = 1, \ldots, K$.
6: Compute $\widehat{R}_i^{(k)}$ using Eqs. equation B.11–equation B.15.
7: Update the pool jointly: $z_i \leftarrow z_i + \eta_k(G_i + \widehat{R}_i^{(k)})$.
8: **end for**
9: Form $\mathcal{C}(o)$ by converting particles to physical trajectories and wrapping headings.
10: Score every $\tau \in \mathcal{C}(o)$ with Eq. B.6.
11: **return** $\widehat{\tau} = \arg\min_{\tau \in \mathcal{C}(o)} \mathcal{E}_w(\tau; o)$.

Table B.1: **Model size and inference efficiency.** EPDMS is measured on NAVSIM-v2 `navhard`. The two full-model inference variants share the final contrastive checkpoint. The base/plain row uses the separately trained non-contrastive model and is not a matched-checkpoint performance ablation.

Method	Inference setting	Params. (M)	Time (ms)↓	FPS↑	EPDMS↑
LTF	Default	56.036	10.09	99.16	33.7
DiffusionDrive	Default	60.781	17.05	58.66	35.9
MeanFuser	Default	55.581	11.33	88.29	37.8
EnergyDrive (base)	Plain	68.377	68.30	14.64	34.5
EnergyDrive (full)	Repulsion, $K = 64$	68.377	92.67	10.79	**44.2**
EnergyDrive (full)	Vocabulary, 2,048	68.377	33.54	29.82	41.3

Scene-level calibration and per-particle cap. A shared scalar matches the median interaction norm to the median learned-field norm:

$$q = \frac{\operatorname{median}_i \|G_i\|_F}{\operatorname{median}_i \|R_i^{\text{raw}}\|_F}, \qquad R_i = qR_i^{\text{raw}}. \tag{B.14}$$

When the denominator is zero, we set $q = 0$. Using one scalar per scene preserves relative interaction magnitudes across its particles. We then cap the weighted interaction separately for each particle:

$$\widehat{R}_i^{(k)} = \begin{cases} \min\left\{\lambda_k, \dfrac{\|G_i\|_F}{\|R_i\|_F}\right\} R_i, & \|G_i\|_F > 0 \text{ and } \|R_i\|_F > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{B.15}$$

Consequently, $\|\widehat{R}_i^{(k)}\|_F \le \|G_i\|_F$, while the learned field itself is not attenuated. Together with orthogonality, this gives $\|G_i + \widehat{R}_i^{(k)}\|_F \le \sqrt{2}\|G_i\|_F$. The interaction is inspired by SVGD (Liu & Wang, 2016), but does not replace the learned field with a kernel average of other particles' fields.

Update schedule. Particles start from independent standard Gaussian noise and are updated by

$$z_i^{(k+1)} = z_i^{(k)} + \eta_k\big(G_i + \widehat{R}_i^{(k)}\big), \qquad k = 0, \ldots, N-1, \tag{B.16}$$

with $N = 10$ and cosine schedules

$$\eta_k = 0.05 + \frac{0.4 - 0.05}{2}\left[1 + \cos\left(\frac{\pi k}{N-1}\right)\right], \qquad \lambda_k = \frac{\lambda_0}{2}\left[1 + \cos\left(\frac{\pi k}{N-1}\right)\right]. \tag{B.17}$$

We use $\lambda_0 = 1.5$ for default inference and $\lambda_0 = 2.0$ for negative mining. No Langevin noise or momentum is added. Repulsion is zero at the last update. The independent ordinary proposal follows the same learned-field and step-size schedule without any repulsive interaction. Algorithm B.1 summarizes the default planner.

B.5 Sampling Efficiency

Runtime and model size. Table B.1 compares model size, inference time, and planning performance. Iterative generation requires differentiating the learned energy with respect to trajectory

Table C.1: **Baseline checkpoint selection.** We evaluate both SimScale checkpoint variants on every benchmark. Bold identifies the selected aggregate score within each baseline. For a benchmark, all reported submetrics use that selected checkpoint. NAVSIM columns report EPDMS; HUGSIM reports average HD.

Method	Checkpoint	navhard↑	navtest↑	HUGSIM↑
LTF	navtest	31.1	**87.8**	**32.5**
LTF	navhard	**33.7**	87.6	30.0
DiffusionDrive	navtest	34.9	**89.4**	**33.6**
DiffusionDrive	navhard	**35.9**	86.2	26.6
EnergyDrive	Same final EMA	44.2	89.1	37.9

coordinates, and EnergyDrive has higher inference latency than the compared baselines. Nevertheless, on a single NVIDIA RTX PRO 6000 GPU, our default ten-step sampler with 64 repulsive proposals and one ordinary proposal takes 92.67 ms, corresponding to 10.79 FPS. This is below a 500 ms planning budget at 2 Hz; NAVSIM-v2 itself evaluates fixed-horizon trajectories rather than imposing this wall-clock replanning deadline (Cao et al., 2025).

Vocabulary-based inference. As discussed in Section 3.3, an explicit energy also supports inference without iterative generation. We use a fixed, downsampled set $\mathcal{V}$ of 2,048 trajectories from the GTRS-Dense vocabulary (Li et al., 2025c) and select

$$\hat{\tau}_{\mathcal{V}} = \arg\min_{\tau \in \mathcal{V}} \mathcal{E}_w(\tau; o). \tag{B.18}$$

Only the trajectory candidates are reused: scoring uses EnergyDrive's learned and modular energies, not the GTRS-Dense scoring network. This removes the iterative trajectory-gradient updates. With the same final contrastive checkpoint and composed objective, this variant takes 33.54 ms (29.82 FPS) and achieves 41.3 EPDMS on navhard. It is approximately 2.76× faster than the default sampler, with a 2.9-point EPDMS reduction, while remaining 3.5 points above the strongest compared baseline. Thus, one trained energy model supports both more flexible continuous search and faster selection over a fixed candidate set.

C Evaluation Protocols and Discussion

C.1 Details for Baselines

LTF and DiffusionDrive. LTF is a TransFuser-based regression planner (Chitta et al., 2023), whereas DiffusionDrive uses truncated diffusion for trajectory generation (Liao et al., 2025). For both methods, we use the official NAVSIM–SimScale checkpoints released by SimScale (Tian et al., 2026). The release provides navtest and navhard checkpoints for each method. We evaluate both checkpoints on navhard, navtest, and HUGSIM, and report the checkpoint with the higher aggregate score on each benchmark: EPDMS for NAVSIM and average HD for HUGSIM. All component metrics for that benchmark come from the same selected checkpoint; metrics are not mixed across checkpoints.

Table C.1 gives our re-evaluations. The navhard checkpoint is selected for navhard, while the navtest checkpoint is selected for navtest and HUGSIM. This benchmark-wise selection favors the baselines relative to requiring one checkpoint across all three evaluations. In contrast, EnergyDrive uses the exact same final EMA checkpoint of the contrastively trained model for all three benchmarks; only the explicitly stated inference weights differ (Appendix B.2).

MeanFuser. MeanFuser uses MeanFlow for one-step multimodal generation and adaptive reconstruction (Wang et al., 2026b). Since the official checkpoint is not trained on the NAVSIM–SimScale combination used here, we train MeanFuser with its official codebase and configurations on the exact same training data as EnergyDrive. We evaluate the resulting model under our benchmark protocols.

Comparison scope. We compare the evaluated imitation-learning planners without a separate PDM-score-supervised trajectory scorer. This scope permits imitation from pseudo-expert trajectories and binary safety checks for negative mining, but excludes a separate network trained to predict PDM

Table C.2: Planning performance on NAVSIM-v2 `navtest`.

Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	**EPDMS↑**
LTF (Chitta et al., 2023)	98.3	95.6	99.6	99.8	87.1	97.5	97.2	98.3	88.2	87.8
DiffusionDrive (Liao et al., 2025)	98.5	97.1	99.6	99.8	87.4	97.8	98.0	98.3	87.5	89.4
MeanFuser (Wang et al., 2026b)	98.1	97.6	99.7	99.8	87.6	97.3	97.9	98.3	88.0	**89.7**
EnergyDrive (Ours)	97.8	98.3	99.4	99.8	86.6	97.2	96.5	98.3	86.9	89.1

scores. Accordingly, PDM-score-supervised scoring systems and scorer-augmented variants, such as GuideFlow+Scorer (Liu et al., 2026b) with its external GTRS-Dense scorer, are outside the main comparison. The reported advantage therefore concerns the compared imitation-learning baselines, not all published planning systems. Reusing a trajectory vocabulary in Appendix B.5 does not introduce an external scoring network.

C.2 NAVSIM Evaluation

We evaluate the final EMA checkpoints on NAVSIM-v2 `navtest` and `navhard` (Cao et al., 2025). The former supplies recorded observations, while the latter uses the two-stage pseudo-simulation protocol with pre-rendered perturbed observations. We use the benchmark's simulator, scoring functions, and stage-aggregation procedure. The overall two-stage EPDMS is not an arithmetic average of the separately reported stage scores. At each query, the planner receives the observation and ego status, generates proposals, and selects a trajectory without accessing simulator scores or ground-truth BEV labels. Such privileged information is used for evaluation and offline negative mining, not candidate selection.

C.3 Ablation Results for Hyperparameter Sweep

Checkpoint and reference setting. All sweeps in Figure C.1 use the final EMA checkpoint of the base model after 500 training epochs without contrastive learning. They use repulsive proposal generation followed by energy-based selection. The zero-weight setting retains selection by the learned energy over this pool, rather than the single-proposal inference used for the "Base" row in Table 3.

Safety-weight sensitivity. The left panel fixes $w_{\text{prog}} = 0$ and sweeps the drivable-area and collision weights over $\{0, 0.25, 0.5, 1, 2, 4\}$. Across the tested combinations with both weights positive, EPDMS ranges from 37.6 to 39.3, compared with 33.4 without either term. This indicates that including both safety objectives matters more than precise weight selection within this tested range. We use the fixed choice $w_{\text{DA}} = w_{\text{col}} = 1$ for all three quantitative benchmarks.

Progress as an inference-time preference. The progress term is optional: we use $w_{\text{prog}} = 0$ on `navtest` and HUGSIM. On `navhard`, where perturbed observations test recovery from challenging ego states, we use $w_{\text{prog}} = 0.1$ to illustrate a more cautious safety–progress balance. The right panel characterizes this safety–progress trade-off on the base model: increasing the progress weight initially improves NC and DAC while reducing EP, and EPDMS is highest at a moderate positive weight rather than the largest tested weight.

Performance without progress regularization. With $(w_{\text{DA}}, w_{\text{col}}, w_{\text{prog}}) = (1, 1, 0)$, the base-model heatmap achieves 39.3 EPDMS, above all compared baselines, indicating that the aggregate improvement does not depend on the added shorter-travel preference. A positive progress weight does favor shorter travel, but selection still jointly considers the learned driving energy and scene-conditioned safety penalties rather than choosing the shortest candidate. The progress term therefore provides an additional operating-point adjustment, not the sole source of the planner's improvement.

C.4 HUGSIM Evaluation

HUGSIM evaluates planning in a rendered closed-loop environment (Zhou et al., 2025). We use its 436 released scenarios: 88 from nuScenes, 108 from Waymo, 113 from KITTI-360, and 127 from PandaSet. EnergyDrive consumes the rendered front-facing camera views and simulator-provided ego state, with the route command mapped to the model's command encoding. It returns eight

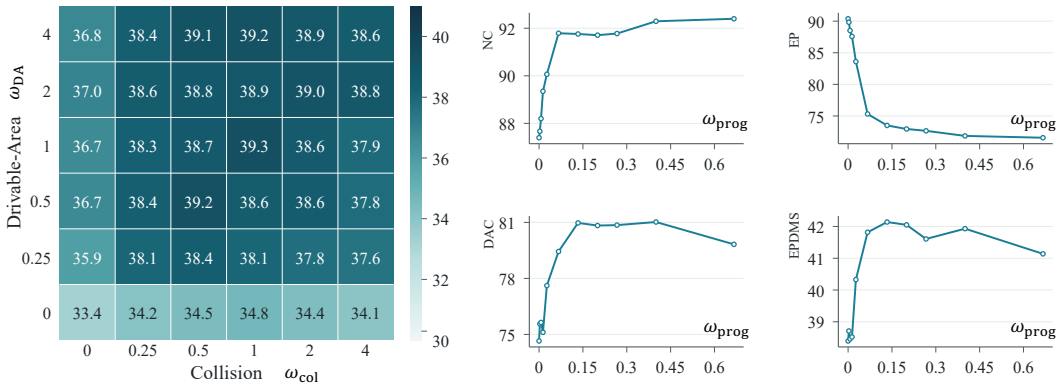

Figure C.1: **Sensitivity to energy-composition weights on NAVSIM-v2 `navhard`.** All sweeps use the final EMA checkpoint after 500 training epochs of the base model without contrastive training. **Left:** Overall two-stage EPDMS under a joint sweep of the drivable-area and collision weights, with $w_{\text{prog}} = 0$. **Right:** Stage-2 NC, DAC, EP, and overall two-stage EPDMS under a progress-weight sweep, with $w_{\text{DA}} = w_{\text{col}} = 1$.

Table C.3: Comparison with GuideFlow on NAVSIM-v2 `navhard`. GuideFlow results are from Tables 1 and 4 of Liu et al. (2026b), arXiv:2511.18729v3; EnergyDrive uses our evaluation. The groups distinguish the use of a separate PDM-score-supervised scorer, not all forms of simulator-derived supervision. Training data and inference settings are not matched across methods.

Method	Result source	EPDMS↑
Without a separate PDM-score-supervised scorer		
GuideFlow (command)	Reported, Table 1	27.1
GuideFlow (plan anchor)	Reported, Table 4	29.0
EnergyDrive (Ours)	Our evaluation	44.2
With a separate PDM-score-supervised scorer (context)		
GuideFlow+Scorer	Reported, Table 1	43.0
GuideFlow+Scorer†	Reported, Table 1	46.7

The scorer-augmented variants use GTRS-Dense. † denotes an adjusted trajectory-scoring strategy at inference, following GuideFlow's Table 1. GuideFlow reports training on `navtrain`; EnergyDrive additionally uses SimScale pseudo-expert trajectories and binary safety checks for negative mining.

waypoints at 0.5 s intervals, and the simulator's iLQR controller converts this plan to controls. The simulation advances at 0.25 s intervals with a maximum of 400 steps per episode.

We use the converged-controller configuration: the upstream 0.05 s wall-clock cutoff is removed, and iLQR runs until its convergence tolerance or iteration limit. This will provide a reproducible evaluation protocol and avoid hardware-dependent controller behavior. Rendering, dynamics, traffic-agent behavior, and scoring otherwise follow the benchmark implementation. Aggregate HD scores are scenario-weighted means over the released scenario set.

We directly transfer the NAVSIM-trained final EMA checkpoints without retraining on HUGSIM. For composed selection, the weights are $(w_{\text{DA}}, w_{\text{col}}, w_{\text{prog}}) = (1, 1, 0)$; all energy normalizations remain as in Eq. B.6. We distinguish the full model from *EnergyDrive without contrastive training*, and distinguish single-proposal sampling from composed selection for either model.

C.5 Comparison with Guidance-Based Planning

Results and provenance. GuideFlow (Liu et al., 2026b) combines constraint-guided flow matching with energy-based refinement. Table C.3 compares its published `navhard` scores with EnergyDrive. We use paper-reported results because we did not obtain an evaluable checkpoint for the reported configurations, and a validated in-house reproduction was not available. These entries were not re-evaluated by us.

Comparison scope. The scorer-augmented variants combine GuideFlow with a separate GTRS-Dense network trained using simulator-derived trajectory-score targets (Li et al., 2025c). Their performance reflects both trajectory generation and score-supervised selection, rather than guidance alone. We include them for context but distinguish them from the main comparison scope in Appendix C.1. This distinction concerns the separate scoring network; EnergyDrive still uses pseudo-expert trajectories and binary safety checks during training.

Interpretation. EnergyDrive's 44.2 EPDMS exceeds the reported no-scorer configurations and the 43.0 scorer-augmented configuration, but not the adjusted-scoring variant at 46.7. The comparison places EnergyDrive competitively among these reported systems without a separate PDM-score-supervised scorer. Since training data and inference procedures differ, these results do not isolate the effect of the generative formulation or establish an advantage over all guidance-based planners.