FastB2D: TOWARD COST-EFFICIENT CLOSED-LOOP BENCHMARK FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

We present FastB2D, a cost-efficient closed-loop benchmark proxy for autonomous driving (AD) development, built on the widely adopted yet costly Bench2Drive. FastB2D reduces the route count by 65% and total evaluation time by over 70%, while estimating Success Rate with a mean absolute error of $\sim$1.2% and a Spearman correlation of 0.9+ on held-out advanced AD model variants, and accurately recovering Driving Score, Driving Efficiency, and Driving Smoothness. FastB2D is constructed by addressing two structural inefficiencies we identify in closed-loop AD benchmarks: statistical route redundancy and highly imbalanced per-route evaluation cost, where a small number of long routes can dominate total testing time. Our cost-aware benchmark compression method learns compact route representations directly from model evaluation records and selects representative low-cost anchor routes through cost-aware clustering. Compared with existing general-purpose benchmark compression methods and compact AD benchmarks, FastB2D further reduces evaluation time by 27.2% on average, lowers performance-estimation error by 64% on average, and demonstrates stronger out-of-distribution generalization. All code and data will be released.

1 INTRODUCTION

Testing is central to autonomous driving (AD), because an AD model must be evaluated not only by how accurately it predicts logged data, but also by how safely and reliably it acts after its own decisions affect the environment (Caesar et al., 2020; Sun et al., 2020; Wilson et al., 2023). Closed-loop testing has therefore become a critical protocol for AD evaluation. Unlike open-loop testing, which usually performs single-pass inference on fixed logged inputs (Dosovitskiy et al., 2017; Jia et al., 2024; linggm3, 2026; Caesar et al., 2021; Gulino et al., 2023; Dauner et al., 2024; Li et al., 2026), closed-loop testing requires a model to drive through routes while continuously interacting with the simulator. At each step, the model observes the scene, predicts an action, changes the future environment state, and then encounters the consequences of its own behavior. This protocol better reflects deployment, but it also makes evaluation expensive.

In practice, the cost of closed-loop testing is not determined by route count alone. Closed-loop benchmarks often intrinsically exhibit both statistical redundancy and substantial cost imbalance. To make this concrete, we study Bench2Drive (Jia et al., 2024), a widely adopted closed-loop benchmark on which most advanced AD models are now evaluated (Chitta et al., 2023; Hu et al., 2023b; Jiang et al., 2023; Jia et al., 2025; Renz et al., 2025; Yang et al., 2026; Tang et al., 2025). Its broad coverage of route types and traffic interactions makes it a natural testbed for analyzing benchmark inefficiency. First, routes are **statistically redundant**. Closed-loop driving repeatedly instantiates common interaction patterns, so different routes can induce similar response patterns across models and provide overlapping measurement information. Second, routes have highly **imbalanced cost**. Since closed-loop evaluation unfolds through long-horizon interaction, route cost depends on how long the simulator must respond to model actions. As a result, a small number of long and interaction-rich routes can dominate the total evaluation time. Figure 1 summarizes these findings on Bench2Drive. In Figure 1(a), MAE measures score estimation error and SP measures rank preservation. Many partial route sets still approximate the full benchmark well, indicating substantial redundancy. In Figure 1(b), per-route frame counts are highly skewed, and the top 20% most expensive routes account for nearly half of the total evaluation cost. These findings show that closed-loop benchmark compression is not

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.

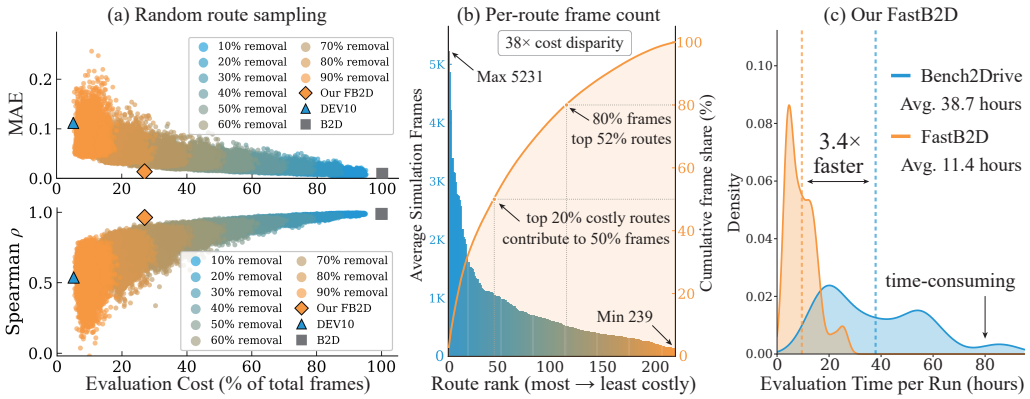

Figure 1: **Route redundancy, cost imbalance, and the FastB2D solution. (a)** Random route sampling on Bench2Drive (B2D) reveals substantial redundancy. For each removal ratio from 10% to 90%, we sample 10,000 retained subsets and evaluate them on 50+ AD model settings. Each subset is compared with the full 220-route benchmark using mean absolute error and Spearman rank correlation. Many subsets remain close to the full benchmark, although stability declines as the removal ratio increases. **(b)** Per-route frame counts for all 220 Bench2Drive routes are highly skewed. The top 20% most expensive routes account for nearly half of the total evaluation cost, showing that route count alone is an incomplete proxy for testing time. **(c)** FastB2D selects low-cost representative anchor routes. By leveraging both statistical redundancy and cost imbalance, it reduces evaluation time while preserving strong agreement with the full benchmark. See more details in Appendix D.

only feasible but also highly valuable, because it can substantially accelerate AD model improvement and analysis, but they suggest that route count is an incomplete proxy for evaluation cost. This motivates the central question of this work:

How can we build a compact, faithful, and cost-efficient closed-loop benchmark for AD?

Naive compression does not fully resolve this challenge. Random sampling treats routes as exchangeable (Huang et al., 2025; Polo et al., 2024) and ignores both redundancy in measurement information and differences in route cost. As Figure 1(a) shows, many random subsets can approximate the full benchmark, but this happens largely by chance, their quality varies widely, and their reliability declines as the removal ratio increases. Expert semantic selection offers a complementary perspective. Compact expert-designed subsets of Bench2Drive, such as DEV10 (Jia et al., 2025), provide intuitive scenario coverage and support lightweight evaluation. Their selection criterion is based on scenario labels rather than on how routes actually distinguish AD models in closed-loop testing. Because route outcomes depend jointly on road geometry, traffic-agent interactions, and simulator stochasticity, routes with similar labels may probe different model behaviors, while routes from different categories may induce correlated responses. Semantic subsets therefore preserve the measurement structure of the full benchmark less consistently than response-based selection, as shown in Section 3.

Together, these observations suggest that an efficient closed-loop benchmark should do more than remove routes. It should retain routes that are statistically representative of the full benchmark and, among redundant candidates, favor cheaper ones. To implement this principle, we propose a response-based and cost-aware Benchmark Compression Method (BCM) (Zhang et al., 2026; Huang et al., 2025; Polo et al., 2024; Vivek et al., 2024) for closed-loop AD testing. Rather than relying on route metadata or expert grouping, BCM learns route representations directly from model evaluation records, so each route is described by how a collection of AD models responds to it. Routes with similar response patterns are grouped in the learned space, and one low-cost anchor route is selected from each group. In this way, our method captures statistical redundancy through response-based representation learning and exploits cost imbalance through cost-aware clustering.

Applied to Bench2Drive, this framework yields FastB2D, a compact benchmark of 77 anchor routes that removes 65% of the original routes and cuts total evaluation time by over 70%. On existing held-out AD model variants, FastB2D estimates Success Rate with an MAE of $\sim$1.2% and a Spearman correlation of over 0.9, and it accurately recovers Driving Score, Driving Efficiency, and Driving

Smoothness. Against existing compact subsets and general-purpose BCMs, FastB2D achieves better out-of-distribution generalization at a smaller evaluation budget.

We show related works in Appendix B and summarize our contributions as follows:

- We identify two structural sources of inefficiency in closed-loop AD benchmarks: statistical route redundancy and highly imbalanced per-route evaluation cost. We show that evaluation cost is not determined by route count alone, and that neither random sampling nor expert semantic compression can reliably preserve the measurement structure of the full benchmark.
- We present FastB2D, a cost-efficient closed-loop benchmark proxy for quick development of AD, instantiated on well-known Bench2Drive. It is built with a response-based and cost-aware BCM that learns route representations from model evaluation records and selects low-cost anchor routes through clustering. FastB2D reduces evaluation time by over 70%, and estimates Success Rate with $\sim$1.2% MAE and over 0.9 SP, while also accurately estimating other key metrics.

2 FastB2D: Construction and Evaluation

This section presents cost-efficient closed-loop benchmark FastB2D. Starting from full Bench2Drive evaluation records, FastB2D first learns response-based route representations, then selects low-cost anchor routes by cost-aware clustering. A new AD model is evaluated only on these anchors, and its full-benchmark performance is estimated from the observed anchor outcomes.

2.1 Response-Based Route Representation

Consider an AD closed-loop benchmark with N routes $\mathcal{R} = \{r_1, \dots, r_N\}$ and per-route average cost $t_j > 0$ measured in simulation frames or wall-clock time. Suppose M models $\mathcal{M} = \{m_1, \dots, m_M\}$ have been evaluated on every route, yielding a binary response matrix $\mathbf{X} \in \{0,1\}^{M \times N}$ in which $x_{ij} = 1$ when model m_i succeeds on route r_j. Following logistic response models from psychometric testing (Lalor et al., 2016; Polo et al., 2024), we assign each model m_i a latent ability vector $\boldsymbol{\theta}_i \in \mathbb{R}^D$, and each route r_j a discrimination vector $\mathbf{a}_j \in \mathbb{R}^D$ together with a bias $b_j \in \mathbb{R}$. The probability that m_i succeeds on r_j is $P(x_{ij} = 1 \mid \boldsymbol{\theta}_i, \mathbf{a}_j, b_j) = \sigma(\mathbf{a}_j^\top \boldsymbol{\theta}_i - b_j)$, where σ is the sigmoid. All learnable parameters are fit by minimizing the negative log-likelihood over observed responses,

$$\mathcal{L}\big(\{\boldsymbol{\theta}_i\}_{i=1}^M, \{\mathbf{a}_j, b_j\}_{j=1}^N\big) = -\sum_{i=1}^M \sum_{j=1}^N \Big[x_{ij} \log P_{ij} + (1 - x_{ij}) \log\big(1 - P_{ij}\big)\Big], \qquad (1)$$

with $P_{ij} = P(x_{ij} = 1)$. Each route r_j is then represented by $\mathbf{v}_j = [\mathbf{a}_j; b_j] \in \mathbb{R}^{D+1}$, so that routes inducing similar response patterns lie close together in this space.

2.2 Cost-Aware Anchor Selection

Given the learned representations, we cluster routes with similar behavior and pick one anchor per cluster. K-Means (Hamerly & Elkan, 2003) on $\{\mathbf{v}_1, \dots, \mathbf{v}_N\}$ produces K clusters $\{\mathcal{C}_1, \dots, \mathcal{C}_K\}$ with centroids $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$, where $K = \lfloor (1-p)N \rfloor$ for target removal ratio p. A central anchor is representative but may be costly, while the cheapest route in a cluster may sit near its boundary and underrepresent the cluster. We therefore select final route set $\mathcal{A}$ by trading off representativeness against cost,

$$\mathcal{A} = \left\{ \arg\min_{r_j \in \mathcal{C}_k} \Big[\alpha \tilde{d}(r_j, \boldsymbol{\mu}_k) + (1-\alpha)\tilde{t}(r_j)\Big] \;\Big|\; k = 1, \dots, K \right\}, \qquad (2)$$

where $\tilde{d}(r_j, \boldsymbol{\mu}_k)$ is the min-max normalized distance $\|\mathbf{v}_j - \boldsymbol{\mu}_k\|_2$ within $\mathcal{C}_k$, $\tilde{t}(r_j)$ is the min-max normalized cost in the same cluster, and $\alpha \in [0,1]$ controls the trade-off. When $\alpha = 1$, anchor selection returns the most central route. When $\alpha = 0$, it returns the cheapest. The selected anchors preserve measurement diversity while suppressing redundant high-cost routes.

Both the representation learning and the anchor selection are built on Success Rate, which directly reflects whether a model can complete a route safely and is the most widely adopted metric in closed-loop AD evaluation (Appendix D.3). As we show in Section 3, anchors selected on Success Rate also generalize well to Driving Score, Driving Efficiency, and Driving Smoothness.

Table 1: Bench2Drive (B2D) vs. FastB2D (FB2D) statistics across five driving skills. Parenthesized green values denote scenario retention; orange values denote reduction from B2D. The Overall row reports unique counts. Total Time is the average wall-clock time across all evaluated models.

Skill	# Scenarios		# Routes		Total Frames		Total Time (h)	
	B2D	FB2D	B2D	FB2D	B2D	FB2D	B2D	FB2D
Merging	16	16 (100%)	80	30 (↓62%)	66,906	19,353 (↓71.1%)	13.7	4.0 (↓70.7%)
Traffic Sign	19	17 (89%)	95	35 (↓63%)	74,323	19,135 (↓74.3%)	17.2	4.9 (↓71.8%)
Emergency Brake	12	9 (75%)	60	17 (↓72%)	47,926	10,971 (↓77.1%)	10.8	2.9 (↓72.8%)
Overtaking	9	9 (100%)	45	18 (↓60%)	44,988	15,851 (↓64.8%)	10.7	3.7 (↓65.1%)
Give Way	2	2 (100%)	10	4 (↓60%)	1,755	632 (↓64.0%)	0.4	0.1 (↓64.8%)
Overall (unique)	44	40 (91%)	220	77 (↓65%)	175,927	49,357 (↓71.9%)	38.7	11.4 (↓70.5%)

Table 2: Full Bench2Drive (B2D) evaluations versus FastB2D (FB2D) predictions for several representative models across four metrics. The bottom row reports mean absolute error (MAE) for Success Rate, Driving Score, and Smoothness, and mean relative error (MRE) for Efficiency.

Model	Success Rate (%)		Driving Score		Efficiency		Smoothness	
	B2D	FB2D	B2D	FB2D	B2D	FB2D	B2D	FB2D
SimLingo (Renz et al., 2025)	66.4±0.3	67.0±3.4	85.9±0.1	86.2±0.5	252.7±0.5	252.9±1.2	0.32±0.01	0.32±0.02
CriticVLA (Yang et al., 2026)	71.3±2.8	72.8±3.9	86.8±0.7	86.9±2.3	259.9±0.7	259.0±0.1	0.20±0.01	0.22±0.01
LEAD_{WP} (Nguyen et al., 2025)	79.5±0.9	79.8±0.6	91.8±0.4	90.2±0.8	207.3±1.5	209.2±1.5	0.19±0.00	0.22±0.00
LEAD_{B2D} (Nguyen et al., 2025)	89.5±1.3	88.4±1.2	96.3±0.3	95.6±0.6	210.7±5.5	213.5±5.8	0.20±0.01	0.20±0.01
$\text{BridgeDrive}_{\text{Lead}}$ (Liu et al., 2025)	90.5±1.3	88.9±1.4	97.1±0.7	97.1±0.2	209.8±2.5	210.6±0.3	0.20±0.01	0.22±0.03
DriveTransformer (Jia et al., 2025)	36.2±3.2	35.0±4.6	63.8±1.4	64.5±1.8	168.0±2.2	170.5±3.1	0.15±0.01	0.16±0.01
PDM-Lite (Dauner et al., 2023)	91.2±0.7	88.5±0.6	96.3±0.5	95.8±0.1	246.4±1.5	249.8±0.8	0.23±0.00	0.22±0.01
TransFuser++ (Chitta et al., 2023)	67.8±2.7	69.0±1.5	85.5±0.8	84.6±0.6	238.9±2.8	242.6±2.7	0.21±0.00	0.23±0.01
HiP-AD (Tang et al., 2025)	70.4±1.1	69.8±3.4	86.7±0.9	83.8±0.8	198.7±3.1	197.6±2.4	0.18±0.01	0.20±0.01
MAE/MRE (test)	1.20%		0.86%		0.89%		1.44%	
MAE/MRE (all test+val)	1.36%		0.76%		1.05%		1.49%	

2.3 The FastB2D Route Set

We collect and verify over 200 Bench2Drive evaluation records and partition them into three disjoint groups. The held-out models in Table 2 form the test set. A further 50 records form a validation (val) set used for the further analyses throughout the paper. See more construction details in Appendix C.

We use latent dimension $D = 10$ and removal ratio $p = 0.65$, yielding $K = 77$ anchors, and set $\alpha = 0.25$ to bias selection toward low-cost anchors while retaining representativeness as a constraint. Table 1 summarizes the resulting benchmark. FastB2D keeps 77 of the 220 routes and covers 40 of the 44 scenarios. Total simulation frames drop from 175,927 to 49,357, and average wall-clock evaluation time drops from 38.7 to 11.4 hours on an A100. Although FastB2D removes 65% of the routes, it cuts evaluation time by 70.5%, because the selected anchors avoid many long and costly routes. As shown in Figure 1, FastB2D anchors concentrate toward shorter routes, confirming that route count alone is an inadequate measure of evaluation cost. Selecting low-cost representatives within response-based clusters shortens evaluation while preserving the benchmark's measurement structure. More detailed analyses of hyperparameter are provided in Section 3.

2.4 Estimating Full-Benchmark Performance

A compressed benchmark should estimate full-benchmark performance from anchor outcomes alone. FastB2D employs two complementary strategies: predictive inference under the learned logistic model and calibrated weighted averaging. See detailed metric definitions in the Appendix D.

For Success Rate, let $\{x_j\}_{j\in\mathcal{A}}$ denote the anchor outcomes for a new model. We first infer its latent ability by maximum likelihood, $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}\{-\sum_{j\in\mathcal{A}}[x_j \log P_j(\boldsymbol{\theta}) + (1-x_j)\log(1-P_j(\boldsymbol{\theta}))]\}$, where $P_j(\boldsymbol{\theta}) = \sigma(\mathbf{a}_j^\top \boldsymbol{\theta} - b_j)$ and $\mathbf{a}_j, b_j$ are fixed after representation learning. We then combine observed anchor outcomes with predicted success probabilities on the remaining routes,

$$\hat{s}_{\text{pred}} = \frac{1}{N}\left(\sum\nolimits_{j\in\mathcal{A}} x_j + \sum\nolimits_{j\notin\mathcal{A}} \sigma(\mathbf{a}_j^\top \hat{\boldsymbol{\theta}} - b_j)\right). \tag{3}$$

Table 3: Per-skill predictions. Each skill score is the success rate over the routes belonging to that skill, and Mean Ability is the average of all five skill scores.

Model	Overtaking		Merging		Emerg. Brake		Give Way		Traffic Sign		Mean Ability	
	B2D	FB2D	B2D	FB2D	B2D	FB2D	B2D	FB2D	B2D	FB2D	B2D	FB2D
SimLingo (Renz et al., 2025)	66.7	67.2	52.5	54.5	80.8	82.5	50.0	50.1	60.5	63.7	62.1	63.6
CriticVLA (Yang et al., 2026)	72.2	72.9	60.0	60.2	80.8	87.3	50.0	50.1	65.3	71.2	65.7	68.3
LEAD_{WP} (Nguyen et al., 2025)	88.9	85.5	80.0	78.0	73.3	78.7	40.0	40.4	77.9	80.0	72.0	72.5
LEAD_{B2D} (Nguyen et al., 2025)	95.6	94.5	84.4	83.8	94.2	92.5	50.0	50.4	91.6	90.6	83.1	82.4
$\text{BridgeDrive}_{\text{Lead}}$ (Liu et al., 2025)	98.9	97.0	85.6	83.2	96.7	94.9	40.0	40.6	91.1	89.2	82.4	81.0
DriveTransformer (Jia et al., 2025)	31.1	37.9	38.8	43.4	31.7	42.9	20.0	29.0	22.1	28.3	28.7	36.3
PDM-Lite (Dauner et al., 2023)	93.3	91.9	85.0	78.0	96.7	96.1	70.0	60.2	90.5	85.4	87.1	82.3
TransFuser++ (Chitta et al., 2023)	52.6	54.6	57.5	59.7	83.9	84.4	40.0	40.1	74.4	73.5	61.7	62.5
HiP-AD (Tang et al., 2025)	70.4	70.6	56.2	57.7	81.7	88.3	50.0	50.1	66.7	71.7	65.0	67.7
MAE (test)	2.26%		2.51%		3.59%		1.94%		3.46%		2.39%	
MAE (all test+val)	3.32%		2.39%		3.51%		1.55%		2.48%		1.29%	

The first term uses actual anchor outcomes, and the second imputes outcomes on unobserved routes through the learned ability–response relationship rather than by simple averaging. This estimator produces route-level success probabilities, so it enables per-skill diagnosis as shown in Section 2.5.

For Driving Score, Driving Efficiency, and Driving Smoothness, we learn a calibrated weighted estimator. Let $X \in \mathbb{R}^{M\times K}$ contain anchor-level metric values for the training models and $\mathbf{y} \in \mathbb{R}^M$ their full-benchmark scores. We learn nonnegative anchor weights by solving

$$\min_{\mathbf{w},b} \|\mathbf{y} - X\mathbf{w} - b\mathbf{1}\|_2^2 + \lambda \sum_{j=1}^{K} w_j \log w_j, \quad \text{s.t.} \quad w_j \geq 0,\ \sum_{j=1}^{K} w_j = 1. \tag{4}$$

The simplex constraint makes the estimate a convex combination of anchor scores, and the entropy term prevents overreliance on any small subset of anchors. The strength λ is selected by cross-validation on the training models. Once trained, a new model only needs anchor evaluations.

To extend FastB2D's applicability as the AD model ecosystem evolves, we additionally train a calibrated SR estimator $\hat{s}_{\text{cal}}$ using the same framework in Eq. 4, with weights learned from anchor-level binary outcomes. At inference, FastB2D selects between them via a divergence gate, adopting $\hat{s}_{\text{pred}}$ when $|\hat{s}_{\text{pred}} - \hat{s}_{\text{cal}}| \leq \tau$ with $\tau = 0.05$ and $\hat{s}_{\text{cal}}$ otherwise. If two estimates are highly consistent within the training distribution, the gate defaults to predictive inference. When they diverge, it signals that the test model lies outside the observed population, where the calibrated estimator's linear structure generalizes more reliably. We provide a detailed analysis in Appendix C.3.

2.5 Validation on Held-Out Models

Table 2 compares full Bench2Drive scores with FastB2D predictions on held-out test models during training. Across Success Rate, Driving Score, Driving Efficiency, and Driving Smoothness, FastB2D closely tracks the full benchmark while requiring only 77 anchor routes.

Bench2Drive also evaluates models on five driving skills (Jia et al., 2024): Overtaking, Merging, Emergency Brake, Give Way, and Traffic Sign. Since FastB2D predicts route-level success probabilities, per-skill scores follow directly from the same predictions. As Table 3 shows, the predicted skill scores closely match the ground truth across all categories, confirming that FastB2D preserves not only aggregate scoring but also the skill-level diagnostic capacity of the full benchmark.

3 Analysis and Ablation Study

(1) How does the latent dimension D affect the compression quality?

The latent dimension D controls the capacity of route representations (Section 2.1). We sweep $D \in \{2, 5, 10, 15, 20, 30, 50, 75, 100\}$ and report MAE and Spearman correlation on the validation set. As shown in Figure 2(a), both metrics first improve and then degrade as D increases, with the best performance at $D = 10$. When D is too small, the representation cannot distinguish routes that test different driving skills, producing coarse clusters and less informative anchors. When D is too large, the per-route parameters exceed what the training data can reliably estimate, and the noisy representations degrade both clustering and prediction. We therefore adopt $D = 10$ as the default.

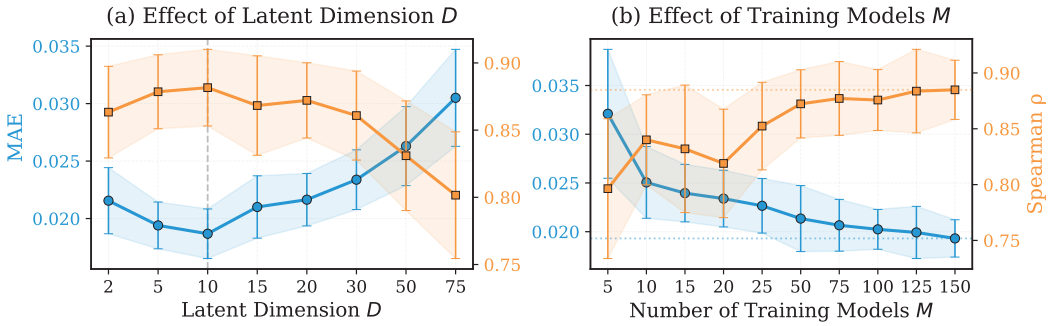

Figure 2: Analysis of key hyperparameters on val set. **(a)** Prediction accuracy (MAE and Spearman ρ) on the validation set as a function of the latent dimension D. Both metrics peak at $D = 10$. **(b)** Prediction accuracy (MAE and Spearman ρ) as a function of the number of training models M.

(2) How does the number of training models M affect the compression quality?

In Section 2.1, the number of M determines how many evaluation records are available for learning route representations. We vary $M \in \{5, 10, 15, 20, 25, 50, 75, 100, 125, 150\}$ by subsampling the training pool, re-run the full pipeline for each M, and keep training and test models strictly disjoint.

As shown in Figure 2(b), prediction accuracy improves steadily with M but with diminishing returns. The largest gains occur in the low-M regime: both Spearman ρ and MAE improve substantially as M increases from 5 to 50. Beyond this point, additional models contribute only marginal improvement, as the training records already capture sufficient response diversity. The variance across runs also decreases with M, reflecting more stable route representations from larger training pools. From a practical standpoint, even $M = 10$ achieves an MAE below 2.5%, suggesting that the compression framework is applicable early in a benchmark's lifecycle when few evaluated models are available. In constructing FastB2D, we use all $M = 158$ available training records. Although accuracy gains plateau beyond $M \approx 50$, the offline training cost is negligible regardless of M, so retaining the full pool yields the most stable route representations at no additional evaluation cost.

(3) How does the cost-awareness weight α affect evaluation cost and prediction accuracy?

The anchor selection in Eq. 2 uses cost-awareness weight α to balance representativeness and evaluation cost. When $\alpha = 1$, each anchor is the route closest to its cluster centroid, maximizing representativeness. When $\alpha = 0$, the cheapest route in each cluster is selected. We sweep $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and report evaluation cost against prediction error in Figure 3 (left).

As α increases, anchors become more representative of their clusters and prediction error decreases, but evaluation cost also rises because longer routes are more frequently selected. This trade-off exhibits clear diminishing returns: from $\alpha = 0$ to 0.25, a modest cost increase yields a notable accuracy gain, whereas further increasing α brings progressively smaller error reductions at substantially higher cost. We therefore adopt $\alpha = 0.25$ as the default, which keeps evaluation cost close to the minimum while recovering most of the accuracy of the pure representativeness setting ($\alpha = 1$).

(4) How does the number of anchor routes K affect the prediction accuracy?

The number of anchor routes K governs the trade-off between evaluation cost and prediction accuracy. We vary the removal ratio from 10% to 90% and compare FastB2D with random subsets of the same size. Figure 3 shows that FastB2D achieves lower prediction error and smaller variance at every ratio, with the largest advantage at high removal ratios, where random subsets become highly unstable.

As K increases, prediction error decreases but the gain quickly saturates. Beyond $K = 77$, corresponding to 65% removal, additional anchors bring only marginal accuracy improvement at much higher cost. To make this trade-off explicit, we define a Cost-Efficiency Score, abbreviated as CE, as the harmonic mean of normalized accuracy and cost reduction: $\mathrm{CE} = 2 \cdot \mathrm{Acc} \cdot \mathrm{Eff} \,/\, (\mathrm{Acc} + \mathrm{Eff})$, where $\mathrm{Acc} = 1 -$ normalized MAE and Eff is the fractional cost reduction relative to the full

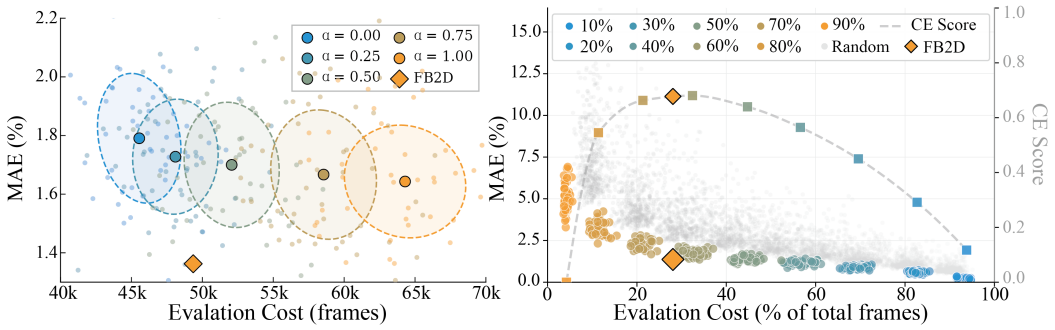

Figure 3: **Left:** Effect of α in Eq. 2 on evaluation cost and prediction error. Each point corresponds to one $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. **Right:** Prediction error vs. evaluation cost across nine removal ratios. Gray: random subsets. Colored: our BCM. Diamond: FastB2D. Dashed curve: CE Score.

Table 4: **(a)** Ablation of anchor selection and estimation method for DS, Eff., and Smooth. Rand.: random 77-route subsets; Unwtd.: unweighted mean; Learned: cross-validated weighted mean. **(b)** FastB2D prediction error vs. simulation variance on val set. All values in %.

Method	DS	Eff.	Smooth.
Rand. + Unwtd.	1.64±0.52	3.38±1.37	2.19±0.71
Rand. + Learned	1.23±0.13	1.37±0.26	1.68±0.34
Ours w/o Learned	1.48±0.31	3.89±1.76	4.71±1.35
Ours	**1.09**±0.11	**1.13**±0.20	**1.43**±0.28

(a) Anchor × estimation (mean error %).

	SR	DS	Eff.	Smooth.
Sim. Var. (1-seed)	2.02±0.79	1.07±0.31	1.00±0.45	0.84±0.20
FB2D Err. (1-seed)	1.54±1.03	0.75±0.65	0.98±0.99	1.25±0.87
Sim. Var. (3-seed)	0.84±0.31	0.44±0.13	0.42±0.20	0.35±0.10
FB2D Err. (3-seed)	0.91±0.65	0.45±0.34	0.70±0.65	0.90±0.79

(b) Prediction error vs. simulation variance.

benchmark. As the dashed curve on the right of Figure 3 shows, the CE Score peaks near $K = 77$, identifying it as the best balance between accuracy and efficiency. The CE Score also provides an explicit rule for choosing the compression ratio and can guide compression for other AD benchmarks. We therefore adopt $K = 77$ as the default. At this operating point, FastB2D delivers close to $3 - 4\times$ evaluation speedup. As Analysis (6) will further show, this saving makes three repeated evaluations still faster than a single full-benchmark pass while yielding more stable estimates.

(5) Do anchors optimized for Success Rate generalize to other evaluation metrics?

Our anchor selection in Section 2 is optimized for Success Rate. We next test whether the same anchors can also estimate the other three metrics: Driving Score, Efficiency, and Smoothness. We vary both the anchor set and the estimator, comparing random subsets with ours and a simple unweighted mean with a learned weighted mean. The unweighted estimator averages anchor-level values directly, whereas the learned estimator fits per-anchor weights using Eq. 4. The left subtable of Table 4 reports all four combinations.

Both components matter, and they are complementary. Learned weighting alone reduces error and variance under either anchor strategy, because it adjusts for the varying representativeness of individual routes. Under unweighted estimation, however, our anchors can even underperform random subsets on Efficiency and Smoothness, since anchors selected for binary SR discrimination need not match the full-benchmark mean on continuous metrics. Pairing our anchors with learned weighting removes this bias and yields the lowest error on all three metrics, confirming that response-based selection transfers well to continuous metrics once the estimator is properly calibrated. FastB2D therefore adopts both components for non-SR estimation. See Appendix D for further discussion.

(6) Is the prediction error of FastB2D acceptable?

Closed-loop simulation in CARLA is inherently stochastic, and evaluation results naturally fluctuate across different random seeds (Renz et al., 2025; Yang et al., 2026). A natural question is whether the prediction error introduced by FastB2D is acceptable relative to this intrinsic simulation variance. Table 4(b) compares the simulation variance (Sim. Var.) with FastB2D prediction error (FB2D Err.) under single-seed and three-seed averaged settings. Under single-seed evaluation, FastB2D prediction

Table 5: Comparison of BCMs on val set. Cost (%): percentage of total simulation frames retained. All error values are in %. CE Score is defined in Section 3; higher is better. Results are averaged over 10 runs. Best results per group are in **bold**. See Appendix E for more results.

Method	Cost (%) ↓	SR ↓	DS ↓	Eff. ↓	Smooth. ↓	Avg Error ↓	CE ↑
Removal ratio = 50% ($K = 110$)							
Random-Equal	50.0±3.3	2.47±0.90	1.20±0.39	2.68±1.67	1.73±0.70	2.02±0.59	0.54±0.08
AnchorPoint (Vivek et al., 2024)	62.3±0.8	1.67±0.14	5.84±0.21	1.36±0.41	5.00±0.61	3.47±1.97	0.08±0.12
SparseEval-Linear (Zhang et al., 2026)	61.1±1.3	2.80±0.68	5.17±0.33	1.93±0.97	4.48±0.84	3.59±1.29	0.00±0.28
SparseEval-MLP (Zhang et al., 2026)	60.3±1.2	1.57±0.12	5.15±0.40	2.05±1.05	3.65±0.88	3.11±1.41	0.25±0.13
TinyBenchmark (Polo et al., 2024)	51.9±1.9	2.31±0.50	1.53±0.35	1.62±0.86	1.69±0.83	1.79±0.31	0.56±0.05
FastB2D (Ours)	**43.9**±1.9	**1.31**±0.12	**0.76**±0.06	**0.77**±0.10	**0.92**±0.07	**0.94**±0.22	**0.72**±0.02
Removal ratio = 65% ($K = 77$)							
Random-Equal	35.0±3.3	3.38±1.18	1.63±0.50	3.56±2.11	2.36±0.96	2.73±0.78	0.56±0.15
AnchorPoint (Vivek et al., 2024)	42.0±1.2	2.77±0.10	6.17±0.51	3.36±1.00	4.12±0.43	4.11±1.29	0.00±0.22
SparseEval-Linear (Zhang et al., 2026)	41.8±2.0	2.31±0.42	5.49±0.79	2.65±1.76	3.98±0.96	3.61±1.25	0.27±0.23
SparseEval-MLP (Zhang et al., 2026)	41.1±1.2	1.90±0.12	5.43±0.90	5.79±1.30	2.84±0.67	3.99±1.66	0.08±0.27
TinyBenchmark (Polo et al., 2024)	37.3±1.3	3.30±0.90	1.75±0.34	2.84±1.40	2.45±0.94	2.59±0.57	0.58±0.10
FastB2D (Ours)	**27.1**±1.7	**1.67**±0.21	**0.97**±0.14	**1.19**±0.15	**1.41**±0.22	**1.31**±0.26	**0.84**±0.02

errors are on a similar scale to the simulation variance across all four metrics, suggesting that the prediction error is modest relative to the simulator's own stochasticity. More importantly, averaging over three seeds substantially reduces both simulation variance and prediction error. This is also the more practical setting, since full-benchmark evaluation typically needs at least three runs to average out CARLA stochasticity. Because FastB2D reduces evaluation time to roughly one quarter, three FastB2D runs still finish faster than a single full-benchmark pass while providing stable estimates.

(7) How does our method compare with existing benchmark compression methods?

We compare FastB2D with several BCM baselines. Table 5 reports cost, mean error, etc. for all four metrics. Each method is evaluated over 10 independent runs, and additional removal ratios appear in Appendix E. Three findings stand out.

First, FastB2D turns route reduction into far greater wall-clock saving. Most prior methods were developed on NLP benchmarks, where per-item evaluation cost is roughly uniform (Zhang et al., 2026; Huang et al., 2025; Polo et al., 2024; Vivek et al., 2024). They emphasize representativeness without explicitly modeling cost. In closed-loop AD, route simulation time varies by more than an order of magnitude, as discussed in Section 1. At the same removal ratio, these baselines retain far more simulation frames than FastB2D, whereas the cost-aware anchor selection in Eq. 2 keeps FastB2D's frame ratio well below the nominal removal ratio.

Second, FastB2D maintains balanced accuracy across heterogeneous metrics. Bench2Drive reports four metrics with very different statistical forms, and baselines often perform well on one but degrade sharply on others. AnchorPoint, for example, achieves competitive SR error at 50% removal yet exceeds 5% MAE on both DS and Smoothness. FastB2D instead keeps every metric within a narrow band, because its response-based representations capture route-level behavior shared across metrics, and the calibrated estimators in Eq. 4 are fitted independently for each metric. As a result, FastB2D attains the highest CE Score at every removal ratio, reaching 0.84 at 65% removal versus 0.58 for the strongest baseline.

Third, FastB2D produces notably more stable predictions across runs. Its standard deviations on per-metric mean error and CE Score are consistently lower than those of all baselines, often by a wide margin. This stability stems from the learned route parameters, which constrain the prediction space and reduce sensitivity to randomness in subset selection. Analysis (8) further compares FastB2D with these baselines in terms of OOD generalization. AD-specific baselines appear in Appendix C.6.

(8) Does FastB2D work well for out-of-distribution (OOD) models?

We examine FastB2D's OOD ability by evaluating models whose Success Rate lies outside the 50–75 training range described in Appendix C. On the high-SR side, we use the held-out test models in Table 2, including LEAD, BridgeDrive, and PDM-Lite (Nguyen et al., 2025; Liu et al., 2025; Dauner

Table 6: OOD generalization of proposed FastB2D. Ratio: percentage of routes retained. Cost (%): percentage of total simulation frames retained. SR and DS denote MAE on Success Rate and Driving Score. Avg MAE is the mean of SR and DS MAE. CE Score is defined in Section 3. Stochastic baselines report mean$\pm$std over 10 runs. Best per ratio in **bold**, and second best are underlined.

Method	Ratio	Cost (%) ↓	SR ↓	DS ↓	Avg MAE ↓	CE ↑
AnchorPoint (Vivek et al., 2024)	50%	62.3±0.8	4.30±0.62	1.93±0.21	3.12±0.33	0.43±0.04
SparseEval-Linear (Zhang et al., 2026)	50%	61.1±1.3	3.50±1.63	1.67±0.24	2.58±0.83	0.46±0.11
SparseEval-MLP (Zhang et al., 2026)	50%	60.3±1.2	8.30±1.96	1.47±0.29	4.89±0.99	0.17±0.12
TinyBenchmark (Polo et al., 2024)	50%	51.9±1.9	4.15±0.49	1.55±0.33	2.85±0.30	0.52±0.03
FastB2D (Ours)	50%	**43.9**±1.9	**1.71**±0.59	**1.00**±0.29	**1.35**±0.33	**0.71**±0.02
AnchorPoint (Vivek et al., 2024)	35%	42.0±1.2	5.59±0.50	2.08±0.21	3.83±0.27	0.55±0.04
SparseEval-Linear (Zhang et al., 2026)	35%	41.8±2.0	3.78±1.44	2.08±0.36	2.93±0.74	0.63±0.09
SparseEval-MLP (Zhang et al., 2026)	35%	41.1±1.2	10.69±1.53	1.69±0.30	6.19±0.78	0.08±0.16
TinyBenchmark (Polo et al., 2024)	35%	37.3±1.3	5.90±0.56	1.57±0.26	3.74±0.31	0.58±0.04
FastB2D (Ours)	35%	**27.1**±1.7	**1.90**±0.27	**1.52**±0.37	**1.71**±0.23	**0.84**±0.01
DEV10 (Jia et al., 2025)	4.5%	4.6	10.09	5.41	7.75	0.00
FastB2D (Ours)	4.5%	**2.5**±0.4	**9.21**±1.61	**4.21**±0.35	**6.71**±0.64	**0.27**±0.16

et al., 2023). On the low-SR side, we additionally evaluate UniAD-Tiny, UniAD-Base, and VAD (Hu et al., 2023b; Jiang et al., 2023), whose Success Rates lie around 13–15%. Table 6 reports the results under a unified OOD setting. Appendix F.5 further evaluates unseen model families.

At 50% route retention, FastB2D achieves the lowest error on both Success Rate and Driving Score among all baselines, along with the best CE score. Two complementary aspects of our design drive this robustness. First, route representations are learned from model evaluation records rather than from architecture families or expert scenario categories, so the resulting anchors capture behaviorally informative structure that transfers to unseen models. Second, the divergence-gated estimator in Section 2.4 detects when a test model lies far from the training distribution and falls back to the calibrated linear estimator, which extrapolates more reliably under distribution shift.

The advantage persists at tighter budgets. Reducing the retention to 35% further tightens the evaluation budget, yet FastB2D continues to deliver the lowest Success Rate and Driving Score errors among all baselines, as reported in the middle block of Table 6. The two strengths identified above carry over to this more constrained setting. Behaviorally informed anchors retain sufficient discriminative coverage to characterize unseen models, and the divergence gate continues to redirect estimation toward the calibrated form whenever a test model drifts from the training distribution.

The same pattern holds at an even more aggressive budget. At 4.5% retention, the bottom block of Table 6 compares FastB2D with DEV10 (Jia et al., 2025), the compact Bench2Drive subset built by expert semantic grouping. FastB2D again achieves substantially lower error on the same OOD models. This is consistent with our earlier observation in Section 1 and Figure 1(a), where neither random subsets nor expert semantic selection reliably approximated the full benchmark. In closed-loop driving, route outcomes depend jointly on road geometry, traffic interactions, and simulator stochasticity, so routes sharing an expert label may probe different failure modes, while routes from different categories may induce correlated responses. Response-based compression therefore offers a more faithful proxy than expert semantic selection, both in distribution and under shift.

4 Conclusion

In this work, we identify two intertwined sources of inefficiency in closed-loop autonomous driving benchmarks, namely statistical route redundancy and highly imbalanced per-route cost, which together show that route count alone is an inadequate proxy for evaluation expense. Guided by this insight, we propose a response-based and cost-aware compression method that learns route representations from model evaluation records and selects low-cost anchor routes through clustering. Instantiated on Bench2Drive, it yields cost-efficient FastB2D, which reduces the route count by 65% and total evaluation time by over 70%, achieves an MAE of $\sim$1.2% and a Spearman correlation of over 0.9 on Success Rate over several held-out model variants, and further cuts evaluation time by 27.2% while lowering estimation error by 64% with stronger out-of-distribution generalization compared with existing compression baselines and compact benchmarks.

AI USE STATEMENT

We used generative AI tools to assist with implementing parts of the code and to polish the manuscript's language. AI tools were not used to generate research ideas, develop the method, design experiments, process data, or analyze and interpret results, and were not used for any other purposes requiring disclosure. All AI-assisted code was reviewed and tested by the authors, and all edited text was checked for accuracy. We take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

Section 2 describes the full construction pipeline of FastB2D, and Section 2.3 specifies all key hyperparameters. The collection and split of the training, test, and validation records are detailed in Appendix C, the complete list of the 77 anchor routes is given in Table 8, and the performance estimators are detailed in Appendix C.3. Experimental results in Section 3 and Appendix E report mean and standard deviation over multiple runs, and the compute budget is reported in Appendix C.7. Data and code will be made public, including the anchor route IDs, the learned route parameters, and the calibrated estimators.

REFERENCES

autonomousvision. Bench2drive leaderboard. `https://github.com/autonomousvision/Bench2Drive-Leaderboard`, 2026.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

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.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024.

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.

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

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.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2024.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *Advances in Neural Information Processing Systems*, 36:7730–7742, 2023.

Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in neural information processing systems*, 16, 2003.

Wei He, Zhongzhan Huang, Mingfu Liang, Senwei Liang, and Haizhao Yang. Blending pruning criteria for convolutional neural networks. In *International Conference on Artificial Neural Networks*, pp. 3–15. Springer, 2021.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

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 (CVPR)*, 2023b.

Wenhui Huang, Songyan Zhang, Qihang Huang, Zhidong Wang, Zhiqi Mao, Collister Chua, Zhan Chen, Long Chen, and Chen Lv. Automot: A unified vision-language-action model with asynchronous mixture-of-transformers for end-to-end autonomous driving. In *International Conference on Machine Learning*, 2026.

Zhongzhan Huang, Guoming Ling, Shanshan Zhong, Hefeng Wu, and Liang Lin. Minilongbench: the low-cost long context understanding benchmark for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11442–11460, 2025.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. *arXiv preprint arXiv:2503.07656*, 2025.

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*, pp. 8340–8350, 2023.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926, 2021.

John P Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 648–657, 2016.

Derun Li, Changye Li, Yue Wang, Jianwei Ren, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Peng Jia, Xianpeng Lang, et al. Learning personalized driving styles via reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025.

Yun Li, Simon Thompson, Yidu Zhang, Ehsan Javanmardi, and Manabu Tsukada. An open-source modular benchmark for diffusion-based motion planning in closed-loop autonomous driving. *arXiv preprint arXiv:2603.01023*, 2026.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024a.

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? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14864–14873, 2024b.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

linggm3. Embench: Emergency benchmark for autonomous driving. `https://github.com/linggm3/EmBench`, 2026.

Shu Liu, Wenlin Chen, Weihao Li, Zheng Wang, Lijin Yang, Jianing Huang, Yipin Zhang, Zhongzhan Huang, Ze Cheng, and Hao Yang. Bridgedrive: Diffusion bridge policy for closed-loop trajectory planning in autonomous driving. *arXiv preprint arXiv:2509.23589*, 2025.

Shuai Liu, Siheng Ren, Xiaoyao Zhu, Quanmin Liang, Zefeng Li, Qiang Li, Xin Hu, and Kai Huang. Unidwm: Towards a unified driving world model via multifaceted representation learning. *arXiv preprint arXiv:2602.01536*, 2026.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. Lead: Minimizing learner-expert asymmetry in end-to-end driving. *arXiv preprint arXiv:2512.20563*, 2025.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2519–2536, 2024.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11993–12003, 2025.

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*, 2017.

Gayathri Saranathan, Cong Xu, Mahammad Parwez Alam, Tarun Kumar, Martin Foltin, Soon Yee Wong, and Suparna Bhattacharya. Sublime: Subset selection via rank correlation prediction for data-efficient llm evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30572–30593, 2025.

Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, pp. 726–737. PMLR, 2023.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15120–15130, 2024.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. Sparsedrivev2: Scoring is all you need for end-to-end autonomous driving. *arXiv preprint arXiv:2603.29163*, 2026.

Weizhe Tang, Junwei You, Jiaxi Liu, Zhaoyi Wang, Rui Gan, Zilin Huang, Feng Wei, and Bin Ran. Hermes: A holistic end-to-end risk-aware multimodal embodied system with vision-language models for long-tail autonomous driving. *arXiv preprint arXiv:2602.00993*, 2026.

Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25605–25615, 2025.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, 2024.

Sen Wang, Daoyuan Jia, and Xinshuo Weng. Deep reinforcement learning for autonomous driving. *arXiv preprint arXiv:1811.11329*, 2018.

Shaobo Wang, Cong Wang, Wenjie Fu, Yue Min, Mingquan Feng, Isabel Guan, Xuming Hu, Conghui He, Cunxiang Wang, Kexin Yang, et al. Rethinking llm evaluation: Can we evaluate llms with 200x less data? *arXiv preprint arXiv:2510.10457*, 2025.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yanhao Wu, Haoyang Zhang, Fei He, Rui Wu, Congpei Qiu, Liang Gao, Wei Ke, and Tong Zhang. Aligndrive: Aligned lateral-longitudinal planning for end-to-end autonomous driving. *arXiv preprint arXiv:2601.01762*, 2026.

Lijin Yang, Jianing Huang, Zhongzhan Huang, Shu Liu, and Hao Yang. Judge, then drive: A critic-centric vision language action framework for autonomous driving. *arXiv preprint arXiv:2604.27366*, 2026.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. *arXiv preprint arXiv:2205.09329*, 2022.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.

Taolin Zhang, Hang Guo, Wang Lu, Tao Dai, Shu-Tao Xia, and Jindong Wang. Sparseeval: Efficient evaluation of large language models by sparse optimization. *arXiv preprint arXiv:2602.07909*, 2026.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2024.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022.

APPENDIX CONTENTS

A The Novelty of Proposed FastB2D **16**

B Related Work **17**

B.1 End-to-End Autonomous Driving 17

B.2 Closed-Loop Benchmarks for Autonomous Driving 17

B.3 Benchmark Compression and Efficient Evaluation 17

C FastB2D Construction Details **18**

C.1 Collection of training, test, and validation sets 18

C.2 Anchor Route Selection Details 20

C.3 The Details of Performance Estimating 20

C.4 Scope of application of FastB2D 22

C.5 Comparison with Existing BCMs 23

C.6 Comparison with AD-Specific Route Selection 23

C.7 Cost analysis of FastB2D construction 24

C.8 Scenario Coverage Analysis 25

D More Details of Bench2Drive **26**

D.1 Rationale for focusing on Bench2Drive 26

D.2 Metrics adopted in Bench2Drive 26

D.3 Success Rate as the Primary Metric 27

D.4 Consistency of time-consuming routes 28

D.5 Experimental details of Figure 1 29

E Full Comparison across Removal Ratios **29**

F Additional Analysis **30**

F.1 Limitations of FastB2D 30

F.2 Can the proposed BCM be used for other closed-loop AD benchmarks? 30

F.3 The Relationship Between Wall-Clock Time and Frame Count 31

F.4 Stability of Per-Route Frame Counts 32

F.5 Generalization to New Model Families 32

F.6 Risk Screening of Removed Routes 34

F.7 Rotating Anchor Panels 34

F.8 Broader Impacts 34

F.9 Responsible Use 35

A THE NOVELTY OF PROPOSED FASTB2D

A cost-efficient closed-loop AD benchmark. The primary contribution of this work is FastB2D, a compact and cost-efficient closed-loop AD benchmark, together with the response-based and cost-aware compression method that constructs it. FastB2D builds on the widely adopted Bench2Drive and delivers a concrete release package consisting of 77 anchor routes, the learned route parameters, and the calibrated estimators for Success Rate, Driving Score, Efficiency, and Smoothness. The package removes 65% of the routes and cuts total evaluation time by over 70%, reducing an average 39-hour Bench2Drive run to roughly 11 hours on an A100. All route IDs, learned weights, and code will be released, so that AD researchers can adopt FastB2D in existing Bench2Drive evaluation pipelines without re-implementing the compression machinery.

Robust generalization beyond the training distribution. FastB2D is constructed from records of models with Success Rate in the 50–75 range, which raises a natural question of whether its predictions remain reliable on models with substantially higher or lower Success Rate. Analysis (8) addresses this directly through a unified OOD setting. On the high-Success-Rate side, we evaluate LEAD, BridgeDrive, and PDM-Lite at Success Rates ranging from 75% to over 90%. On the low-Success-Rate side, we additionally evaluate UniAD-Tiny, UniAD-Base, and VAD at Success Rates around 13–15%. In both regimes FastB2D achieves the lowest Success Rate and Driving Score errors among all compression baselines at the matched budget $K = 77$, and the advantage persists at the much smaller budget $K = 10$, where it continues to outperform the expert-curated DEV10. This robustness is not incidental. Response-based representations are learned from how models respond to each route rather than from any property of the training models themselves, so the route-level structure transfers across performance regimes. The divergence gate further detects when a test model drifts from the training distribution and falls back to the calibrated linear estimator, which extrapolates more reliably than predictive inference under shift. Appendix F.5 further shows that the fixed pipeline remains accurate on AutoMoT and SparseDriveV2, and that rebuilding FastB2D from an entirely independent construction pool preserves its fidelity.

Tailored to closed-loop testing. The construction of FastB2D is specifically designed for closed-loop AD testing rather than adapted from generic compression. Two structural properties of closed-loop AD evaluation shape every design choice. Per-route simulation cost varies by more than an order of magnitude with route length, traffic density, and interaction complexity, unlike the roughly uniform forward-pass cost in NLP benchmarks. Even within FastB2D's selected anchors, frame counts span 139 to 3,154 in Table 8. We therefore couple a normalized cost term with a representativeness term inside each cluster through Eq. 2. Route outcomes also depend on long-horizon interaction with the simulator and cannot be inferred from static scenario labels, so we describe routes by behavioral response patterns learned from model evaluation records. NLP-oriented BCMs such as TinyBenchmark (Polo et al., 2024), AnchorPoint (Vivek et al., 2024), and SparseEval (Zhang et al., 2026) assume uniform per-item cost and therefore retain expensive long routes that dominate the evaluation budget regardless of the removal ratio. This closed-loop-specific construction is what translates route removal into proportional wall-clock saving rather than nominal route reduction.

Compression in response space rather than input space. A natural alternative is to treat benchmark compression as data curation and select routes by their input properties, such as route metadata, scenario tags, road topology, or visual embeddings. Such properties describe what a route contains but not how models behave on it. In closed-loop driving, outcomes emerge from long-horizon interaction among the policy, traffic agents, and the simulator, so routes that look alike can separate models differently, while routes that look different can induce correlated outcomes. FastB2D therefore measures redundancy directly in the space of model responses. Appendix C.6 supports this choice. Three input-based strategies built on metadata, scenario tags, and route topology all incur higher cost and substantially larger error than FastB2D under the same 77-route budget.

Insights beyond cost reduction. Beyond a faster proxy, this work yields three findings about closed-loop AD evaluation. First, route redundancy and cost imbalance recur across benchmarks. The most expensive 20% of Bench2Drive routes account for nearly half of the total cost, and EmBench shows the same pattern in Appendix F.2, which suggests that future benchmarks should report item-level cost and remove redundant items. Second, semantic labels are an unreliable guide to measurement. Routes sharing a scenario label can probe different failure modes, which explains why the expert-curated DEV10 estimates the full benchmark less accurately than FastB2D. Third, compression changes

which evaluation protocol is affordable. A full Bench2Drive run takes 38.7 hours on average, whereas three FastB2D seeds take 34.2 hours, so seed-averaged evaluation becomes cheaper than a single full pass. The more than 200 released evaluation records, which cost roughly 8,700 A100 GPU-hours, further allow future work to study benchmark design without regenerating this pool.

B Related Work

Our work intersects three areas: end-to-end autonomous driving models, closed-loop evaluation protocols, and benchmark compression for efficient evaluation.

B.1 End-to-End Autonomous Driving

End-to-end AD replaces the classical perception, prediction, and planning stack with a single differentiable model that maps sensor inputs directly to planning outputs, reducing error accumulation across handcrafted modules (Chen et al., 2024; Hu et al., 2023b; Wu et al., 2022). Recent progress has rapidly broadened this paradigm, spanning transformer-based sensor fusion (Prakash et al., 2021; Chitta et al., 2023; Shao et al., 2023), query-based unified planning (Hu et al., 2023b; Jiang et al., 2023; Li et al., 2024b; Jia et al., 2025), planner-centric supervision (Dauner et al., 2023; Li et al., 2024a; Wu et al., 2026; Li et al., 2025), and generative trajectory modeling (Liao et al., 2025; Zheng et al., 2024). Language-grounded approaches have also become increasingly important, especially vision-language models (VLMs) and vision-language-action models (Zhang et al., 2024; Zhou et al., 2022) that bring open vocabulary perception, reasoning, and action generation into driving (Sima et al., 2024; Tian et al., 2024; Shao et al., 2024; Renz et al., 2025; Hwang et al., 2024; Jiang et al., 2024; Tang et al., 2026). World models further extend this line by enabling predictive simulation and long-horizon forecasting (Hu et al., 2023a; Gao et al., 2024; Liu et al., 2026). Reinforcement learning fine-tuning and large-scale driving pretraining further accelerate progress (Kiran et al., 2021; Wang et al., 2018; Sallab et al., 2017). As architectures and training paradigms continue to diversify, comprehensive closed-loop evaluation becomes both more important and more expensive. Our work addresses this need by compressing a representative closed-loop benchmark so that emerging end-to-end models can be assessed at a fraction of the original cost while preserving ranking fidelity.

B.2 Closed-Loop Benchmarks for Autonomous Driving

Open-loop benchmarks such as nuScenes (Caesar et al., 2020), Waymo Open (Sun et al., 2020), and Argoverse 2 (Wilson et al., 2023) compare predictions against logged ground truth, but cannot capture how a model's own actions reshape future states. Closed-loop benchmarks address this gap by requiring the model to drive routes while continuously interacting with a simulator, and recent efforts improve scale and realism along two complementary directions. Simulator-based benchmarks expand CARLA (Dosovitskiy et al., 2017) into denser interaction settings, with Bench2Drive (Jia et al., 2024) providing 220 routes spanning 44 scenarios and five driving skills under four metrics, and EmBench (linggm3, 2026) targeting safe takeover in emergency scenarios. Log-replay benchmarks instead unroll reactive or non-reactive agents on real-world driving logs (Caesar et al., 2021; Gulino et al., 2023; Dauner et al., 2024; Li et al., 2026), and scenario generation methods further synthesize challenging interactions to probe model robustness. Despite these advances, evaluation cost scales unfavorably with realism, as a full Bench2Drive run consumes roughly 40 GPU-hours per model and per-route times are highly skewed. Existing efforts to alleviate this cost rely on expert-curated compact subsets such as DEV10 (Jia et al., 2025), which select scenarios by semantic labels rather than by how routes actually distinguish models in closed-loop testing. Our work instead introduces a measurement-driven and cost-aware compression framework, instantiated on Bench2Drive as FastB2D, that preserves the ranking fidelity of the full benchmark while substantially reducing wall-clock cost.

B.3 Benchmark Compression and Efficient Evaluation

Most efficiency work in machine learning has focused on compressing model or reducing training data (He et al., 2021; Luo et al., 2017; Yang et al., 2022). Compression of evaluation sets has drawn attention only recently, especially in methods inspired by Item Response Theory (IRT) (Lalor et al., 2016). In psychometrics, IRT models how test items distinguish examinees of different ability, and

adaptive testing uses this structure to shorten tests while preserving measurement precision. Recent work brings this view to large language model evaluation by learning latent item representations (Polo et al., 2024), selecting anchor examples whose responses predict overall performance (Vivek et al., 2024), fitting estimators that map subset results to full benchmark scores (Zhang et al., 2026), and analyzing how subsampling affects rank reliability (Perlitz et al., 2024). Related directions such as coreset selection, active testing, etc, also aim to reduce evaluation cost. However, most existing methods (Saranathan et al., 2025; Wang et al., 2025; Marion et al., 2023; Huang et al., 2025) assume similar evaluation cost across test items. This assumption is often reasonable in NLP, but it breaks down in closed-loop AD, where route simulation time can vary by more than an order of magnitude with route length, traffic density, and interaction complexity. Our work extends benchmark compression to this setting by learning route representations from model responses, selecting low cost anchors, and fitting separate estimators for Success Rate, Driving Score, Driving Efficiency, and Driving Smoothness from one compact route set. Data curation methods that select samples by input properties, such as metadata or learned embeddings, pursue a related goal, but they ignore how models respond under closed-loop interaction, as discussed in Appendix A and evaluated in Appendix C.6.

C FastB2D Construction Details

C.1 Collection of training, test, and validation sets

Reliable route representations require a large pool of AD models with diverse evaluation records. Closed-loop AD evaluation contrasts with LLM benchmark compression in this respect, where dozens of strong public checkpoints are routinely available to train a compression pipeline. As Table 2 shows, the number of openly released SoTA end-to-end AD models is small, and we deliberately reserve all of them for testing so that the predicted scores can be validated against entirely independent systems. To compensate for the resulting scarcity of training records, we synthesize a controlled population of model variants by perturbing the training and inference configurations of two strong base models, SimLingo (Renz et al., 2025) and CriticVLA (Yang et al., 2026), and then filter the variants through statistical tests so that the surviving pool behaves meaningfully differently from the base models and remains disjoint from the test set. As demonstrated throughout the paper, this synthetic-record strategy is both feasible and broadly applicable, and provides a practical recipe for compressing future closed-loop AD benchmarks where evaluated SoTA models remain scarce.

Validity of retaining SimLingo and CriticVLA in the test set. Since the training pool is generated by perturbing SimLingo and CriticVLA, a natural concern is whether their inclusion in the test set introduces leakage. The two-stage filtering below is designed to prevent exactly this, ensuring that retained variants are statistically distinct from both base models in per-seed Success Rate and route-level response patterns. We further verify this empirically. The test MAE in Table 2 is 1.20%, and removing SimLingo and CriticVLA to enforce strict family-level held-out evaluation yields a comparable 1.24%. The negligible gap confirms that the filtering produces effective held-out behavior, so we keep both base models in the test set to broaden architectural and performance coverage. Analysis (8) further supports this split with reliable predictions on unseen architectures and on Success Rate ranges outside the training pool. Given the scarcity of openly released closed-loop AD models, this synthesize-and-filter strategy offers a practical recipe for future closed-loop benchmark compression.

Configuration axes. For each base model, we sample independent configurations along seven axes that fall into three functional groups.

The first group concerns supervision and language, and controls what training signal and what natural-language form the model sees. The reasoning module is either enabled or disabled, determining whether the model emits an explicit chain of rationales before predicting waypoints. The two auxiliary supervision streams, namely QA data and reasoning data, are independently included or excluded during training, yielding four combinations. The reasoning and prompt language is set to Chinese or English, which changes all natural-language inputs and intermediate reasoning text. These axes shift the inductive bias of the language head while leaving the perception backbone untouched.

The second group concerns optimization, covering the number of training epochs drawn from $\{5, 7, 9, 11, 13, 15\}$ and the batch size drawn from $\{4, 8, 16, 24\}$. Together they alter gradient noise, effective learning rate, and convergence regime, so configurations with identical supervision can still land on different local minima.

The third group concerns architecture and control, and governs the output side of the model. The waypoint decoder is either a multi-layer perceptron head or an LSTM head, which produce structurally different waypoint sequences from the same backbone features. The speed-control variant is drawn from four progressively modified PID-based controllers V1 to V4, all sharing the same lateral controller and the same longitudinal PID with $K_p = 1.75$, $K_i = 1.0$, $K_d = 2.0$ over a 20-frame integral window, and differing only in how the desired speed v_{des} and the brake gate $\mathcal{B}$ are computed from the predicted ego-frame waypoints $\{\mathbf{w}_i\}_{i=0}^{9}$ at times $t_i = (i+1)\Delta t$ with $\Delta t = 0.25\,\text{s}$. The baseline V1 sets

$$v_{\text{des}} = 2\,\|\mathbf{w}_0 - \mathbf{w}_2\|, \qquad \mathcal{B} = \big[\, v_{\text{des}} < 0.4 \,\big] \;\vee\; \big[\, v/v_{\text{des}} > 1.1 \,\big]. \tag{5}$$

V2 additionally requires a low far-segment speed $v_{\text{far}} = \frac{1}{3}\sum_{i=3}^{5} \|\mathbf{w}_i - \mathbf{w}_{i-1}\|/\Delta t < 0.3$ before the threshold branch can fire. V3 further replaces the ratio branch with the conjunction $[\, v/v_{\text{des}} > 1.15 \,] \wedge [\, v - v_{\text{des}} > 1.0 \,]$ to suppress spurious low-speed braking. V4 keeps the brake gate of V3 but lifts the desired speed at startup via $v_{\text{des}} = \max\big(2\,\|\mathbf{w}_0 - \mathbf{w}_2\|,\ \frac{1}{2}\,\|\mathbf{w}_3 - \mathbf{w}_2\|/\Delta t\big)$.

Random combinations across these three groups yield a large family of model variants that share backbone capacity with the base models but differ along at least one supervisory, optimization, or control dimension. This breadth is essential because the response-based representations in Section 2.1 can only separate routes that elicit different success patterns across the training pool.

Two-stage filtering. A pure cross-product over the configuration axes still produces many variants whose closed-loop behavior is statistically indistinguishable from the corresponding base model. To remove such redundancy, we apply two filtering stages.

Let $\mathcal{B}$ denote a base model and let v be a candidate variant evaluated with $n_v \in [5, 20]$ random seeds on the full $N = 220$-route Bench2Drive set. Write $\{s_v^{(k)}\}_{k=1}^{n_v}$ and $\{s_{\mathcal{B}}^{(k)}\}_{k=1}^{n_{\mathcal{B}}}$ for the per-seed Success Rates of v and $\mathcal{B}$. In the first stage, we apply Welch's two-sample t-test under the null hypothesis $H_0 : \mathbb{E}[s_v] = \mathbb{E}[s_{\mathcal{B}}]$ and retain v when the resulting p-value satisfies $p_v < 0.05$. The retained pool is

$$\mathcal{S} = \big\{\, v : p_v < 0.05 \,\big\}. \tag{6}$$

Variants in $\mathcal{S}$ produce per-seed Success Rates unlikely to coincide with $\mathcal{B}$ under seed variation alone.

The second stage examines route-level binary response vectors. For each $v \in \mathcal{S}$, let $\mathbf{x}_v \in \{0,1\}^N$ denote its per-route success vector aggregated across seeds, and define the Hamming distance to $\mathcal{B}$ as

$$d_H(v, \mathcal{B}) = \sum\nolimits_{j=1}^{N} \big[\, x_{v,j} \neq x_{\mathcal{B},j} \,\big]. \tag{7}$$

Writing $\bar{d}_H = \frac{1}{|\mathcal{S}|}\sum_{v\in\mathcal{S}} d_H(v, \mathcal{B})$ for the mean Hamming distance over $\mathcal{S}$, we keep v only when $d_H(v, \mathcal{B}) \geq 0.3\,\bar{d}_H$. This step discards variants that pass the t-test by chance but still produce response patterns very close to $\mathcal{B}$, leaving a population whose route-level behavior is reliably distinct from both base models.

Final split. The two filtering stages yield 208 retained variants. We randomly split them into 158 training records and 50 validation records at a ratio of roughly 3:1, with the validation set held out for all analyses in Section 3. The test set instead consists of the openly released SoTA models listed in Table 2, ensuring that test models share neither base architecture nor configuration with any record used during representation learning or estimator calibration.

Cross-architecture generalization. We reserve most of the open-source AD models for the test set to ensure rigorous validation, which limits the architectural diversity of the training pool. Nonetheless, the compression pipeline does not rely on architecture-specific features. The route representations learned in Section 2.1 capture how routes discriminate among models through their response patterns, and the diverse configurations across supervision, optimization, and control dimensions produce rich response diversity for learning informative representations. The experiments confirm strong

Table 7: Effect of the clustering algorithm on anchor selection. All methods use the same route representations, 77 routes, and the cost-aware criterion in Eq. 2. Cost (%): percentage of total simulation frames retained.

Clustering	Cost (%) ↓	SR MAE ↓	DS MAE ↓
K-Means (default)	27.01±1.81	1.71±0.20	0.961±0.104
K-Medoids	27.99±2.45	1.75±0.17	1.023±0.122
GMM	27.39±1.20	1.73±0.14	0.967±0.079
Ward	26.97	1.61	0.909

cross-architecture generalization: as shown in Table 2 and Table 6, test models from entirely unseen architecture families all receive accurate predictions from FastB2D without any retraining. To better support the evolving AD model ecosystem, we will additionally release a **fully-trained version of FastB2D** in which all available evaluation records, including those held out for the analyses in this paper, participate in representation learning and estimator calibration. This expanded training pool spans a broader range of architectures and performance levels, yielding more robust route representations. Since all records participate in training, this version cannot be validated internally and is intended for direct community use on new models after release.

C.2 Anchor Route Selection Details

Table 8 lists the complete set of 77 anchor routes selected by FastB2D. Each anchor is the representative of one behavioral group produced by K-Means on the learned route representations (Section 2.2). The frame count is the average number of simulation frames per evaluation run, serving as a proxy for evaluation cost. The last two columns report the learned estimator weights w_j from Eq. 4 for Success Rate and Driving Score, respectively. Both sets of weights are constrained to the probability simplex and reflect the relative contribution of each anchor to the full-benchmark estimate.

Frame counts vary by more than an order of magnitude, from 139 to 3,154, reflecting the substantial cost heterogeneity across routes. The total frame count of the 77 anchors is 49,357, compared to 175,927 for the full benchmark, confirming the cost reduction reported in Section 2.3. The SR estimator weights are relatively uniform across anchors, consistent with the high entropy regularization selected by cross-validation. In contrast, the DS estimator weights show greater variation, indicating that certain anchors are more informative for Driving Score prediction. This difference arises because Success Rate is a binary outcome with relatively homogeneous information content per route, whereas Driving Score is a continuous metric where some routes carry disproportionate predictive power.

Setting of the hyperparameter sweeps. In Analysis (1), the latent dimension D is swept with all $M = 158$ construction records and the route budget fixed at 77, so that only D varies.

Choice of clustering algorithm. We adopt K-Means because FastB2D selects one anchor per cluster, so the cluster count directly equals the route budget. To test whether this choice matters, we replace K-Means with K-Medoids, a Gaussian mixture model, and Ward agglomerative clustering. All variants use the same learned route representations, select 77 routes, and apply the same cost-aware criterion in Eq. 2. Ward clustering is deterministic and therefore has no standard deviation. Table 7 shows that all four methods reach comparable cost and accuracy. The learned representations thus expose a clear route structure that standard clustering methods recover equally well, and we keep K-Means for its simplicity.

C.3 The Details of Performance Estimating

Section 2.4 introduces two Success Rate estimators, predictive inference under the logistic model in Eq. 3 and calibrated weighted averaging (WA) in Eq. 4, and combines them through a divergence gate. To justify this design, Figure 4 reports the MAE of each estimator together with a simple averaging baseline across three SR ranges.

The predictive estimator achieves the lowest error in the $[50, 75]$ range, which covers the operating regime of most existing AD models and is therefore the most practically relevant. Within this range, the logistic ability model is well calibrated, and predictive inference fully exploits the learned response structure to estimate SR accurately. For models outside this range, the calibrated estimator is

Table 8: Complete list of 77 FastB2D anchor routes. **ID**: route index in the original Bench2Drive benchmark. **Frames**: average simulation frames per run. w_{SR} and w_{DS}: learned estimator weights for Success Rate and Driving Score (Eq. 4).

ID	Frames	w_{SR}	w_{DS}	ID	Frames	w_{SR}	w_{DS}	ID	Frames	w_{SR}	w_{DS}
2	375	.0129	.0180	68	156	.0118	.0154	147	446	.0124	.0104
4	3,154	.0121	.0104	73	438	.0136	.0158	149	434	.0138	.0205
5	938	.0131	.0085	74	290	.0125	.0089	150	374	.0139	.0182
8	230	.0137	.0116	75	1,257	.0122	.0087	161	376	.0126	.0092
11	254	.0127	.0123	82	154	.0140	.0164	162	383	.0136	.0125
13	388	.0131	.0199	86	488	.0132	.0154	169	532	.0123	.0099
17	2,668	.0126	.0120	88	373	.0136	.0106	171	523	.0135	.0141
19	1,038	.0121	.0148	95	374	.0120	.0113	174	2,569	.0120	.0137
21	1,316	.0133	.0112	97	682	.0123	.0128	177	965	.0126	.0109
23	307	.0136	.0115	100	143	.0136	.0115	184	2,774	.0130	.0071
27	153	.0127	.0145	102	776	.0135	.0124	186	327	.0134	.0085
28	161	.0139	.0136	107	1,125	.0117	.0076	187	474	.0138	.0107
32	473	.0126	.0095	108	215	.0141	.0156	191	2,745	.0130	.0119
33	502	.0134	.0130	109	455	.0134	.0088	193	227	.0137	.0119
34	979	.0120	.0100	112	488	.0117	.0133	198	1,099	.0129	.0142
35	1,161	.0132	.0113	113	320	.0135	.0118	200	446	.0135	.0165
38	563	.0133	.0133	118	592	.0126	.0115	204	739	.0128	.0091
41	177	.0139	.0163	120	322	.0125	.0084	205	384	.0132	.0113
43	192	.0129	.0079	123	423	.0133	.0116	208	436	.0119	.0111
48	1,087	.0132	.0147	124	390	.0140	.0183	209	370	.0123	.0146
49	424	.0135	.0161	128	1,123	.0118	.0163	210	348	.0121	.0150
52	362	.0138	.0119	133	289	.0138	.0124	211	268	.0136	.0151
58	310	.0141	.0219	137	339	.0124	.0094	213	871	.0127	.0107
60	139	.0134	.0158	140	1,038	.0121	.0177	217	259	.0122	.0149
63	400	.0138	.0141	142	300	.0140	.0177	218	262	.0124	.0145
66	145	.0118	.0146	143	280	.0130	.0151				

better suited, because its linear structure does not rely on parametric ability assumptions and remains reliable when the latent ability falls far from the calibration region. The gated estimator inherits both strengths. Within $[50, 75]$ the two estimates agree and the gate defaults to predictive inference, while outside this range their divergence triggers a switch to the calibrated form, yielding the best or near-best MAE across all three ranges. Simple averaging, in contrast, exceeds 3 MAE percentage points for models with SR below 75 and shows notably higher variance, confirming that learned estimation is necessary for reliable benchmark compression.

The remaining design choice is the gate threshold τ. We first sweep τ from 0.5% to 12% on the validation set and select the value that minimizes SR estimation MAE. This procedure identifies $\tau = 5\%$ and a stable margin around it, within which the gate makes the same decisions and attains the same best validation error. We then visualize the effect of τ on the held-out test models in Figure 5. The gated estimator remains accurate near $\tau = 5\%$ and stays competitive with or better than the two single-estimator baselines, showing that the validation-selected threshold transfers well to the test set and remains robust under out-of-distribution shift.

Role of calibration for continuous metrics. For Driving Score, Driving Efficiency, and Driving Smoothness, much of the accuracy comes from the metric-specific calibrated estimator rather than from anchor selection alone. Table 4(a) shows that our anchors with unweighted averaging can be less accurate than random subsets on Efficiency and Smoothness, because anchors selected for binary Success Rate need not match the full-benchmark mean of a continuous metric. Learned weighting removes this bias, and the combination of response-based anchors and calibration yields the lowest error on all three metrics. When adapting FastB2D to a new benchmark or metric, a calibrated estimator should therefore be fitted for each metric rather than shared across metrics.

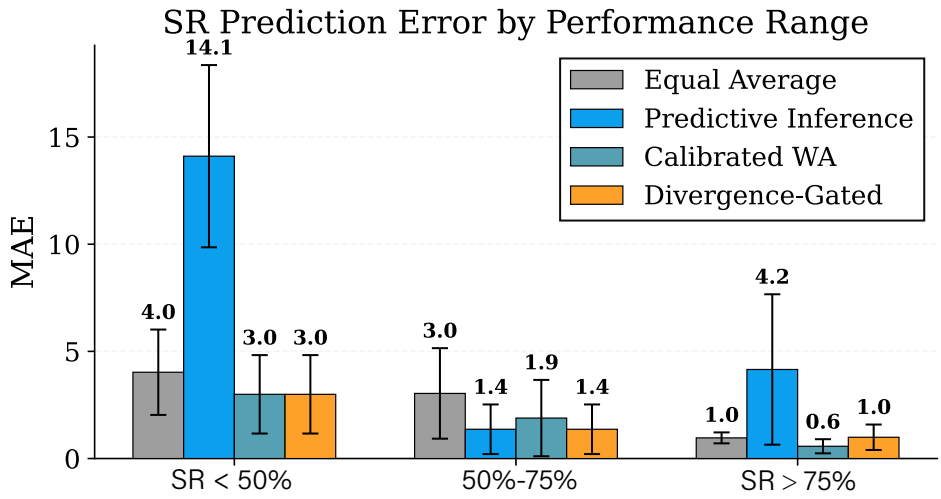

Figure 4: Estimation error of four SR estimation strategies across three performance ranges. Predictive Inference is the most accurate in the practically dominant $[50, 75]$ range, while the divergence-gated approach preserves this advantage and remains robust outside the training distribution.

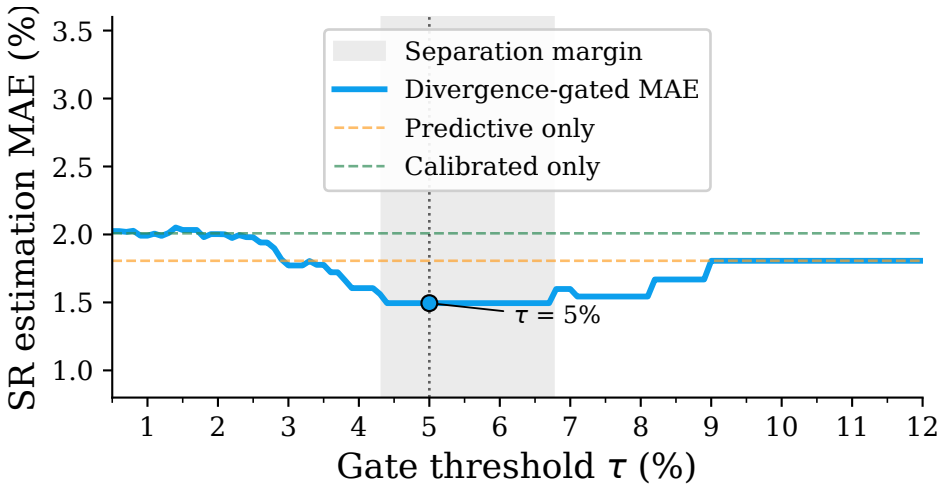

Figure 5: SR estimation MAE on held-out test models as a function of the gate threshold τ. We choose τ by sweeping 0.5% to 12% on the validation set. The shaded region marks the separation margin, where all thresholds produce the same gating decisions and achieve the same best validation error. The selected $\tau = 5\%$ lies in this region and yields the lowest test MAE, showing that the validation-chosen threshold transfers well and remains robust under distribution shift.

C.4 Scope of application of FastB2D

The construction of FastB2D in Appendix C restricts the training pool to records whose Bench2Drive Success Rate falls in the 50–75 range. This choice does not imply that FastB2D is only applicable to models within this regime, and instead serves two complementary purposes. First, this range covers the operating regime of current SoTA end-to-end AD models (autonomousvision, 2026), so anchors selected from these records align with the population of models that practitioners are most likely to evaluate. Second, holding out models above and below this range provides a controlled setting for testing how reliably the method generalizes under distribution shift, which is exactly the OOD setting examined in Analysis (8).

The experiments throughout the paper confirm that FastB2D generalizes well beyond the training range. Table 2 reports accurate predictions on test models whose Success Rate spans a wide regime,

ranging from weaker models such as DriveTransformer at 36%+ to stronger models such as PDM-Lite at 90%+. Analysis (8) further shows that prediction error remains low under the OOD setting, and the divergence-gated estimator in Appendix C.3 provides additional robustness when a test model lies far from the training distribution. Together, these results demonstrate that FastB2D is broadly applicable across the full performance spectrum encountered in closed-loop AD evaluation.

To better support the evolving AD model ecosystem, we will additionally release a fully-trained version of FastB2D in which all available evaluation records, including those held out for the analyses in this paper, participate in representation learning and estimator calibration. Because every record contributes to training, this version cannot be validated internally and is intended to be evaluated by users on their own models after release. The 77 anchor routes, the learned route parameters, and the calibrated estimators will all be made publicly available together with the code used in this work.

C.5 Comparison with Existing BCMs

Our compression method differs from existing BCMs in three substantive ways, each motivated by structural properties of closed-loop AD evaluation that NLP-oriented BCMs do not encounter.

Cost-aware anchor selection. Existing BCMs such as TinyBenchmark (Polo et al., 2024), AnchorPoint (Vivek et al., 2024), and SparseEval (Zhang et al., 2026) were developed on NLP benchmarks, where per-item evaluation cost is roughly uniform across a forward pass. They therefore optimize for representativeness alone and treat all retained items as equally cheap. Closed-loop AD violates this assumption, since per-route simulation time varies by more than an order of magnitude with rollout length, traffic interaction, and failure timing. Our method couples a normalized per-route cost term into the anchor selection objective in Eq. 2, so that representativeness and wall-clock saving are jointly optimized within each cluster instead of being addressed as separate objectives. Table 5 confirms that this change yields substantially lower frame ratios at matched route counts and translates route reduction into proportionally larger time savings.

Metric-aware estimation pipeline. Standard BCMs target a single accuracy metric and degrade on other metrics, as also shown in Table 5. Bench2Drive instead reports four metrics with very different statistical properties, ranging from binary Success Rate to Driving Score, Driving Efficiency, and Driving Smoothness. FastB2D recovers all four from a single anchor set by pairing predictive inference under the learned logistic ability model in Eq. 3 for Success Rate with entropy-regularized calibrated estimators in Eq. 4 for the three continuous metrics. The logistic form is fitted once during representation learning, and the calibrated weights are learned independently per metric, so each estimator can exploit the structure of its own metric. Existing BCMs provide no analogous mechanism for jointly handling response-based and continuous evaluation signals from one compressed route set.

Divergence-gated estimation for OOD models. Existing BCMs produce a single point estimate and offer no internal mechanism for detecting when a test model lies outside the training population. This becomes problematic in closed-loop AD, where the model ecosystem evolves rapidly and new architectures regularly fall outside any fixed training pool. Our method combines predictive inference with a calibrated linear estimator and selects between them based on their divergence, falling back to the calibrated form whenever the two disagree by more than $\tau = 0.05$. This design exploits the complementary strengths of the two estimators, since predictive inference is more accurate within the training distribution while the calibrated linear form extrapolates more reliably outside it. Analysis (8) shows that the gated estimator improves robustness on unseen architectures and on Success Rate ranges outside the training pool, where prior BCMs become unstable.

C.6 Comparison with AD-Specific Route Selection

Beyond general-purpose BCMs, we compare FastB2D with route selection strategies built on the structure of driving scenarios. These strategies select routes before any model is evaluated and use only information available in the route definitions.

Metadata Stratification clusters routes by town, weather, route length, elevation change, and turning statistics, then selects the route closest to each cluster center.

Scene Tag Coverage first selects one route from each of the 44 scenario types and then distributes the remaining 33 selections randomly across these types.

Table 9: **Left:** comparison with AD-specific route selection baselines under the same 77-route budget. Cost (%): percentage of total simulation frames retained. **Right:** comparison with the expert-curated DEV10 under the same 10-route budget on the validation models. All errors are MAE.

Method	Cost (%) ↓	SR ↓	DS ↓
Metadata Stratification	36.39±2.38	5.81±1.68	2.57±0.64
Scene Tag Coverage	32.50±0.63	3.99±0.21	1.74±0.07
Route Topology Stratification	35.97±1.43	8.24±2.25	3.78±0.94
FastB2D (Ours)	**27.01**±1.81	**1.71**±0.20	**0.961**±0.104

Method	SR ↓	DS ↓	Avg ↓
DEV10	11.17	5.29	8.23
FastB2D (Ours)	**6.29**±0.99	**2.71**±0.72	**4.50**±0.72

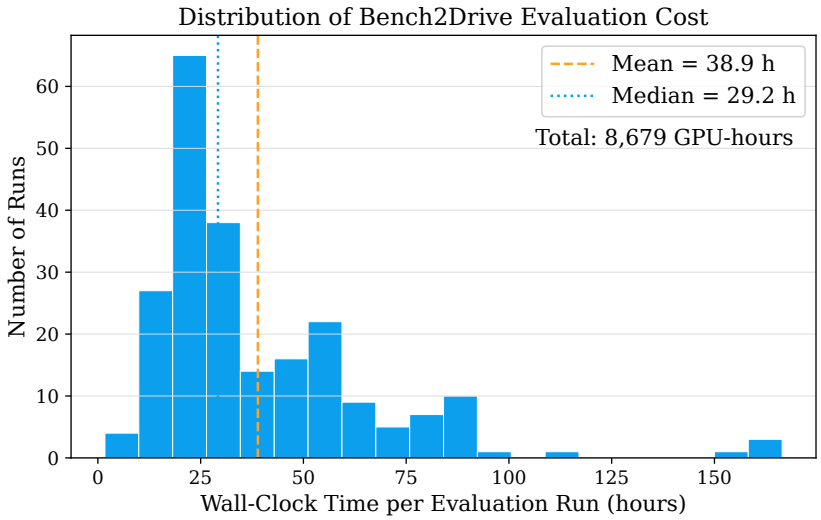

Figure 6: Distribution of wall-clock time per Bench2Drive evaluation run. The collected evaluation pool totals approximately 8,700 A100 GPU-hours. A single full evaluation costs 39 hours on average (mean) but varies widely, with the median at 29 hours and individual runs ranging from under 2 hours to over 160 hours depending on the model.

Route Topology Stratification aligns and resamples waypoint trajectories, clusters routes by spatial layout and turning pattern, and selects the route closest to each cluster center.

All three baselines select 77 routes and estimate full-benchmark scores by direct weighted aggregation. The left part of Table 9 shows that FastB2D achieves both the lowest evaluation cost and the lowest prediction error. Scene Tag Coverage is the strongest static strategy, yet its Success Rate error remains more than twice that of FastB2D. Static route properties describe what a route contains but not how models respond to it, and they carry no cost signal, so they neither remove measurement redundancy nor avoid expensive routes. The comparison covers complete evaluation pipelines under the same route budget, and Table 4(a) separately isolates the contribution of the learned estimator.

Comparison with DEV10 on the validation set. Table 6 compares FastB2D with the expert-curated DEV10 subset on OOD models. Under the same 10-route budget on the validation models, FastB2D also reduces the average MAE from 8.23 to 4.50, as shown in the right part of Table 9.

C.7 Cost analysis of FastB2D construction

Constructing FastB2D requires a pool of evaluation records for representation learning and anchor selection. We collect over 200 Bench2Drive evaluation records in total, consuming approximately 8,700 A100 GPU-hours, or about 360 GPU-days (roughly $20,000), as shown in figure 6.

Although the total cost appears substantial, collecting evaluation records does not require dedicated effort in practice. The evaluation records are natural byproducts of closed-loop AD research, as any group that iterates on AD models accumulates them through regular experimentation. The records used in this work were gathered over the course of developing and evaluating multiple model variants,

not collected specifically for benchmark compression. The compression pipeline itself, including representation learning, clustering, and estimator calibration, completes in under 0.1 GPU-hour.

The full record pool is also not strictly necessary. As shown in Analysis (2), even $M = 10$ training records yield an MAE below 2.5%, and prediction quality stabilizes around $M = 50$. A research group with only a small number of completed evaluations can already apply the pipeline to obtain a reliable compressed benchmark. Further reducing the data requirement is a direction for future work.

C.8 Scenario Coverage Analysis

Our compression framework selects anchor routes entirely from learned response patterns, without access to any scenario type labels. Despite this, FastB2D retains 40 of the 44 scenario types defined in Bench2Drive (As shown in Table 1), covering 91% of all scenario types. This high coverage emerges naturally because routes that test distinct driving competencies tend to elicit distinct model response patterns, which BCM captures through its learned representations. The four absent types are ControlLoss, HardBreakRoute, VanillaSignalizedTurnEncounterGreenLight, and OppositeVehicleTakingPriority. Below we analyze each and show that these omissions do not compromise neither evaluation completeness nor safety-relevant coverage. In every case, the driving competencies involved are already tested by retained scenarios of equal or greater difficulty.

ControlLoss. Near-ceiling success rates make this scenario unable to distinguish models. Across 208 models, the five ControlLoss routes achieve success rates between 99.5% and 100%, with only 3 failures out of 1,040 total evaluations. Since nearly all models pass every route, this scenario contributes no discriminative signal to the evaluation. All five routes fall into the same response-based cluster as the retained InvadingTurn, which tests a similar reactive-avoidance competency within the Emergency Brake skill.

HardBreakRoute. Its high success rate offers limited discrimination, and harder retained scenarios already test the same braking competency. With a mean success rate of 94.3%, HardBreakRoute ranks among the easier Emergency Brake types. Its five routes are distributed across five separate clusters, each represented by an anchor from a retained braking or obstacle-avoidance scenario. The retained set includes nine other Emergency Brake types such as DynamicObjectCrossing, StaticCutIn, and BlockedIntersection, all with substantially lower success rates, collectively testing the same competency at greater difficulty.

VanillaSignalizedTurnEncounterGreenLight. Most routes are too easy to be informative, and the harder red-light counterpart is already retained. Four of the five routes have success rates above 98.5%, offering little discriminative value. The harder counterpart VanillaSignalizedTurnEncounterRedLight is retained and tests a superset of signal-compliance competencies, as the ego vehicle must additionally detect the red light and wait before turning. Further retained scenarios covering the same skills include VanillaNonSignalizedTurn, SignalizedJunctionRightTurn, and SignalizedJunctionLeftTurn.

OppositeVehicleTakingPriority. Harder retained scenarios already cover the same yielding and right-of-way competency. This scenario requires the ego vehicle to yield or brake when an oncoming vehicle asserts right-of-way. Its five routes show heterogeneous response patterns in the learned representations and are each assigned to a different cluster anchored by retained scenarios. The retained OppositeVehicleRunningRedLight tests the same core competency with an added red-light violation, and additional retained scenarios involving right-of-way negotiation include InvadingTurn, BlockedIntersection, and VehicleTurningRoute.

In summary, ControlLoss and HardBreakRoute have near-ceiling success rates that provide negligible discriminative value, while VanillaSignalizedTurnEncounterGreenLight and OppositeVehicleTakingPriority are subsumed by harder retained scenarios testing the same driving competencies. All 20 absent routes are represented within response-based clusters by anchors from functionally related scenarios, confirming that these omissions do not create an evaluation gap.

D MORE DETAILS OF BENCH2DRIVE

D.1 RATIONALE FOR FOCUSING ON BENCH2DRIVE

The compression framework in Section 2 applies to any closed-loop AD benchmark for which a response matrix and per-route costs are available. We instantiate it on Bench2Drive for three complementary reasons that span benchmark structure, structural inefficiency, and practical relevance.

(1) Structured evaluation protocol. Bench2Drive provides a comprehensive and well-structured evaluation protocol. Its 220 test routes span 44 interactive scenario types organized into 5 driving skill categories, and each model is assessed on 4 complementary metrics (Jia et al., 2024). This multi-granularity design enables analysis of route redundancy and cost imbalance at the scenario, skill, and metric levels, and it allows direct verification that compression preserves both aggregate scores in Table 2 and skill-level diagnosis in Table 3. Few existing closed-loop AD benchmarks combine this breadth of scenarios with a structured skill and metric hierarchy.

(2) Redundancy and cost imbalance. The compression framework is most effective when route redundancy and per-route cost vary widely, which are exactly the structural properties Bench2Drive exhibits. As shown in Figure 1, many random subsets approximate full-benchmark scores well, indicating that response patterns across the 220 routes are highly correlated. Per-route frame counts are simultaneously skewed by more than an order of magnitude, ranging from 139 to 3,154 frames among FastB2D anchors alone, as listed in Table 8. These two properties together constitute the structural inefficiency that response-based representation learning and cost-aware anchor selection are designed to exploit, making Bench2Drive a natural and demanding testbed for the framework.

(3) Wide community adoption. Bench2Drive is currently a primary closed-loop evaluation standard in the AD community, with a wide range of architectures evaluated on it (Chitta et al., 2023; Hu et al., 2023b; Jiang et al., 2023; Jia et al., 2025; Renz et al., 2025; Yang et al., 2026; Tang et al., 2025), and a single full run averages roughly 39 GPU-hours on an A100, as shown in Figure 6. Compressing such a widely used and expensive benchmark delivers immediate practical value, since researchers can adopt FastB2D directly within existing Bench2Drive pipelines to accelerate model iteration. Wide adoption also provides a methodological benefit. Many SoTA AD models have been openly released and evaluated on Bench2Drive, giving us a diverse pool of architectures that we reserve exclusively for the test set, as detailed in Appendix C. As a result, every test model in Table 2 and Table 6 comes from an architecture that contributed no record to representation learning or estimator calibration, allowing FastB2D to be validated under genuine cross-architecture and out-of-distribution conditions.

The framework itself is not tied to Bench2Drive. We additionally apply it to EmBench (linggm3, 2026) and obtain effective compression, as reported in Appendix F.

D.2 METRICS ADOPTED IN BENCH2DRIVE

Bench2Drive evaluates AD models with four complementary metrics (Jia et al., 2024). The first two capture goal-achieving ability, while the remaining two measure driving quality beyond task completion.

Success Rate (SR). Success Rate measures the proportion of routes completed within the allotted time with no traffic violations. A route is successful only if the ego vehicle reaches its destination with no infraction recorded, such as no collision and no red-light violation. Formally, $\text{SR} = n_{\text{success}}/n_{\text{total}}$, where n_{success} and n_{total} denote the number of successful and total routes, respectively.

Driving Score (DS). Driving Score follows the CARLA Leaderboard protocol (Dosovitskiy et al., 2017) and combines route completion with infraction penalties. Formally,

$$\text{DS} = \frac{1}{n_{\text{total}}} \sum_{i=1}^{n_{\text{total}}} \text{RC}_i \cdot \prod_j p_{i,j}, \tag{8}$$

where $\text{RC}_i \in [0, 100]$ is the route completion percentage for route i and $p_{i,j} \in (0, 1]$ is the penalty factor for the j-th infraction. More severe infractions carry lower multipliers. Since penalties compound multiplicatively, a single serious infraction can substantially reduce the overall score.

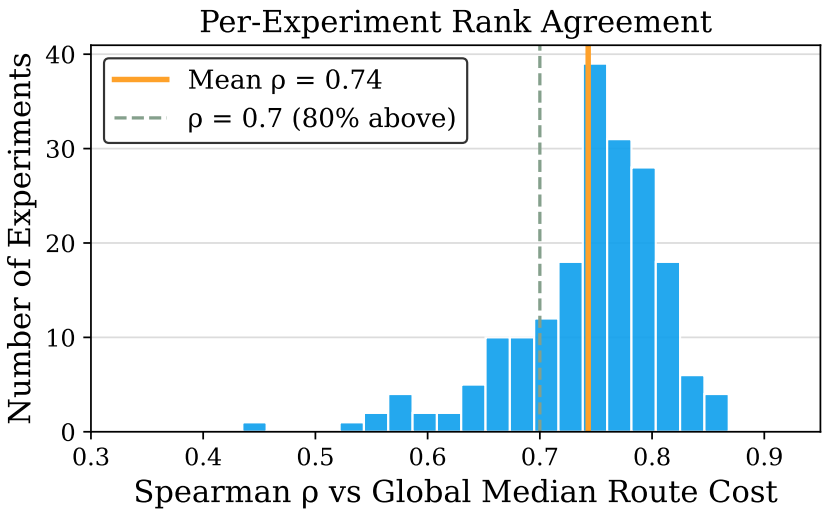

Figure 7: Distribution of per-experiment Spearman ρ between route-level frame counts and the global median across all experiments. The distribution is concentrated in the high-correlation region, confirming that route cost rankings are stable across models and seeds.

Driving Efficiency. Driving Efficiency assesses whether the ego vehicle maintains a reasonable speed relative to surrounding traffic. At each checkpoint, the ego vehicle's speed is compared to the average speed of nearby vehicles, yielding a speed percentage. The standard CARLA Leaderboard uses four checkpoints per route. Bench2Drive increases this to 20, checking speed every 5% of the route length to reduce measurement variance. Driving Efficiency is reported separately from Driving Score and is defined as the mean speed percentage $\overline{v}_{\%} = (\sum_i v_{\%,i})/C$ across all C checkpoints. Routes where the ego vehicle fails to pass the initial 5% checkpoint are excluded. Speed percentage values exceeding 1000% are filtered as measurement anomalies.

Driving Smoothness. Driving Smoothness evaluates trajectory comfort following the nuPlan protocol (Caesar et al., 2021). At each frame, six kinematic variables are checked against expert-derived thresholds: longitudinal acceleration, lateral acceleration, yaw rate, yaw acceleration, longitudinal jerk, and jerk magnitude. A frame is smooth only if all six fall within their bounds. Frame-level evaluation can, however, unfairly penalize necessary reactions such as emergency braking. To address this, Bench2Drive segments the trajectory into intervals of 20 timesteps and evaluates smoothness per segment. The final score is the ratio of smooth segments to total segments: Smoothness $= S_{\text{smooth}}/S_{\text{total}}$. Segments where the ego vehicle remains stopped with speed below 0.1 for more than 60 seconds are counted as smooth, as low speed presents no comfort concern. Routes with fewer than 20 total frames are excluded from the computation.

D.3 Success Rate as the Primary Metric

Bench2Drive evaluates AD models with four metrics: Success Rate, Driving Score, Driving Efficiency, and Driving Smoothness (Appendix D.2). Among them, Success Rate holds a central role in the evaluation framework, and we accordingly adopt it as the primary metric for compression. This choice is motivated by three considerations.

Metric hierarchy in Bench2Drive. Bench2Drive organizes its four metrics into two tiers. Success Rate and Driving Score measure goal-achieving ability, while Driving Efficiency and Driving Smoothness capture supplementary quality aspects "beyond the goal achieving ability" (Jia et al., 2024). Of the two goal-achieving metrics, Driving Score "lacks details and its heavy punishment encourages over-conservative driving strategies" (Jia et al., 2024). SR, in contrast, provides a clean binary signal of whether a model completes a route without infractions, making it the most direct and interpretable indicator of driving competence. It is also a widely reported metric in closed-loop AD evaluation.

Table 10: Full comparison of benchmark compression methods across four removal ratios on val set. Cost (%): percentage of total simulation frames retained. CE Score is defined in Section 3. Results are averaged over 10 runs (mean±std). Best results per group are in **bold**; second best are underlined.

Method	Cost (%) ↓	SR ↓	DS ↓	Eff. ↓	Smooth. ↓	Avg MAE ↓	CE ↑
Removal ratio = 25% ($K = 165$)							
Random-Equal	**75.1**±3.1	1.42±0.46	0.69±0.20	3.67±2.18	1.01±0.41	1.70±0.59	0.37±0.00
AnchorPoint	84.8±0.3	**0.38**±0.04	2.83±0.11	1.83±0.27	2.90±0.21	1.99±0.08	0.24±0.00
SparseEval-Linear	84.0±0.7	8.37±2.35	2.67±0.20	1.84±0.27	2.44±0.14	3.83±0.58	0.00±0.00
SparseEval-MLP	83.8±1.1	1.44±0.11	2.93±0.25	2.84±1.88	1.96±0.55	2.29±0.53	0.24±0.01
TinyBenchmark	81.7±1.3	0.95±0.13	1.59±0.29	3.14±1.65	1.82±0.53	1.88±0.32	0.28±0.01
FastB2D (Ours)	75.8±1.8	0.64±0.06	**0.47**±0.03	**1.35**±0.09	**0.61**±0.04	**0.76**±0.02	**0.39**±0.01
Removal ratio = 50% ($K = 110$)							
Random-Equal	50.0±3.3	2.47±0.90	1.20±0.39	2.68±1.67	1.73±0.70	2.02±0.59	0.54±0.08
AnchorPoint	62.3±0.8	1.67±0.14	5.84±0.21	1.36±0.41	5.00±0.61	3.47±1.97	0.08±0.12
SparseEval-Linear	61.1±1.3	2.80±0.68	5.17±0.33	1.93±0.97	4.48±0.84	3.59±1.29	0.00±0.28
SparseEval-MLP	60.3±1.2	1.57±0.12	5.15±0.40	2.05±1.05	3.65±0.88	3.11±1.41	0.25±0.13
TinyBenchmark	51.9±1.9	2.31±0.50	1.53±0.35	1.62±0.86	1.69±0.83	1.79±0.31	0.56±0.05
FastB2D (Ours)	**43.9**±1.9	**1.31**±0.12	**0.76**±0.06	**0.77**±0.10	**0.92**±0.07	**0.94**±0.22	**0.72**±0.02
Removal ratio = 65% ($K = 77$)							
Random-Equal	35.0±3.3	3.38±1.18	1.63±0.50	3.56±2.11	2.36±0.96	2.73±0.78	0.56±0.15
AnchorPoint	42.0±1.2	2.77±0.10	6.17±0.51	3.36±1.00	4.12±0.43	4.11±1.29	0.00±0.22
SparseEval-Linear	41.8±2.0	2.31±0.42	5.49±0.79	2.65±1.76	3.98±0.96	3.61±1.25	0.27±0.23
SparseEval-MLP	41.1±1.2	1.90±0.12	5.43±0.90	5.79±1.30	2.84±0.67	3.99±1.66	0.08±0.27
TinyBenchmark	37.3±1.3	3.30±0.90	1.75±0.34	2.84±1.40	2.45±0.94	2.59±0.57	0.58±0.10
FastB2D (Ours)	**27.1**±1.7	**1.67**±0.21	**0.97**±0.14	**1.19**±0.15	**1.41**±0.22	**1.31**±0.26	**0.84**±0.02
Removal ratio = 80% ($K = 44$)							
Random-Equal	20.1±2.6	4.77±1.43	2.32±0.62	13.48±8.35	3.43±1.45	6.00±2.24	0.10±0.10
AnchorPoint	23.6±0.7	3.73±0.24	6.51±0.79	8.12±2.81	3.81±1.09	5.54±0.83	0.29±0.12
SparseEval-Linear	23.0±1.9	2.60±0.36	4.95±1.15	9.27±2.16	3.84±0.91	5.16±0.78	0.41±0.09
SparseEval-MLP	25.8±1.5	**2.24**±0.20	6.41±0.98	10.68±5.52	4.37±1.03	5.93±1.42	0.13±0.14
TinyBenchmark	20.7±2.2	4.62±1.12	2.54±0.46	12.79±3.91	3.73±1.46	5.92±0.89	0.13±0.13
FastB2D (Ours)	**11.3**±1.1	3.33±0.44	**1.53**±0.22	**3.53**±0.60	**2.20**±0.27	**2.65**±0.20	**0.94**±0.00

Foundation of skill-level evaluation. Bench2Drive evaluates five driving skills: Merging, Overtaking, Emergency Brake, Give Way, and Traffic Sign. Each skill score is defined as the Success Rate over skill-specific routes (Jia et al., 2024), and their average forms the summary metric, Mean Ability. Success Rate thus underpins not only the overall benchmark score but also the fine-grained skill-level diagnosis that distinguishes Bench2Drive from earlier closed-loop benchmarks.

Clean signal that generalizes across metrics. Beyond its importance in the evaluation protocol, Success Rate also provides clean route-level signal for learning how each route discriminates among models. Its binary outcome is free from the multiplicative penalty accumulation in Driving Score and the thresholding and aggregation variability in Driving Efficiency and Driving Smoothness. This yields more stable route representations for anchor selection, which in turn generalize well to the other three metrics, as confirmed in Section 3. Given this central role, we optimize the compression framework for Success Rate and apply predictive inference (Eq. 3) to this metric.

D.4 Consistency of time-consuming routes

Cost-aware anchor selection (Eq. 2) assumes that the relative evaluation cost of each route is stable across experiments, i.e., routes that are costly for one model remain costly for others. To verify this, we collect frame counts for all routes from all independent experiments and compute the Spearman rank correlation ρ between each experiment's route-level frame counts and the global median.

As shown in Figure 7, the distribution is concentrated in the high-correlation region, with a mean ρ of 0.74 and over 80% of experiments exceeding 0.7. This confirms that route cost rankings are well preserved across different models and seeds, supporting the use of a fixed cost estimate in Eq. 2.

D.5 Experimental details of Figure 1

Figure 1 illustrates route redundancy and cost imbalance in Bench2Drive from three complementary perspectives. We describe the experimental setup for each panel below.

Random sampling. We use validation set in Appendix C, each tested on the full 220-route Bench2Drive benchmark. For each of nine removal ratios from 10% to 90%, we randomly draw 10,000 retained subsets of the corresponding size and compare each subset against the full benchmark using two complementary metrics. Mean absolute error on per-model Success Rate captures score-level agreement, while Spearman rank correlation captures whether the relative ranking of models is preserved. Reporting both is essential, because a subset can match the absolute scores of the full benchmark while distorting model ordering, or vice versa. Evaluation cost is reported as the percentage of total simulation frames retained rather than the percentage of routes retained, since per-route simulation length varies by more than an order of magnitude and a uniform route-count ratio would substantially understate the variance in actual testing time across subsets.

Per-route cost distribution. Per-route evaluation cost is measured as the average number of simulation frames across all collected evaluation records. Closed-loop simulation runs at a fixed control frequency, so frame count is directly proportional to wall-clock time and remains independent of model architecture or hardware, as further verified in Appendix F.3. The histogram covers all 220 routes and reveals a strongly right-skewed distribution, in which the top 20% most expensive routes account for nearly half of the total simulation budget while many short routes contribute only a small fraction. This imbalance indicates that any compression strategy based on route count alone forfeits substantial wall-clock saving, because the dominant cost concentrates in a narrow band of long routes that route-count quotas cannot selectively remove.

FastB2D. Figure 1 (c) visualizes the 77 FastB2D anchor routes against the per-route cost distribution from panel (b), illustrating how the anchors concentrate toward shorter routes. The same anchor set is additionally placed alongside the random subsets in panel (a) under identical cost and error protocols. FastB2D retains 27% of the total simulation frames while achieving a Success Rate MAE of 1.36% and a Spearman correlation of 0.93 on held-out test models, and sits closer to the low-cost and low-error region than typical random subsets at the same route count. This positioning reflects how response-based representation learning addresses the redundancy revealed in panel (a) and cost-aware clustering addresses the imbalance revealed in panel (b), with the two design ingredients acting jointly rather than in isolation.

E Full Comparison across Removal Ratios

Table 10 extends the main paper's comparison to removal ratios of 25% and 80%, beyond the 50% and 65% reported in Section 3. The three advantages identified there, namely smaller frame retention, lower and more balanced prediction error, and lower aggregate variance, largely hold across this wider range. From 50% onward, FastB2D retains the fewest frames among all methods and the gap to baselines tends to widen with compression. FastB2D also attains the lowest average MAE at every ratio, with the strongest baseline still showing roughly twice this error at the two extremes, and its standard deviation on average MAE is the smallest throughout.

Two boundary cases deserve mention. At 25% removal, where 165 of the 220 routes are retained, FastB2D's frame retention is close to that of Random-Equal since cost-aware selection has little room to operate when most routes are already kept. On SR, AnchorPoint at 25% and SparseEval-MLP at 80% achieve lower error than FastB2D, although FastB2D remains the best on Driving Score, Efficiency, Smoothness, and the overall average in both cases. Overall, the extended comparison indicates that FastB2D's gains are broadly consistent across the practical compression range.

F Additional Analysis

F.1 Limitations of FastB2D

Bootstrap cost of benchmark construction. Building FastB2D requires a pool of full Bench2Drive evaluation records, totalling roughly 8,700 A100 GPU-hours in our case. Two factors keep the practical cost modest. First, such records typically arise as byproducts of ordinary AD research, and ours were accumulated during the routine development of model variants. Second, Analysis (2) shows that as few as 50 records suffice for stable predictions, and even $M = 10$ already keeps MAE below 2.5%. A research group with only a handful of completed evaluations can therefore apply the framework directly, and further lowering this requirement is a natural direction for future work.

Inherited stochasticity of closed-loop simulation. FastB2D estimates full-benchmark scores from anchor outcomes, but it does not remove the intrinsic stochasticity of CARLA, which Analysis (6) shows contributes roughly half of the observed Success Rate variance. This is a general property of closed-loop AD evaluation, and the standard remedy is to average over multiple seeds. Since FastB2D reduces evaluation time to about one quarter of the full benchmark, three repeated FastB2D runs still finish faster than a single full Bench2Drive pass while approaching the variance of seed-averaged full evaluation, which makes this remedy substantially more affordable.

Limited architectural diversity in the training pool. As validated in Appendix C, to preserve genuine cross-architecture validation, we reserve all openly released SoTA AD models in Table 2 for the test set, leaving the training pool dominated by configuration-perturbed variants of two base models. This partly reflects the current AD ecosystem, where strong open-source models remain scarce. Two design choices help compensate. Our route representations describe how a population of models responds to each route rather than relying on architecture-specific cues, and the divergence-gated estimator in Section 2.4 routes OOD models to the calibrated linear form for more reliable extrapolation. Analysis (8) and Table 6 accordingly report accurate predictions on test models from entirely unseen architectures and across both high and low Success Rate regimes, and the forthcoming fully-trained release will further broaden architectural coverage as more open AD models appear. Appendix F.5 further shows that the fixed pipeline remains accurate on AutoMoT and SparseDriveV2, and that rebuilding FastB2D from an entirely independent construction pool that includes a LiDAR-based model preserves its fidelity.

Rare failures on removed routes. Redundancy with respect to aggregate scores does not guarantee that every rare failure is retained. FastB2D mitigates this risk by predicting success on all removed routes, which recovers nearly all of the most failure-prone ones in Appendix F.6, but it cannot guarantee that every safety-critical regression is caught. FastB2D should therefore serve as a development-time proxy, and formal or safety-related conclusions should be drawn on the full benchmark.

Gap between simulation and real-world driving. FastB2D uses the same simulator, scenarios, agents, and metrics as Bench2Drive, so it reduces evaluation cost without widening the existing gap between simulation and real-world driving. Closed-loop simulation remains valuable because it offers a scalable, reproducible, and controlled way to observe the consequences of model actions, while real-world testing provides the final validation before deployment. The compression method itself only requires route-level outcomes and costs, so it can in principle be applied to closed-loop benchmarks built on other simulation paradigms, such as token-space or object-level simulators.

F.2 Can the proposed BCM be used for other closed-loop AD benchmarks?

The compression framework in Section 2 is designed for closed-loop AD benchmarks in general, not for Bench2Drive alone. To validate its generality, we apply the same pipeline to EmBench (linggm3, 2026), a closed-loop AD benchmark that evaluates safe takeover ability under emergency driving conditions.

EmBench exhibits the same structural properties that motivate our method. As Figure 8 shows, per-item simulation costs vary by more than 13× between the longest and shortest items, and a small fraction of expensive routes accounts for the majority of total simulation frames. This cost imbalance

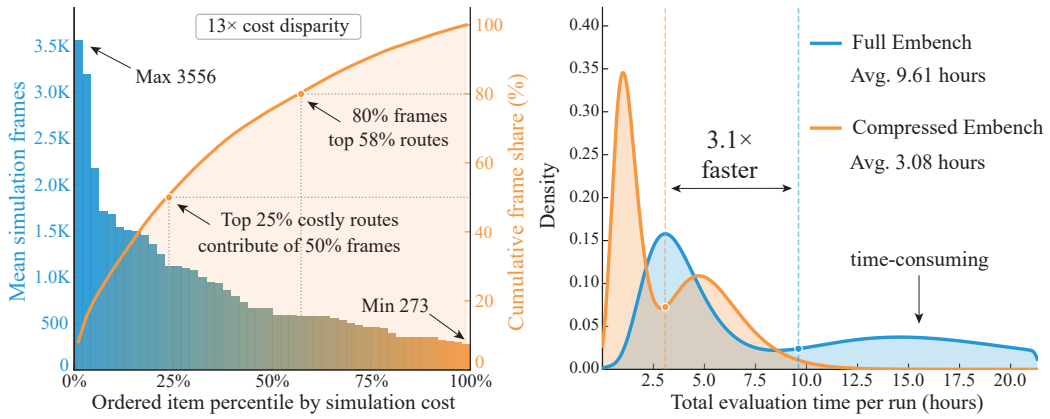

Figure 8: Applying our BCM to EmBench (linggm3, 2026). (a) Per-item frame distribution sorted by cost, showing substantial cost imbalance. (b) Per-run evaluation time before and after compression, showing that the compressed benchmark requires 32% of the evaluation time of the full benchmark.

Table 11: Compression and prediction results of applying our BCM to EmBench (linggm3, 2026). Route Ratio, Frame Ratio, and Time Ratio report the proportion retained after compression. SR MAE and DS MAE report the prediction error on Success Rate and Driving Score, respectively.

	Route Ratio	Frame Ratio	Time Ratio	SR MAE	DS MAE
Compressed EmBench	45%	32.96%	32.05%	1.29%	0.77%

closely mirrors the pattern observed in Bench2Drive, confirming that it is an inherent characteristic of closed-loop AD benchmarks rather than an artifact of any single benchmark design. The resulting compressed benchmark retains 45% of the original routes while reducing frame cost and wall-clock time to roughly 33%, and maintains faithful performance predictions on held-out models for both Success Rate and Driving Score. Table 11 summarizes the compression statistics. As Figure 8 further shows, per-run evaluation time is consistently reduced by roughly 3× after compression, demonstrating that the cost savings translate directly into wall-clock speedup.

These results confirm that both route redundancy and cost imbalance generalize beyond Bench2Drive, and that BCM can exploit them to produce faithful and cost-efficient compressed benchmarks for other closed-loop AD evaluation settings.

F.3 The Relationship Between Wall-Clock Time and Frame Count

This paper uses simulation frame count as the primary measure of evaluation cost. We explain why frame count is more suitable than wall-clock time for cross-model and cross-platform comparison.

Wall-clock time per frame varies widely across configurations. In closed-loop simulation, inference speed depends on model architecture and input resolution. Across all evaluation runs in our dataset, the per-frame time ranges from 0.05 to 2.64 seconds, a difference of more than 50×. Beyond model-level variation, the same model can also exhibit different inference speeds on different hardware platforms due to differences in GPU type and system load. Reporting wall-clock time directly would therefore conflate the inherent cost of a route with both model-specific and hardware-specific factors, making cost comparison across models or across computing environments unreliable.

In contrast, frame count is determined by the simulation and remains stable regardless of which model or hardware is used. Within any single model configuration, frame count and wall-clock time are nearly perfectly correlated. Figure 9(a) shows four representative experiments, each fitted with a linear regression. Despite their different slopes, all four achieve Pearson $r \geq 0.98$, confirming that frame count captures the relative cost structure among routes regardless of absolute inference speed. This pattern holds broadly: across all evaluated models, the mean Pearson correlation exceeds 0.96.

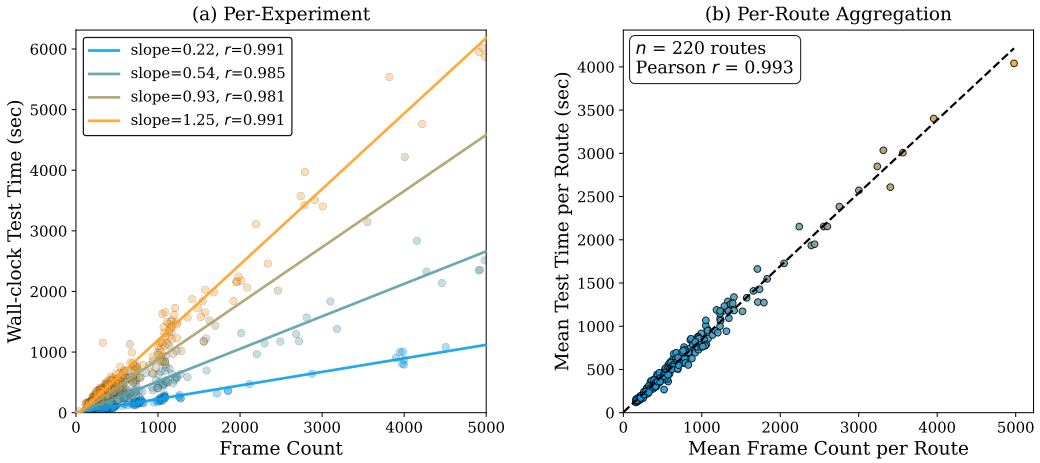

Figure 9: Relationship between frame count and wall-clock test time. (a) Four representative experiments, each showing all route-level data points and a per-experiment linear fit. Slopes differ by model but Pearson $r \geq 0.98$ in every case. (b) Per-route aggregation across all model configurations: mean frame count versus mean test time for each of the 220 routes, with Pearson $r = 0.993$.

The correlation strengthens further when aggregating across models. Figure 9(b) plots mean frame count versus mean wall-clock time for each of the 220 routes, averaged over all model configurations. The Pearson correlation reaches $r = 0.993$, meaning routes with more frames consistently take longer regardless of the model. Frame count therefore serves as a reliable, hardware-independent proxy for evaluation cost that supports fair comparison across both models and computing environments.

F.4 Stability of Per-Route Frame Counts

The preceding analysis in Figure 7 demonstrates that route cost rankings are preserved across experiments. Here we provide complementary evidence by visualizing the frame count distribution for individual routes. For each of the 220 routes, we collect frame counts from all experiment runs and estimate the distribution via kernel density estimation. Figure 10 displays 36 routes spanning short, medium, and long evaluation scenarios. Across all route lengths, the distributions are concentrated and unimodal, confirming that frame count on a given route remains consistent across different models. This stability reflects that simulation frame count is primarily determined by route-level properties such as trajectory length, traffic density, and the number of navigation waypoints. Together with the rank-order consistency shown in Figure 7, this per-route concentration supports the use of average frame count as a robust, model-agnostic measure of evaluation cost.

F.5 Generalization to New Model Families

The test models in Table 2 are openly released models that never enter construction, and Analysis (8) covers models outside the construction score range. We further examine model families that are not part of the construction pool from two angles.

Fixed pipeline on new model families. We apply the FastB2D pipeline without any refitting to AutoMoT (Huang et al., 2026) and SparseDriveV2 (Sun et al., 2026), two recent model families absent from the construction pool. Each model is evaluated over three independent seeds. The top block of Table 12 shows low errors on all four metrics, with Success Rate MAE below 1% for both models.

Rebuilding from an independent construction pool. Exact anchor identities can change across construction pools, because K-Means is stochastic and routes within one response cluster are interchangeable by design. What should remain stable is estimation fidelity. We therefore rebuild FastB2D from three pools of equal size $M = 30$. Original uses 30 records from the original pool, Mixed combines 15 original records with 15 new ones, and New uses 30 records that are entirely independent of the original pool, including records from AutoMoT and SparseDriveV2. AutoMoT

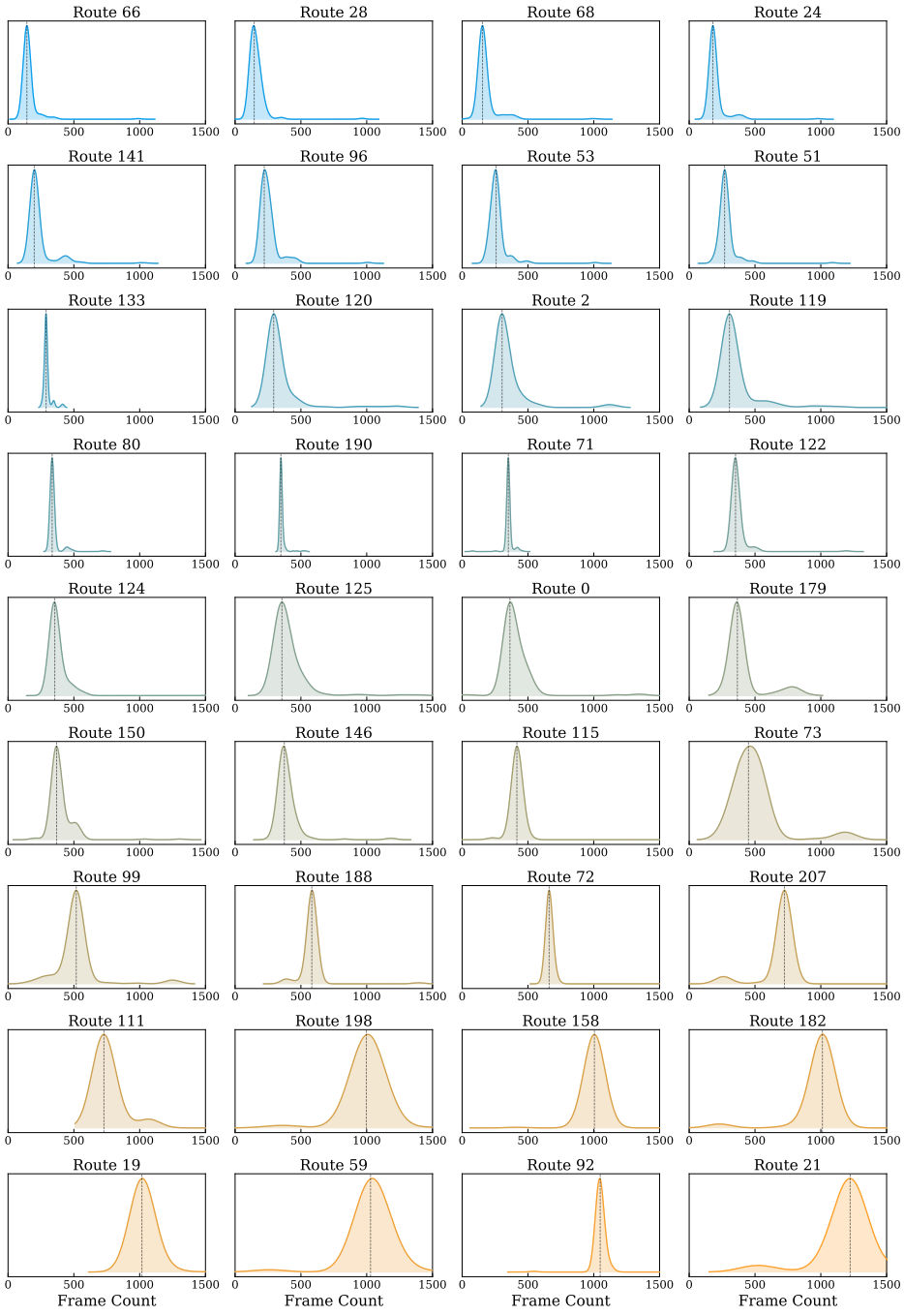

Figure 10: Per-route frame count distributions across all evaluated models. Each subplot shows the kernel density estimate of observed frame counts for a single route. Most routes exhibit a concentrated distribution, illustrating that frame count is a stable, route-intrinsic measure of evaluation cost.

uses LiDAR input and thus also introduces a new sensor suite. For each pool, we relearn the route representations, reselect the anchors, recalibrate the estimators, and evaluate on the same held-out models. The bottom block of Table 12 shows that errors remain close across the three pools, so fidelity is preserved even when the construction records are completely replaced.

Why the variance is higher. The standard deviations in the bottom block are larger than those of the default FastB2D, because each pool contains only 30 records rather than 158. Fewer records

Table 12: Generalization to model families outside the construction pool. **Top:** the fixed FastB2D pipeline applied to new model families over three seeds. **Bottom:** FastB2D rebuilt from construction pools of $M = 30$ records and evaluated on the same held-out models. Eff. reports MRE in %, and the other columns report MAE.

Setting	SR ↓	DS ↓	Eff. ↓	Smooth. ↓
Fixed pipeline on new model families				
AutoMoT (Huang et al., 2026)	0.85±1.06	0.99±1.51	0.65±1.15	1.47±0.10
SparseDriveV2 (Sun et al., 2026)	0.83±0.99	1.64±0.59	2.15±0.62	0.21±1.65
Rebuilt from a construction pool of $M = 30$				
Original: 30 original records	2.32±1.76	1.07±0.98	1.28±0.88	1.67±1.24
Mixed: 15 original + 15 new	2.22±2.15	1.21±0.99	1.53±1.05	1.94±1.39
New: 30 independent records	2.37±2.21	1.38±1.28	1.67±1.22	2.26±1.67

Table 13: Risk screening of the 143 removed routes on the test models. Top $q\%$: the $q\%$ of removed routes that fail most often in full-benchmark evaluation. Recall: the share of these routes recovered by the predicted risk ranking.

Metric	Value	Recovered / Total
ROC AUC on removed routes	0.913	–
Recall of top 25% routes	97.2%	35 / 36
Recall of top 50% routes	94.4%	68 / 72
Recall of top 75% routes	97.2%	104 / 107

yield noisier route parameters, and this noise propagates to both anchor selection and estimation. The trend matches Analysis (2), where error and variance both decrease as M grows and the gains plateau near $M = 50$. When rebuilding FastB2D on a new model pool, we therefore recommend at least 50 construction records for stable performance.

F.6 Risk Screening of Removed Routes

A compressed benchmark could hide failures that occur only on removed routes. FastB2D does not discard these routes. After observing the 77 anchor outcomes, Eq. 3 assigns a predicted success probability to each of the 143 removed routes, which turns them into a ranked list of predicted risk. We evaluate this ranking on the test models. Table 13 reports the ROC AUC for separating failed from successful removed routes, together with the recall of the most failure-prone removed routes at three levels. FastB2D reaches an ROC AUC of 0.913 and recovers 35 of the 36 most failure-prone removed routes. In practice, users can run the 77 anchors for routine evaluation and then inspect the removed routes with the highest predicted risk, which combines efficient testing with targeted failure analysis. Formal and safety-related conclusions should still rely on the full benchmark.

F.7 Rotating Anchor Panels

Any fixed public benchmark can be overfitted, and a public set of 77 anchors is no exception. Because routes within one response cluster are interchangeable, FastB2D can mitigate this risk by periodically rotating the active anchor panel. We construct three alternative anchor panels and report their cost and accuracy in Table 14. All three panels perform comparably to the default FastB2D panel, so rotation does not compromise evaluation fidelity. A hosted evaluator can further keep the active panel hidden.

F.8 Broader Impacts

FastB2D is an evaluation benchmark designed to reduce the computational cost of closed-loop autonomous driving evaluation. It does not introduce new driving capabilities, deploy control policies on real vehicles, or release pretrained driving models. The primary societal effect is positive: by reducing the GPU hours required for reliable benchmarking, FastB2D lowers the energy consumption and carbon footprint of AD research, making rigorous evaluation more accessible to resource-

Table 14: Cost and accuracy of alternative anchor panels. The default row is the 65% removal result of FastB2D in Table 5. Eff. reports MRE in %, and the other error columns report MAE.

Panel	**Cost (%)** ↓	**SR** ↓	**DS** ↓	**Eff.** ↓	**Smooth.** ↓
Default	27.1±1.7	1.67±0.21	0.97±0.14	1.19±0.15	1.41±0.22
Panel 1	27.21	1.577	1.100	1.320	1.289
Panel 2	27.23	1.696	0.977	1.385	1.542
Panel 3	26.48	1.605	0.930	1.415	1.496

constrained groups and accelerating progress toward safer autonomous driving systems. We do not identify direct negative societal impacts arising from this work.

F.9 Responsible Use

Our released assets consist of a compressed route subset and evaluation code for autonomous driving benchmarking, which pose no direct risk for misuse. FastB2D is designed as a lightweight proxy that reduces evaluation time by over 70% for rapid model development and iterative experimentation. For formal reporting or real-world deployment decisions, we recommend that practitioners evaluate on the complete benchmark to ensure comprehensive coverage of all driving scenarios and edge cases. To limit overfitting to a public anchor set, the active panel can be rotated periodically without loss of fidelity, as shown in Appendix F.7.