Beyond Imitation: Constraint-Aware Trajectory Generation with Flow Matching For End-to-End Autonomous Driving

Lin Liu[1,2], Guanyi Yu[2], Ziying Song [1], JunQiao Li[2], Caiyan Jia[1], Feiyang Jia[1], Peiliang Wu[3], Yandan Luo[4]
[1]Beijing Jiaotong University [2]Qcraft [3]Yanshan University [4]The University of Queensland
{23120379, songziying, cyjia}@bjtu.edu.cn.

Abstract

*Planning is a critical component of end-to-end autonomous driving. However, prevailing imitation learning methods often suffer from mode collapse, failing to produce diverse trajectory hypotheses. Meanwhile, existing generative approaches struggle to incorporate crucial safety and physical constraints directly into the generative process, necessitating an additional optimization stage to refine their outputs. To address these limitations, we propose CATG, a novel planning framework that leverages Constrained Flow Matching. Concretely, CATG explicitly models the flow matching process, which inherently mitigates mode collapse and allows for flexible guidance from various conditioning signals. Our primary contribution is the novel imposition of explicit constraints directly within the flow matching process, ensuring that the generated trajectories adhere to vital safety and kinematic rules. Secondly, CATG parameterizes driving aggressiveness as a control signal during generation, enabling precise manipulation of trajectory style. Notably, on the NavSim v2 challenge, CATG achieved **2nd** place with an EPDMS score of 51.31 and was honored with the **Innovation Award**.*

1. Introduction

End-to-end multimodal planning [2, 10, 13, 15] has established itself as a critical methodology in autonomous driving systems, significantly enhancing robustness and adaptability during inference when compared to single-trajectory prediction approaches. This capability is especially vital in ambiguous or highly interactive driving scenarios—such as unprotected left turns, merging in dense traffic, or navigating intersections—where multiple distinct trajectories may be equally appropriate. Despite these advantages, the majority of contemporary multimodal methods remain dependent on imitation learning frameworks. Such approaches [2, 3, 8–10, 14, 15] learn from a limited set of demonstrated expert trajectories, and due to the lack of strategy diversity of ground-truth trajectories, often yield predictions that are homogenized, and deficient in behavioral diversity.

In response to these shortcomings, several alternative strategies have been proposed. A series of works incorporates generative models, such as diffusion processes, to capture a broader distribution of plausible trajectories. However, many of these methods [13, 17] do not explicitly supervise the generative denoising process, still relying heavily on behavior cloning objectives. As a result, they remain susceptible to mode collapse. Another paradigm [16, 18, 19] represents a further shift, depending entirely on generative models for trajectory planning and abandoning the use of imitation learning. While these methods benefit from generative models, they introduce new challenges: the stochasticity in noise initialization can lead to high-variance predictions, and the absence of a mechanism for hard constraint integration, such as obstacle avoidance or compliance with traffic rules, compromises the safety and interpretability of generated trajectories.

To address these limitations, we propose CATG, a novel trajectory generation framework based on flow matching that completely eliminates imitation learning while enabling flexible injection of explicit constraints into the generative process. Our contributions are threefold:

(1) **Novel generative framework.** We introduce CATG, a multimodal trajectory generator built upon flow matching. Unlike conventional methods, CATG eliminates the reliance on imitation learning while supporting diverse and flexible conditional controls.

(2) **Constraint-guided generation.** We explicitly integrate feasibility and safety constraints into the generative process through a progressive mechanism: prior-informed anchor design is used to construct constraint-guided probability flows, and energy-based guidance further steers trajectories toward feasible regions.

(3) **Reward-conditioned controllability.** We treat environmental reward signals as conditional inputs, enabling controllable trade-offs between aggressive and conservative driving styles during inference.

CATG is extensively evaluated on the ICCV NAVSIM V2 End-to-End Driving Challenge, where it demonstrates superior planning accuracy and robust generalization to out-of-distribution data. When combined with an open-source scoring model, CATG achieves an EPDMS score of 51.31, competitive with state-of-the-art alternatives.

2. Preliminary

Let $\mathbb{R}^d$ denote the data space, two important objects we use in this paper are: the probability density path $p : [0,1] \times \mathbb{R}^d \to \mathbb{R}_{>0}$, which is a time dependent probability density function i.e., $\int p_t(x)dx = 1$, and a time-dependent vector field, $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. A vector field v_t can be used to construct a time-dependent diffeomorphic map, called a flow, $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. This flow serves as a probability path $p_t(x)$ connecting the source distribution $X_0 \sim \pi_0$ and target distribution $X_1 \sim \pi_1$, defined via the ordinary differential equation (ODE):

$$\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)) \tag{1}$$

$$\phi_0(x) = x \tag{2}$$

And, we can model the vector field v_t with a neural network, $v_t(t;\theta)$. Let X_1 denote a random variable distributed according to an unknown data distribution π_1. We assume that we only have access to data samples from π_1, but not to the density function itself. Furthermore, we let π_0 be a simple distribution, such as a standard normal distribution. Given a target probability density path $p_t(x)$ and a corresponding vector field $u_t(x)$, which generates $p_t(x)$, we define the Flow Matching (FM) objective as:

$$L_{FM}(\theta) = \mathbb{E}_{t,p_t(x)}||v_t(x) - u_t(x)||^2 \tag{3}$$

In CATG, we use rectified flow to construct a probability path ϕ:

$$X_t = tX_1 + (1-t)X_0 \tag{4}$$

So, the drift force $v : \pi_0 \to \pi_1$ is set to drive the flow to follow the direction $(X_1 - X_0)$ of the linear path pointing from X_0 to X_1 as much as possible, by solving a simple least squares regression problem:

$$min_v \int_0^1 \mathbb{E}[||(X_1 - X_0) - v(X_t, t)||^2]dt, \tag{5}$$

where X_t is the linear interpolation of X_0 and X_1.

3. Method

3.1. Flexible conditioning signal

We followed the Transfuser [4] as our perception backbone. For flow matching progress, we sample X_0 from a standard Gaussian distribution and normalize the target trajectory X_1 to the range $[-1,+1]$. CATG constructs a flow with the starting point as X_0 and the endpoint asX_1. Then, we apply positional encoding to X_t and utilize a Unet Encoder [5] to encode X_t into a feature F_{X_t}. Subsequent to the CATG perception module, CATG obtains the agent's query Q_{ag}, ego query Q_{eg}, and BEV feature F_B. In a separate preprocessing step, the BEV map segmentation result is first converted into a binary road map $M_{0,1}$ and then fused with BEV grid positional encoding Pos_B. Finally, CATG fuses the feature F_{X_t} with all these elements (Q_{ag}, Q_{eg}, F_B and $M_{0,1}$) through multiple layers of cross-attention as shown in Fig. 2.

$$F_{X_t} = F_{X_t} + Tim_{ebed}(t) \tag{6}$$

$$F_{X_t} = CrossAttn(F_{X_t}, Q_{ag}) \tag{7}$$

$$F_{X_t} = CrossAttn(F_{X_t}, Fusion(F_B, M_{0,1}, Pos_B)) \tag{8}$$

$$F_{X_t} = CrossAttn(F_{X_t}, Q_{eg}) \tag{9}$$

In order to flexibly control the trajectory generation style in a classifier-free manner [7] during inference, we introduce three distinct types of conditional control signals:

(1) **Trajectory anchor:** CATG treat pre-clustered trajectory anchors as high-level abstractions of driving modes. CATG first constructs a trajectory vocabulary $vocab_{anchor}$ of size 8,192 by applying FPS (farthest-point sampling) over the entire training dataset. CATG is trained in a classifier-free guidance [7] manner, where driving anchors are incorporated as conditional signals to guide trajectory generation. During training, the anchor most similar to the GT trajectory is utilized as the conditional signal, which is determined by DTW distance between trajectory vocabulary and GT trajectory. At inference time, a pre-trained scoring model, GTRS [12] (with a V2-99 backbone), is employed to select the top-100 anchors with the highest likelihood, which subsequently serve as conditional inputs for generating diverse and compliant trajectories.

(2) **Target point:** During training, CATG takes the endpoint of the GT trajectory as the conditional signal. During testing, in contrast, the endpoint of the anchor obtained from a scoring model serves as the conditional control signal.

(3) **Driving command:** The driving command is also a type of control signal. CATG converts the command types in NAVSIM [6] into a one-hot encoding for use as a conditional signal.

3.2. Constraint-Aware Trajectory Generation

A significant challenge in generative models is the lack of interpretability in their intermediate representations, posing difficulties for directly constraining the outputs. Specifically in the NAVSIM V2 [6] challenge, constraining the generated trajectories to satisfy the Driving Area Compli-

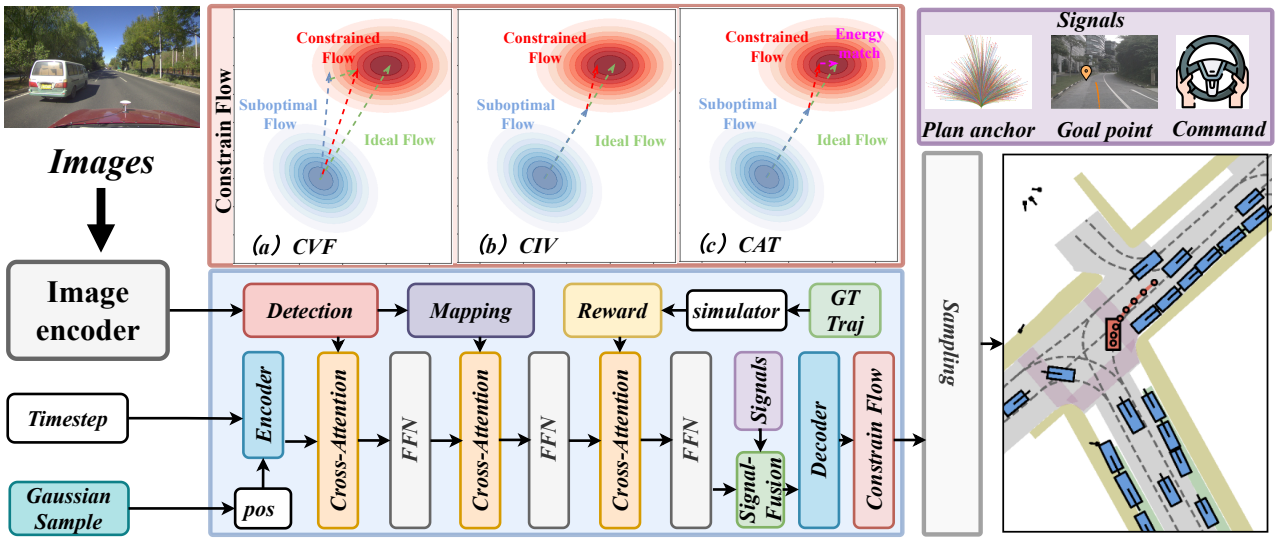

Figure 1. The architecture of CATG with Flow Matching. The image is encoded into image features, which are subsequently fed into detection and mapping modules to generate perceptual features. Before training, CATG processes each gt trajectory (GT Traj) through simulator to compute and offline store its score. Subsequently, CATG encodes Gaussian-sampled latent variables and the timestamp t into input features via a encoder. These features are fused with perceptual features, trajectory rewards, and conditioning signals (including plan anchor, goal point, and driving command) through cross-attention. The signal fusion module consists of multi-layer cross-attention blocks and adheres to the classifier-free [7] training paradigm. The velocity field decoded by the decoder is refined via our three correction strategies: CVF, CIV, and CAT (Sec 3.2). Finally, the driving trajectory is generated through sampling.

ance (DAC) metric proved highly challenging. Unlike constraints such as inter-agent collision avoidance which can be integrated by using vehicle distances as conditional signals, as seen in Diffusion-Planner [19], road geometry is far more complex. Therefore, in the following discussion, we will primarily focus on constraining trajectories to satisfy road compliance. However, it is noteworthy that our method can also be adapted to other types of constraints. To address this, we introduced three more direct and efficient methods for constraining the generation. The Flow Matching generation process is formulated as:

$$X_{t+1} = X_t + v_t dt \tag{10}$$

Since the formulation above indicates that the generated state X_{t+1} at the next timestep is determined by the intermediate variable X_t and the velocity field v_t, a compelling hypothesis arises: could one constrain the generation process by imposing constraints on these two quantities ?

(1) **Constraining velocity field v_t (CVF):** Based on the road segmentation result, a trajectory X_1^C that satisfying the DAC constraint is first selected from trajectory vocabulary $vocab_{anchor}$. Subsequently, for a given Gaussian sample X_0 as the flow's starting point, the ideal velocity field that leads to trajectory X_1^C can be computed.

$$v_t^c = \frac{X_1^C - X_0}{1 - 0} \tag{11}$$

CATG leverages this precomputed field v_t^c to correct the potentially biased velocity field v_t predicted by the model. Consequently, we propose the concept of a synthetic velocity field $v_t^{'}$, which is a combination of the model predicted velocity field v_t and the precomputed one v_t^c during the sampling process as shown in Fig. 1 (a):

$$v_t^{'} = v_t + \frac{2\lambda v_t \cdot v_t^c}{||v_t^c||^2} v_t^c, \tag{12}$$

where λ was set to -0.1.

(2) **Constraining intermediate variables X_t (CIV):** A flow generated by a model-predicted velocity field often deviates from the ideal, leading to a final sample that fails to meet constraints. This flow can be discretized into a series of intermediate variables $X_0, ..., X_t, ..., X_1$; Therefore, if these intermediate variables can be effectively constrained, the final generated outcome can consequently be controlled. However, correcting X_t at every timestep is inefficient. Instead, inspired by [11], CATG addresses this by correcting the flow at its origin. It replaces the initial Gaussian random sample X_0 with an anchor X_1^C selected from the trajectory vocabulary $vocab_{anchor}$ as shown in Fig. 1 (b),

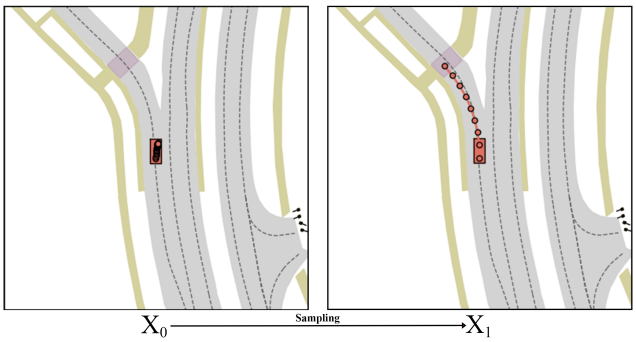

Figure 2. The figure illustrates that sampling by using an anchor as the starting point results in a more reasonable trajectory.

which complies with the DAC constraint, even though this anchor might perform poorly on other evaluation metrics. However, CATG can refines this anchor to make it more reasonable. As shown in Fig. 2, this approach of starting from a DAC-compliant anchor enables the model to produce more plausible trajectories.

(3) **Constraint-Aware Training (CAT):** In contrast to Diffusion-Planner [19], which only introduces energy term during inference, we incorporate constraints into the training phase by encoding them as an energy function. When trajectory are sampled along the direction of ascending energy, they exhibit a higher probability of satisfying the constraints as shown in Fig. 1 (c). Specifically, the DAC constraint can be represented by computing a Euclidean Signed Distance Field. The energy of a trajectory decreases as it moves closer to the road boundary, penalizing undesirable deviations. We follow the Energy Matching [1] framework for model training. A two-stage procedure is employed, the first stage trains the Flow Matching process, and the second stage trains the Energy Matching process.

3.3. Reward as condition:

To control trajectory aggressiveness at inference time, CATG utilizes an EP (ego process) score as a conditioning signal. This score is derived by evaluating each GT trajectory in the NavTrain set within the NAVSIM simulator. By setting the EP condition to 1 during inference, the model is encouraged to produce more aggressive driving behavior.

4. Experiments

4.1. Experiments Setup

Our model is trained in two stages. The first stage of training encompasses the Flow Matching process, the perception module, and the map segmentation module. It was conducted with a batch size of 64, a learning rate of 2×10^{-4}, and trained for 90 epochs by using **NavTrain split**. The second stage of training adhered to the Energy Matching framework, focusing solely on fine-tuning the Flow Matching process. This stage used a batch size of 64, a learning rate of 2×10^{-4}, and trained for 10 epochs by using **NavTrain split**. During inference, CATG generates 100 candidate trajectories with 100 sampling steps.These candidates and trajectory vocabulary $vocab_{anchor}$ are then ranked by an open-source, pre-trained GTRS [12] scorer model (with a V2-99 backbone) to select the most plausible trajectory as the final output.

Table 1. Results of proposed CATG architecture in NAVSIM V2

Metric Name	Team: bjtu_jia_team & qcraft
extended pdm score combined	51.3116
no at fault collisions stage one	98.2142
drivable area compliance stage one	100
driving direction compliance stage one	99.6428
traffic light compliance stage one	100
ego progress stage one	80.8379
time to collision within bound stage one	98.5714
lane keeping stage one	90
history comfort stage one	94.2857
two frame extended comfort stage one	57.1428
no at fault collisions stage two	88.9016
drivable area compliance stage two	95.4416
driving direction compliance stage two	97.9186
traffic light compliance stage two	96.8362
ego progress stage two	77.9218
time to collision within bound stage two	88.0227
lane keeping stage two	56.6261
history comfort stage two	98.3082
two frame extended comfort stage two	64.4264

4.2. Experiments result

We present our proposed CATG architecture's results as shown in Tab. 1.

5. Limitation

Sampling trajectories with 100 steps remains computationally expensive. Nevertheless, accelerating this process may lead to a degradation in trajectory quality. Therefore, a promising direction for future work is to enhance sampling efficiency while preserving the quality of the generated trajectories.

6. Conclusion

We presents an end-to-end planner that leverages flow matching. Our approach is capable of incorporating flexible conditional signals to control trajectory generation. Furthermore, we innovatively propose three distinct strategies to enforce explicit constraints throughout the generation process. Experimental results presented in Tab. 1 demonstrate that our framework achieves a EPDMS of 51.31.

References

[1] Michal Balcerak, Tamaz Amiranashvili, Antonio Terpin, Suprosanna Shit, Lea Bogensperger, Sebastian Kaltenbach, Petros Koumoutsakos, and Bjoern Menze. Energy matching: Unifying flow matching and energy-based models for generative modeling, 2025. 4

[2] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024. 1

[3] Zhili Chen, Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Ppad: Iterative interactions of prediction and planning for end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 239–256. Springer, 2025. 1

[4] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023. 2

[5] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022. 2

[6] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024. 2

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 2, 3

[8] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17853–17862, 2023. 1

[9] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21983–21994, 2023.

[10] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023. 1

[11] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[12] 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, 2025. 2, 4

[13] 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. *arXiv preprint arXiv:2411.15139*, 2024. 1

[14] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22432–22441, 2025. 1

[15] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024. 1

[16] Junming Wang, Xingyu Zhang, Zebin Xing, Songen Gu, Xiaoyang Guo, Yang Hu, Ziying Song, Qian Zhang, Xiaoxiao Long, and Wei Yin. He-drive: Human-like end-to-end driving with vision language models, 2024. 1

[17] 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, 2025. 1

[18] Wenzhao Zheng, Ruiqi Song, Xianda Guo, and Long Chen. Genad: Generative end-to-end autonomous driving. *arXiv preprint arXiv:2402.11502*, 2024. 1

[19] Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance, 2025. 1, 3, 4