Hydra-MDP: マルチターゲットHydra-Distillationによるエンドツーエンドのマルチモーダルプランニング

李振新[1,2] 李海林[3] 王志浩[1,4] 蘭世毅[1] 于志丁[1] 季義信[5] 李志基[5] 朱志岳[6] Jan Kautz[1]
Zuxuan Wu[2] Yu-Gang Jiang[2] Jose M. Alvarez[1] [1] NVIDIA[2] 復旦大学[3] 東中国師範大学[4]
北京工業大学[5] 南京大学[6] 南開大学

Abstract

我々は、教師-生徒モデルに複数の教師を用いる新しいパラダイムであるHydra-MDPを提案する。このアプローチでは、生徒モデルを学習するために、人間とルールベースの教師の両方からの知識抽出を使用し、様々な評価指標に合わせた多様な軌道候補を学習するためのマルチヘッドデコーダを特徴とする。ルールベースの教師の知識により、Hydra-MDPは、環境が計画にどのような影響を与えるかを、非差別的な後処理に頼るのではなく、エンドツーエンドで学習する。この方法はNavsimチャレンジで1st位を達成し、多様な運転環境と条件下で汎化の大幅な改善を実証した。詳細はhttps://github.com/NVlabs/Hydra-MDP をご覧ください。

1. Introduction

エンドツーエンドの自律走行は、生のセンサー入力でニューラルプランナーを学習するもので、完全な自律性を実現するための有望な方向性と考えられている。この分野での有望な進歩にもかかわらず[11, 12]、最近の研究[4, 8, 14]では、模倣学習(IL)手法の複数の脆弱性と限界、特に機能不全メトリクスや暗黙のバイアス[8, 14]などのオープンループ評価における固有の問題が明らかにされている。安全性、効率性、快適性、交通ルールの遵守を保証できないため、これは非常に重要である。この主な制限に対処するために、いくつかの研究が閉ループメトリクスを組み込むことを提案しており、これは、機械学習されたプランナが、単に人間のドライバーを模倣するだけでなく、本質的な基準を満たすことを保証することによって、エンドツーエンドの自律走行をより効果的に評価する。

したがって、エンドツーエンドのプランニングは、理想的にはマルチターゲットかつマルチモーダルなタスクであり、マルチターゲットプランニングは、オープンループとクローズドループの設定から様々な評価指標を満たすことを含む。この文脈において、マルチモーダルは各メトリックに対して複数の最適解が存在することを示す。

Existing end-to-end approaches [4, 11, 12] often try to

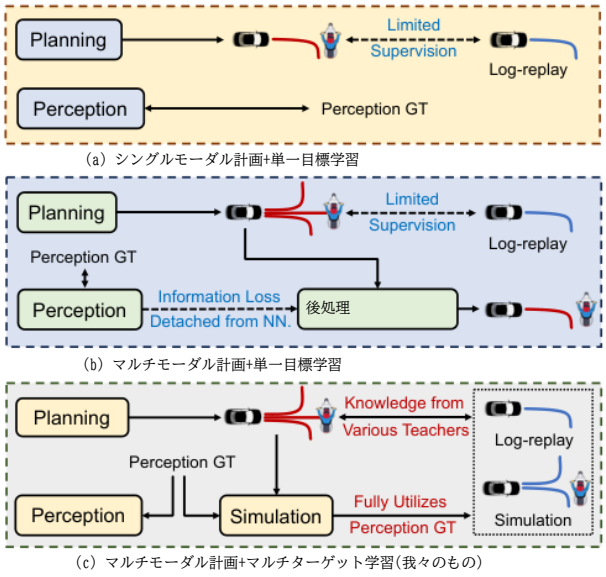

図1. エンドツーエンドの計画パラダイムの比較。

後処理による閉ループ評価を考慮するが、これは合理化されておらず、完全なエンドツーエンドパイプラインと比較して追加情報が失われる可能性がある。一方、ルールベースのプランナ[8, 18]は不完全な知覚入力に苦戦する。これらの不完全な入力は、グランドトゥルース(GT)ラベルの代わりに予測された知覚に依存するため、クローズドループとオープンループの両方のメトリクスの下でルールベースのプランニングの性能を低下させる。

この問題に対処するために、我々はHydra-MDP(Multi-target Hydra-distillationによるマルチモーダル計画)と呼ばれる新しいエンドツーエンドの自律走行フレームワークを提案する。HydraMDPは、新しい教師-生徒知識蒸留(KD)アーキテクチャに基づいている。生徒モデルは、人間とルールベースの教師の両方から、KDを通して様々な評価指標に合わせた多様な軌道候補を学習する。マルチターゲットHydra-distillationをマルチヘッドデコーダでインスタンス化することで、特化した教師からの知識を効果的に統合する。

Hydra-MDP: End-to-end Multimodal Planning with Multi-target Hydra-Distillation

Zhenxin Li[1, 2] Kailin Li[3] Shihao Wang[1, 4] Shiyi Lan[1] Zhiding Yu[1] Yishen Ji[5]
Zhiqi Li[5] Ziyue Zhu[6] Jan Kautz[1] Zuxuan Wu[2] Yu-Gang Jiang[2] Jose M. Alvarez[1]
[1]NVIDIA [2]Fudan University [3]East China Normal University
[4]Beijing Institute of Technology [5]Nanjing University [6]Nankai University

Abstract

We propose Hydra-MDP, a novel paradigm employing multiple teachers in a teacher-student model. This approach uses knowledge distillation from both human and rule-based teachers to train the student model, which features a multi-head decoder to learn diverse trajectory candidates tailored to various evaluation metrics. With the knowledge of rule-based teachers, Hydra-MDP learns how the environment influences the planning in an end-to-end manner instead of resorting to non-differentiable post-processing. This method achieves the 1^{st} place in the Navsim challenge, demonstrating significant improvements in generalization across diverse driving environments and conditions. More details by visiting https://github.com/NVlabs/Hydra-MDP.

1. Introduction

End-to-end autonomous driving, which involves learning a neural planner with raw sensor inputs, is considered a promising direction to achieve full autonomy. Despite the promising progress in this field [11, 12], recent studies [4, 8, 14] have exposed multiple vulnerabilities and limitations of imitation learning (IL) methods, particularly the inherent issues in open-loop evaluation, such as the dysfunctional metrics and implicit biases [8, 14]. This is critical as it fails to guarantee safety, efficiency, comfort, and compliance with traffic rules. To address this main limitation, several works have proposed incorporating closed-loop metrics, which more effectively evaluate end-to-end autonomous driving by ensuring that the machine-learned planner meets essential criteria beyond merely mimicking human drivers.

Therefore, end-to-end planning is ideally a multi-target and multimodal task, where multi-target planning involves meeting various evaluation metrics from either open-loop and closed-loop settings. In this context, multimodal indicates the existence of multiple optimal solutions for each metric.

Existing end-to-end approaches [4, 11, 12] often try to consider closed-loop evaluation via post-processing, which is not streamlined and may result in the loss of additional information compared to a fully end-to-end pipeline. Meanwhile, rule-based planners [8, 18] struggle with imperfect perception inputs. These imperfect inputs degrade the performance of rule-based planning under both closed-loop and open-loop metrics, as they rely on predicted perception instead of ground truth (GT) labels.

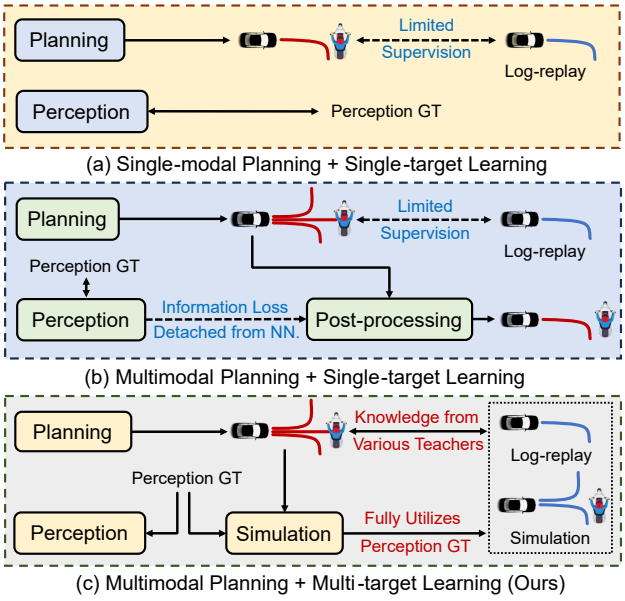

Figure 1. **Comparison between End-to-end Planning Paradigms.**

To address the issues, we propose a novel end-to-end autonomous driving framework called Hydra-MDP (Multimodal Planning with Multi-target Hydra-distillation). Hydra-MDP is based on a novel teacher-student knowledge distillation (KD) architecture. The student model learns diverse trajectory candidates tailored to various evaluation metrics through KD from both human and rule-based teachers. We instantiate the multi-target Hydra-distillation with a multi-head decoder, thus effectively integrating the knowledge from specialized teachers. Hydra-MDP also features an ex-

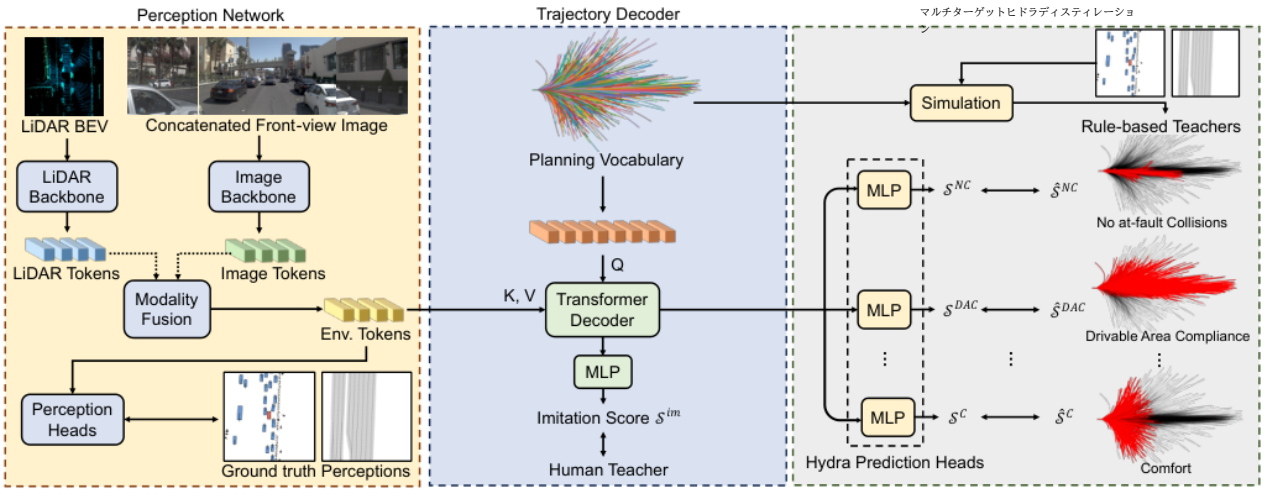

図2. Hydra-MDPの全体アーキテクチャ。

Hydra-MDPはまた、拡張可能なKDアーキテクチャを特徴としており、追加の教師を簡単に統合することができる。

生徒モデルは学習時に環境観測を使用し、教師モデルはグランドトゥルース(GT)データを使用する。この設定により、教師モデルはより良い計画予測を生成することができ、生徒モデルが効果的に学習できるようになる。環境観測で学生モデルを訓練することで、テスト中にGT知覚にアクセスできない現実的な条件を扱うことに長けている。

Our contributions are summarized as follows:

1. We propose a universal framework of end-to-end multi-modal planning via multi-target hydra-distillation, allowing the model to learn from both rule-based planners and human drivers in a scalable manner.

2. 2. Navsim におけるシミュレーションベースの評価指標において、本アプローチは最先端の性能を達成した。

2. Solution

2.1. Preliminaries

Oはセンサーの観測値、P^とPはグランドトゥルースと予測された知覚(3D物体検出、車線検出など)、T^はエキスパート軌道、T*は予測された軌道を表すとする。L_{im}は模倣損失を表す。本節では、まず、一般的な2つのパラダイムと、我々の提案するパラダイム(図1)を紹介する:

A. シングルモーダル計画+単一目標学習。このパラダイム[11, 12, 14]では、計画ネットワークはセンサーの観測値から計画された軌道を直接回帰する。グランドトゥルースの認識は補助的な監督として使用することができるが、計画出力には影響しない。知覚損失は単純化のため式には含まれていない。処理全体は次のように定式化できる:

$$\mathcal{L} = \mathcal{L}_{im}(T^*, \hat{T}), \tag{1}$$

ここで、L_{im}は通常L2損失である。

B. マルチモーダル計画+単一目標学習。このアプローチ[1, 4]は、複数の軌跡{T_i } ki=1 を予測し、そのエキスパート軌跡との類似度を計算する:

$$\mathcal{L} = \sum_i \mathcal{L}_{im}(T_i, \hat{T}), \tag{2}$$

ここで、L_{im}はKL-Divergence [4]またはmax-margin loss [1]である。知覚出力Pは、コスト関数f (T_i, P) を介して、適切な軌道を後処理するために明示的に使用される。最もコストの低い軌道が選択される:

$$T^* = \arg\min_{T_i} f(T_i, P), \tag{3}$$

これは不完全知覚Pに基づく非微分化プロセスである。

C. マルチモーダル計画+マルチターゲット学習。我々は、ニューラルネットワークf ～を介して、様々なコスト(例えば、衝突コスト、走行可能領域コンプライアンスコスト)を同時に予測するこのパラダイムを提案する。これは教師-生徒蒸留法で行われ、教師は真実の知覚P^にアクセスできるが、生徒はセンサーの観測値Oにのみ依存する。このパラダイムは次のように定式化できる:

$$\mathcal{L} = \sum_i \mathcal{L}_{im}(T_i, \hat{T}) + \mathcal{L}_{kd}(f(T_i, \hat{P}), \tilde{f}(T_i, O)). \tag{4}$$

ここでは、わかりやすくするために1つのコスト関数fのみを考える。予測コストが最も低い軌道が選択される:

$$T^* = \arg\min_{T_i} \tilde{f}(T_i, O). \tag{5}$$

このフレームワークは、微分不可能な後処理によって制限されるものではないことを強調する。より多くのコスト関数を含むか、我々の実装で模倣類似性を活用することで、エンドツーエンドで簡単にスケーリングすることができます(Sec. 2.4)。

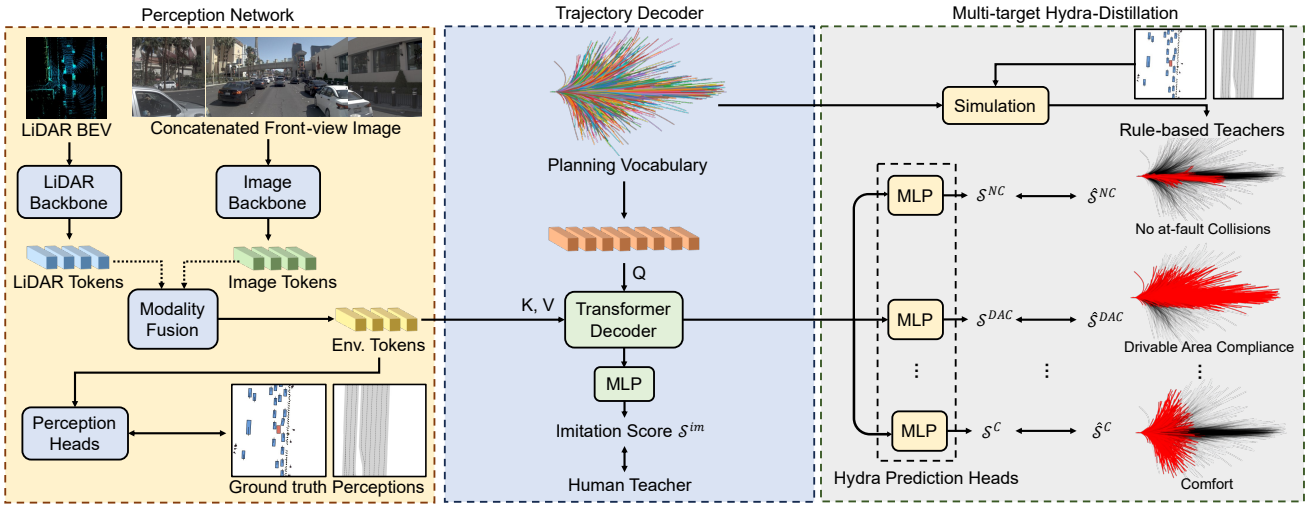

Figure 2. **The Overall Architecture of Hydra-MDP.**

tendable KD architecture, allowing for easy integration of additional teachers.

The student model uses environmental observations during training, while the teacher models use ground truth (GT) data. This setup allows the teacher models to generate better planning predictions, helping the student model to learn effectively. By training the student model with environmental observations, it becomes adept at handling realistic conditions where GT perception is not accessible during testing.

Our contributions are summarized as follows:

1. We propose a universal framework of end-to-end multimodal planning via multi-target hydra-distillation, allowing the model to learn from both rule-based planners and human drivers in a scalable manner.
2. Our approach achieves the state-of-the-art performance under the simulation-based evaluation metrics on Navsim.

2. Solution

2.1. Preliminaries

Let O represent sensor observations, $\hat{P}$ and P denote ground truth and predicted perceptions (*e.g.* 3D object detection, lane detection), $\hat{T}$ be the expert trajectory, and T^* be the predicted trajectory. $\mathcal{L}_{im}$ represents the imitation loss. We first introduce the two prevailing paradigms and our proposed paradigm (Fig. 1) in this section:

A. Single-modal Planning + Single-target Learning. In this paradigm [11, 12, 14], the planning network directly regresses the planned trajectory from the sensor observations. Ground truth perceptions can be used as auxiliary supervision but does not influence the planning output. Perception losses are not included in the formula for simplicity. The whole processing can be formulated as:

$$\mathcal{L} = \mathcal{L}_{im}(T^*, \hat{T}), \tag{1}$$

where $\mathcal{L}_{im}$ is usually an L2 loss.

B. Multimodal Planning + Single-target Learning. This approach [1, 4] predicts multiple trajectories $\{T_i\}_{i=1}^{k}$, whose similarities to the expert trajectory are computed:

$$\mathcal{L} = \sum_i \mathcal{L}_{im}(T_i, \hat{T}), \tag{2}$$

where $\mathcal{L}_{im}$ can be KL-Divergence [4] or the max-margin loss [1]. Perception outputs P are explicitly used to post-process suitable trajectories via a cost function $f(T_i, P)$. The trajectory with the lowest cost is selected:

$$T^* = \arg\min_{T_i} f(T_i, P), \tag{3}$$

which is a non-differentiable process based on imperfect perception P.

C. Multimodal Planning + Multi-target Learning. We propose this paradigm to simultaneously predict various costs (e.g., collision cost, drivable area compliance cost) via a neural network $\tilde{f}$. This is performed in a teacher-student distillation manner, where the teacher has access to ground truth perception $\hat{P}$ but the student relies only on sensor observations O. This paradigm can be formulated as:

$$\mathcal{L} = \sum_i \mathcal{L}_{im}(T_i, \hat{T}) + \mathcal{L}_{kd}(f(T_i, \hat{P}), \tilde{f}(T_i, O)). \tag{4}$$

Here, we only consider one cost function f for clarity. The trajectory with the lowest predicted cost is selected:

$$T^* = \arg\min_{T_i} \tilde{f}(T_i, O). \tag{5}$$

We stress that this framework is not restricted by non-differentiable post-processing. It can be easily scaled in an end-to-end fashion by involving more cost functions or leveraging imitation similarity in our implementation (Sec. 2.4).

2.2. 全体的な枠組み

図2に示すように、Hydra-MDPは知覚ネットワークと軌跡デコーダの2つのネットワークから構成される。

知覚ネットワーク。我々の知覚ネットワークは、画像バックボーン、LiDARバックボーン、3Dオブジェクト検出とBEVセグメンテーションのための知覚ヘッドから構成される、公式チャレンジベースラインTransfuser [5, 6]をベースに構築されている。複数の変換層[19]が、両方のバックボーンのステージから特徴を接続し、異なるモダリティから意味のある情報を抽出する。知覚ネットワークの最終出力は環境トークンF_{env}からなり、画像とLiDAR点群の両方から得られる豊富な意味情報を符号化する。

軌跡デコーダ。Vadv2[4]に従い、連続行動空間を離散化するために固定計画語彙を構築する。語彙を構築するために、まずオリジナルのnuPlanデータベース[2]からランダムに700Kの軌跡をサンプリングする。各軌跡T_i (i = 1, · · · , k)は、(x,y,ヘディング)の40個のタイムスタンプから構成され、チャレンジにおける所望の10Hz周波数と4秒の未来ホライズンに対応する。計画語彙V_kは、700K個の軌跡のK-meansクラスタリング中心として形成される。V_kはMLPでk個の潜在クエリとして埋め込まれ、変換エンコーダ[19]の層に送られ、エゴステータスEに追加される:

$$\mathcal{V}_k' = Transformer(Q, K, V = Mlp(\mathcal{V}_k)) + E. \quad (6)$$

F_{env}に環境の手がかりを取り込むために、変換デコーダが活用される:

$$\mathcal{V}_k'' = Transformer(Q = \mathcal{V}_k', K, V = F_{env}). \quad (7)$$

ログ再生軌跡T^を用いて、距離ベースˆのクロスエントロピー損失を実装し、人間のドライバーを模倣する:

$$\mathcal{L}_{im} = -\sum_{i=1}^{k} y_i \log(\mathcal{S}_i^{im}), \quad (8)$$

ここで、S_i^{im}はV_k'のi番目のソフトマックススコアであり、y_iは対数再生と語彙間のL2距離によって生成される模倣目標である。L2距離にソフトマックスを適用し、確率分布を生成する:

$$y_i = \frac{e^{-(\hat{T}-T_i)^2}}{\sum_{j=1}^{k} e^{-(\hat{T}-T_j)^2}}. \quad (9)$$

この模倣目標の背後にある直感は、人間の運転行動に近い軌道提案に報酬を与えることである。

2.3. マルチターゲット水蒸気蒸留法

模倣目標はプランナーに一定の手がかりを与えるが、クローズドループ設定の下では、モデルが計画決定を運転環境と関連付けるには不十分であり、衝突や走行可能領域の離脱などの失敗につながる[14]。

そこで、エンドツーエンドのプランナの閉ループ性能を向上させるために、本課題においてプランナをシミュレーションベースのメトリクスと整合させる学習戦略であるMulti-target Hydra-Distillationを提案する。

蒸留プロセスは、2つのステップを通して学習目標を拡張する:(1) 訓練データセット全体に対する計画語彙V_kのオフラインシミュレーション[8]を実行する。(2) 訓練プロセス中にV_kの各軌跡に対するシミュレーションスコアからの監視を導入する。与えられたシナリオに対して、ステップ1は、各メトリックm∈Mとi番目の軌道に対して、グランドトゥルースシミュレーションスコア$\{\hat{S}_i^m \mid i = 1, \cdots, k\}_{m=1}^{|M|}$を生成する。スコア予測のために、潜在ベクトル$V_k'$は、ヒドラ予測ヘッドのセットで処理され、予測スコア$\{S_i^m \mid i = 1, \cdots, k\}_{m=1}^{|M|}$ が得られる。バイナリクロスエントロピー損失により、ルールベースの運転知識をエンドツーエンドのプランナに抽出する:$-\sum_{m,i} \hat{S}_i^m \log(S_i^m) + (1 - \hat{S}_i^m) \log(1 - S_i^m)$.(10) 軌跡$T_i$に対して、各サブスコアの蒸留損失は式4の学習コスト値として働き、そのメトリックに関連する特定の交通ルールの違反を測定する。

2.4. 推論と後処理

2.4.1 Inference

予測された模倣スコア$\{S_i^{im} \mid i = 1, \cdots, k\}$とメトリックサブスコア$\{S_i^m \mid i = 1, \cdots, k\}_{m=1}^{|M|}$ が与えられたとき、与えられたシナリオで各軌跡が選択される可能性を測定する組立コストを以下のように計算する:

$$\tilde{f}(T_i, O) = -\left(w_1 \log \mathcal{S}_i^{im} + w_2 \log \mathcal{S}_i^{NC} + w_3 \log \mathcal{S}_i^{DAC} + w_4 \log\left(5\mathcal{S}_i^{TTC} + 2\mathcal{S}_i^{C} + 5\mathcal{S}_i^{EP}\right)\right), \quad (11)$$

ここで$\{w_i\}_{i=1}^{4}$は、異なる教師の不完全なフィッティングを緩和するための信頼度重み付けパラメータを表す。最適な重みの組み合わせはグリッド探索によって得られ、通常以下の範囲に収まる: $0.01 \leqq w_1 \leqq 0.1$, $0.1 \leqq w_2$, $w_3 \leqq 1$, $1 \leqq w_4 \leqq 10$, 模倣よりもルールベースのコストを優先する必要性を示す。最後に、全体的なコストが最も低い軌道が選択される。

2.4.2 モデルの組み立て

我々は2つのモデルアンサンブル技術を紹介する: エンコーダの混合とサブスコアアンサンブルである。前者は異なるビジョンエンコーダからの特徴を結合するために線形層を使用し、後者は軌跡選択のための独立したモデルからの部分スコアの加重和を計算する。

3. Experiments

3.1. データセットと測定基準

データセット Navsimデータセットは、既存のOpenScene [7]データセットをベースにしており、

2.2. Overall Framework

As shown in Fig. 2, Hydra-MDP consists of two networks: a **Perception Network** and a **Trajectory Decoder**.

Perception Network. Our perception network builds upon the official challenge baseline Transfuser [5, 6], which consists of an image backbone, a LiDAR backbone, and perception heads for 3D object detection and BEV segmentation. Multiple transformer layers [19] connect features from stages of both backbones, extracting meaningful information from different modalities. The final output of the perception network comprises environmental tokens F_{env}, which encode abundant semantic information derived from both images and LiDAR point clouds.

Trajectory Decoder. Following Vadv2 [4], we construct a fixed planning vocabulary to discretize the continuous action space. To build the vocabulary, we first sample 700K trajectories randomly from the original nuPlan database [2]. Each trajectory $T_i(i = 1, ..., k)$ consists of 40 timestamps of $(x, y, heading)$, corresponding to the desired 10Hz frequency and a 4-second future horizon in the challenge. The planning vocabulary $\mathcal{V}_k$ is formed as K-means clustering centers of the 700K trajectories, where k denotes the size of the vocabulary. $\mathcal{V}_k$ is then embedded as k latent queries with an MLP, sent into layers of transformer encoders [19], and added to the ego status E:

$$\mathcal{V}'_k = Transformer(Q, K, V = Mlp(\mathcal{V}_k)) + E. \quad (6)$$

To incorporate environmental clues in F_{env}, transformer decoders are leveraged:

$$\mathcal{V}''_k = Transformer(Q = \mathcal{V}'_k, K, V = F_{env}). \quad (7)$$

Using the log-replay trajectory $\hat{T}$, we implement a distance-based cross-entropy loss to imitate human drivers:

$$\mathcal{L}_{im} = -\sum_{i=1}^{k} y_i \log(\mathcal{S}_i^{im}), \quad (8)$$

where $\mathcal{S}_i^{im}$ is the i-th softmax score of $\mathcal{V}''_k$, and y_i is the imitation target produced by L2 distances between log-replays and the vocabulary. Softmax is applied on L2 distances to produce a probability distribution:

$$y_i = \frac{e^{-(\hat{T}-T_i)^2}}{\sum_{j=1}^{k} e^{-(\hat{T}-T_j)^2}}. \quad (9)$$

The intuition behind this imitation target is to reward trajectory proposals that are close to human driving behaviors.

2.3. Multi-target Hydra-Distillation

Though the imitation target provides certain clues for the planner, it is insufficient for the model to associate the planning decision with the driving environment under the closed-loop setting, leading to failures such as collisions and leaving drivable areas [14]. Therefore, to boost the closed-loop performance of our end-to-end planner, we propose Multi-target Hydra-Distillation, a learning strategy that aligns the planner with simulation-based metrics in this challenge.

The distillation process expands the learning target through two steps: (1) running offline simulations [8] of the planning vocabulary $\mathcal{V}_k$ for the entire training dataset; (2) introducing supervision from simulation scores for each trajectory in $\mathcal{V}_k$ during the training process. For a given scenario, step 1 generates ground truth simulation scores $\{\hat{\mathcal{S}}_i^m | i = 1, ..., k\}_{m=1}^{|M|}$ for each metric $m \in M$ and the i-th trajectory, where M represents the set of closed-loop metrics used in the challenge. For score predictions, latent vectors $\mathcal{V}''_k$ are processed with a set of Hydra Prediction Heads, yielding predicted scores $\{\mathcal{S}_i^m | i = 1, ..., k\}_{m=1}^{|M|}$. With a binary cross-entropy loss, we distill rule-based driving knowledge into the end-to-end planner:

$$\mathcal{L}_{kd} = -\sum_{m,i} \hat{\mathcal{S}}_i^m \log \mathcal{S}_i^m + (1 - \hat{\mathcal{S}}_i^m) \log(1 - \mathcal{S}_i^m). \quad (10)$$

For a trajectory T_i, its distillation loss of each sub-score acts as a learned cost value in Eq. 4, measuring the violation of particular traffic rules associated with that metric.

2.4. Inference and Post-processing

2.4.1 Inference

Given the predicted imitation scores $\{\mathcal{S}_i^{im} | i = 1, ..., k\}$ and metric sub-scores $\{\mathcal{S}_i^m | i = 1, ..., k\}_{m=1}^{|M|}$, we calculate an assembled cost measuring the likelihood of each trajectory being selected in the given scenario as follows:

$$\begin{aligned}\tilde{f}(T_i, O) = -\,(&w_1 \log \mathcal{S}_i^{im} + w_2 \log \mathcal{S}_i^{NC} + w_3 \log \mathcal{S}_i^{DAC} \\ &+ w_4 \log (5\mathcal{S}_i^{TTC} + 2\mathcal{S}_i^{C} + 5\mathcal{S}_i^{EP})),\end{aligned} \quad (11)$$

where $\{w_i\}_{i=1}^{4}$ represent confidence weighting parameters to mitigate the imperfect fitting of different teachers. The optimal combination of weights is obtained via grid search, which typically fall within the following ranges: $0.01 \leq w_1 \leq 0.1, 0.1 \leq w_2, w_3 \leq 1, 1 \leq w_4 \leq 10$, indicating the necessity to prioritize rule-based costs over imitation. Finally, the trajectory with the lowest overall cost is chosen.

2.4.2 Model Ensembling

We present two model ensembling techniques: Mixture of Encoders and Sub-score Ensembling. The former technique uses a linear layer to combine features from different vision encoders, while the latter calculates a weighted sum of sub-scores from independent models for trajectory selection.

3. Experiments

3.1. Dataset and metrics

Dataset. The Navsim dataset builds on the existing OpenScene [7] dataset, a compact version of nuPlan [3] with only

Method	Inputs	NC	DAC	EP	TTC	C	Score
PDM-Closed [8]◇	Perception GT	94.6	99.8	89.9	86.9	99.9	89.1
Transfuser [5]	LiDAR & Camera	96.5	87.9	73.9	90.2	100	78.0
Vadv2-$\mathcal{V}_{4096}$ [4]*	LiDAR & Camera	97.1	88.8	74.9	91.4	100	79.7
Vadv2-$\mathcal{V}_{4096}$ [4]*-PP	LiDAR & Camera	97.0	89.1	75.0	91.2	100	79.9
Vadv2-$\mathcal{V}_{8192}$ [4]*	LiDAR & Camera	97.2	89.1	76.0	91.6	100	80.9
Hydra-MDP-$\mathcal{V}_{4096}$	LiDAR & Camera	97.7	91.5	77.5	92.7	100	82.6
Hydra-MDP-$\mathcal{V}_{8192}$	LiDAR & Camera	97.9	91.7	77.6	92.9	100	83.0
Hydra-MDP-$\mathcal{V}_{8192}$-PDM	LiDAR & Camera	97.5	88.9	74.8	92.5	100	80.2
Hydra-MDP-$\mathcal{V}_{8192}$-W	LiDAR & Camera	98.1	**96.1**	77.8	93.9	100	85.7
Hydra-MDP-$\mathcal{V}_{8192}$-W-EP	LiDAR & Camera	**98.3**	96.0	**78.7**	**94.6**	**100**	**86.5**

表1. Navtest Splitでの性能。PDM-Closedの公式Navsim実装は、nuPlan実装と比較して、ブレーキ操作やオフセットの定式化に一貫性がないため、エラーが発生しやすい可能性があります[8]。すべてのエンドツーエンドの手法は、公式のTransfuser [5]を知覚ネットワークとして使用している。* 距離に基づく模倣損失を学習に採用した。PP: 後処理にトランスフォーマー知覚を用いる。PDM:学習目標はPDM総合スコアである。W: 推論中の重み付き信頼度。EP:モデルは連続EP(Ego Progress)メトリックに適合するように学習される。

Method	Img. Resolution	Backbone	NC	DAC	EP	TTC	C	Score
PDM-Closed [8]◇	-	-	94.6	**99.8**	**89.9**	86.9	99.9	89.1
Hydra-MDP-A	256 × 1024	ViT-L*	98.4	97.7	85.0	94.5	100	89.9
Hydra-MDP-B	512 × 2048	V2-99	98.4	97.8	86.5	93.9	100	90.3
Hydra-MDP-C	256 × 1024 256 × 1024 512 × 2048	ViT-L* ViT-L† V2-99	**98.7**	98.2	86.5	**95.0**	**100**	**91.0**

表2. スケールアップがNavtestスプリットに与える影響。PDM-Closedの公式Navsim実装。* ViT-LはDepth Anything [20]から初期化。ViT-LはObjects365[17]とCOCO[15]で事前学習されたEVA[9]である。V2-99[13]はDD3D[16]から初期化される。

nuPlan [3]のコンパクトなバージョンで、関連するアノテーションと2Hzでサンプリングされたセンサーデータのみである。データセットは主に、自車両の履歴データを将来の計画に外挿できないような、意図の変化を伴うシナリオに焦点を当てている。このデータセットは、オブジェクトの意味カテゴリと3Dバウンディングボックスを持つ注釈付き2D高精細マップを提供する。データセットは2つのパートに分かれている: NavtrainとNavtestは、それぞれ1192と136のシナリオを含む。

メトリクス。この課題に対して、我々はPDMスコアに基づいてモデルを評価する:

$$PDM_{score} = NC \times DAC \times DDC \times \frac{(5\times TTC+2\times C+5\times EP)}{12}, \quad (12)$$

ここで、サブメトリクスN C、DAC、T T C、C、EPは、故障なし衝突、走行可能領域コンプライアンス、衝突までの時間、快適さ、自我の進行に対応する。蒸留プロセスとその後の結果については、実装上の問題からDDCは無視される[1] 。

3.2. 実装の詳細

NVIDIA A100 GPU 8 台を使用し、20 エポックで合計 256 バッチサイズで Navtrain 分割でモデルを学習する。学習率と重み減衰は公式ベースラインに従って1×10^{-4}と0.0に設定される。4フレームからのLiDAR点をBEV平面上にスプラットして密度BEV特徴を形成し、ResNet34 [10]を用いて符号化する。画像の場合、フロントビュー画像は中央で切り取られたフロントレフトビュー画像とフロントライトビュー画像を連結され、デフォルトで256×1024の入力解像度が得られる。

[1]https://github.com/autonomousvision/navsim/issues/14

ResNet34 は特に指定がない限り、特徴抽出にも適用される。データやテスト時間の拡張は使用しない。

3.3. Main Results

表1に示すように、我々の結果は、Hydra-MDPの絶対的な優位性を強調している。1に示すように、Hydra-MDPのベースラインに対する絶対的な優位性が強調されている。異なる計画語彙[4]の探索において、より大きな語彙V_{8192}を利用することで、異なる手法間で改善が見られる。さらに、微分不可能な後処理は、我々のフレームワークよりも性能向上が少ないが、重み付き信頼度は、包括的に性能を向上させる。異なる学習目標の効果を除去するために、初期の実験では連続的な指標EP(Ego Progress)は考慮されず、PDMスコア全体の蒸留を試みる。それにもかかわらず、PDMスコアの不規則な分布は性能劣化を引き起こし、これは我々のマルチターゲット学習パラダイムの必要性を示唆している。Hydra-MDP-V_{8192} -W-EPの最終版では、EPの蒸留によって対応するメトリックを改善することができる。

3.4. スケールアップとモデルアンサンブル

先行文献[11]では、バックボーンが大きいと、計画性能のわずかな改善にしかつながらないことが示唆されている。とはいえ、より大きなバックボーンで我々のモデルのスケーラビリティをさらに実証する。表 2は、ViT-L[9, 20]とV2-99[13]を画像バックボーンとして、Hydra-MDPの3つの最良性能バージョンを示す。最終的な提出には、これら3つのモデルのアンサンブルされたサブスコアを推論に使用する。

Method	Inputs	NC	DAC	EP	TTC	C	Score
PDM-Closed [8]◇	Perception GT	94.6	99.8	89.9	86.9	99.9	89.1
Transfuser [5]	LiDAR & Camera	96.5	87.9	73.9	90.2	100	78.0
Vadv2-$\mathcal{V}_{4096}$ [4]*	LiDAR & Camera	97.1	88.8	74.9	91.4	100	79.7
Vadv2-$\mathcal{V}_{4096}$ [4]*-PP	LiDAR & Camera	97.0	89.1	75.0	91.2	100	79.9
Vadv2-$\mathcal{V}_{8192}$ [4]*	LiDAR & Camera	97.2	89.1	76.0	91.6	100	80.9
Hydra-MDP-$\mathcal{V}_{4096}$	LiDAR & Camera	97.7	91.5	77.5	92.7	100	82.6
Hydra-MDP-$\mathcal{V}_{8192}$	LiDAR & Camera	97.9	91.7	77.6	92.9	100	83.0
Hydra-MDP-$\mathcal{V}_{8192}$-PDM	LiDAR & Camera	97.5	88.9	74.8	92.5	100	80.2
Hydra-MDP-$\mathcal{V}_{8192}$-W	LiDAR & Camera	98.1	**96.1**	77.8	93.9	100	85.7
Hydra-MDP-$\mathcal{V}_{8192}$-W-EP	LiDAR & Camera	**98.3**	96.0	**78.7**	**94.6**	**100**	**86.5**

Table 1. **Performance on the Navtest Split.** ◇ The official Navsim implementation of PDM-Closed is potentially prone to errors due to inconsistent braking maneuvers and offset formulation compared with the nuPlan implementation [8]. All end-to-end methods use the official Transfuser [5] as the perception network. * Our distance-based imitation loss is adopted for training. PP: Transfuser perception is used for post-processing. PDM: The learning target is the overall PDM score. W: Weighted confidence during inference. EP: The model is trained to fit the continuous EP (Ego Progress) metric.

Method	Img. Resolution	Backbone	NC	DAC	EP	TTC	C	Score
PDM-Closed [8]◇	-	-	94.6	**99.8**	**89.9**	86.9	99.9	89.1
Hydra-MDP-A	256 × 1024	ViT-L*	98.4	97.7	85.0	94.5	100	89.9
Hydra-MDP-B	512 × 2048	V2-99	98.4	97.8	86.5	93.9	100	90.3
Hydra-MDP-C	256 × 1024 256 × 1024 512 × 2048	ViT-L* ViT-L† V2-99	**98.7**	98.2	86.5	**95.0**	**100**	**91.0**

Table 2. **The Impact of Scaling Up on the Navtest Split.** ◇ The official Navsim implementation of PDM-Closed. * ViT-L is initialized from Depth Anything [20]. †ViT-L is EVA [9] pretrained on Objects365 [17] and COCO [15]. V2-99 [13] is initialized from DD3D [16].

relevant annotations and sensor data sampled at 2 Hz. The dataset primarily focuses on scenarios involving changes in intention, where the ego vehicle's historical data cannot be extrapolated into a future plan. The dataset provides annotated 2D high-definition maps with semantic categories and 3D bounding boxes for objects. The dataset is split into two parts: Navtrain and Navtest, which respectively contain 1192 and 136 scenarios for training/validation and testing.

Metrics. For this challenge, we evaluate our models based on the PDM score, which can be formulated as follows:

$$PDM_{score} = NC \times DAC \times DDC \times \frac{(5\times TTC+2\times C+5\times EP)}{12}, \quad (12)$$

where sub-metrics NC, DAC, TTC, C, EP correspond to the No at-fault Collisions, Drivable Area Compliance, Time to Collision, Comfort, and Ego Progress. For the distillation process and subsequent results, DDC is neglected due to an implementation problem.[1].

3.2. Implementation Details

We train our models on the Navtrain split using 8 NVIDIA A100 GPUs, with a total batch size of 256 across 20 epochs. The learning rate and weight decay are set to 1×10^{-4} and 0.0 following the official baseline. LiDAR points from 4 frames are splatted onto the BEV plane to form a density BEV feature, which is encoded using ResNet34 [10]. For images, the front-view image is concatenated with the center-cropped front-left-view and front-right-view images, yielding an input resolution of 256 × 1024 by default. ResNet34 is also applied for feature extraction unless otherwise specified. No data or test-time augmentations are used.

[1]https://github.com/autonomousvision/navsim/issues/14

3.3. Main Results

Our results, presented in Tab. 1, highlight the absolute advantage of Hydra-MDP over the baseline. In our exploration of different planning vocabularies [4], utilizing a larger vocabulary $\mathcal{V}_{8192}$ demonstrates improvements across different methods. Furthermore, non-differentiable post-processing yields fewer performance gains than our framework, while weighted confidence enhances the performance comprehensively. To ablate the effect of different learning targets, the continuous metric EP (Ego Progress) is not considered in early experiments and we attempt the distillation of the overall PDM score. Nonetheless, the irregular distribution of the PDM score incurs performance degradation, which suggests the necessity of our multi-target learning paradigm. In the final version of Hydra-MDP-$\mathcal{V}_{8192}$-W-EP, the distillation of EP can improve the corresponding metric.

3.4. Scaling Up and Model Ensembling

Previous literature [11] suggests larger backbones only lead to minor improvements in planning performance. Nevertheless, we further demonstrate the scalability of our model with larger backbones. Tab. 2 shows three best-performing versions of Hydra-MDP with ViT-L [9, 20] and V2-99 [13] as the image backbone. For the final submission, we use the ensembled sub-scores of these three models for inference.

References

[1] Sourav Biswas, Sergio Casas, Quinlan Sykora, Ben Agro, Abbas Sadat, and Raquel Urtasun. Quad: Query-based interpretable neural motion planning for autonomous driving. *arXiv preprint arXiv:2404.01486*, 2024. 2

[2] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 3

[3] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 3

[4] 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, 2, 3, 4

[5] 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*, 2022. 3, 4

[6] NAVSIM Contributors. Navsim: Data-driven non-reactive autonomous vehicle simulation. https://github.com/autonomousvision/navsim, 2024. 3

[7] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023. 3

[8] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pages 1268–1281. PMLR, 2023. 1, 3, 4

[9] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[11] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1, 2, 4

[12] 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, 2

[13] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In コンピュータビジョンとパターン認識ワークショップに関するIEEE/CVF会議予稿集，ページ0-0，2019．4

[14] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? *arXiv preprint arXiv:2312.03031*, 2023. 1, 2, 3

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4

[16] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 4

[17] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 4

[18] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 1

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[20] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024. 4

References

[1] Sourav Biswas, Sergio Casas, Quinlan Sykora, Ben Agro, Abbas Sadat, and Raquel Urtasun. Quad: Query-based interpretable neural motion planning for autonomous driving. *arXiv preprint arXiv:2404.01486*, 2024. 2

[2] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 3

[3] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 3

[4] 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, 2, 3, 4

[5] 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*, 2022. 3, 4

[6] NAVSIM Contributors. Navsim: Data-driven non-reactive autonomous vehicle simulation. https://github.com/autonomousvision/navsim, 2024. 3

[7] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023. 3

[8] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pages 1268–1281. PMLR, 2023. 1, 3, 4

[9] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[11] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1, 2, 4

[12] 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, 2

[13] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 4

[14] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? *arXiv preprint arXiv:2312.03031*, 2023. 1, 2, 3

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 4

[16] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 4

[17] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 4

[18] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 1

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[20] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024. 4