CASA: CLUSTER-ADAPTIVE SCALING FOR HIERARCHICAL DATA SELECTION IN END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

As autonomous driving datasets continue to grow, efficiently constructing training sets from large-scale data is becoming increasingly important. Effective selection therefore requires considering not only which individual samples are valuable, but also how the training budget should be distributed across different scene types. We introduce CASA, a hierarchical data-selection framework that jointly addresses these two levels. CASA first organizes driving data into behaviorally meaningful clusters by coupling semantic context with future motion, and performs task-aware ranking within each cluster. At the cluster level, CASA learns a structure-aware scaling model with shared parameters across clusters while accounting for shared coverage between related clusters, allowing a small number of probes to guide marginal-utility-based budget allocation. By integrating sample-level selection with structure-aware cluster allocation, CASA provides a unified approach to constructing training sets under varying budgets. On NAVSIM with DiffusionDrive, CASA learns effective scaling relationships from low-budget probes and maintains the highest mean Predictive Driver Model Score (PDMS) among five representative data-selection baselines even when the training budget exceeds four times the largest probe size. This sustained advantage demonstrates CASA's effectiveness in improving data efficiency for end-to-end driving.

1 INTRODUCTION

End-to-end autonomous driving has shifted a growing part of system development from hand-designed modules toward data-driven policy learning. As model capacity increases, the training set becomes an equally important design variable: performance depends not only on how much driving data is available, but also on which data are used for training. Data-selection studies have highlighted the strong dependence of training outcomes on the data selected under limited budgets (Mahmood et al., 2025; Lu et al., 2026; Jiang et al., 2025). This makes efficient training-set construction an important optimization problem rather than merely a preprocessing choice.

Existing approaches typically address this problem at two different levels. Sample-level methods estimate the value of individual examples using uncertainty, task feedback, feature coverage, or information content (Lu et al., 2026; Jiang et al., 2025; Tan et al., 2025), whereas distribution-level methods organize data into domains or clusters and optimize how the training budget is distributed across them (Shen et al., 2025; Vo et al., 2024; Dimlioglu et al., 2026). The latter is particularly appealing when training is expensive, since cluster-level information provides a compact basis for allocating a budget across a large candidate pool. Recent studies have also documented data-scaling behavior in end-to-end driving (Naumann et al., 2025; Zheng et al., 2024). This motivates using a scaling model to estimate the performance gain from allocating additional samples to each cluster and to guide budget allocation according to the predicted marginal gains.

However, driving scenes are difficult to classify using a single source of information. A scene may involve construction, congestion, car following, and pedestrian interaction at the same time. Similar ego trajectories may occur in different road environments, while similar road environments may call for different driving behaviors. Scene classification therefore needs to account for both road semantics and ego motion. Meanwhile, cluster-level budget allocation cannot replace sample-level selection,

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.

since samples within the same cluster can still differ in factors such as trajectory supervision quality. Moreover, autonomous driving scenes can be divided into many fine-grained clusters. Training a planner and fitting a separate scaling law for each cluster would require an increasing number of pilot experiments as the number of clusters grows.

We address these three challenges with CASA, a hierarchical data-selection framework. First, CASA learns scene clusters from a representation that combines road semantics with ego motion, allowing the taxonomy to reflect both scene context and driving behavior. Second, it ranks samples within each cluster by considering trajectory supervision quality and how well the current model handles the corresponding scenes. Third, rather than fitting an independent scaling law for every fine-grained cluster, CASA fits a compact model whose main parameters are shared across clusters, with cluster-specific value and dispersion describing their differences. A pairwise overlap term accounts for coverage shared between clusters. The fitted model then estimates the marginal gain of allocating additional samples to each cluster, guiding budget allocation while the within-cluster ranking determines which samples are selected.

Our main contributions are:

- We formulate end-to-end driving data selection as a hierarchical problem that combines cluster-level composition with within-cluster sample heterogeneity, rather than optimizing either level in isolation.
- We introduce a task-aware intra-cluster ranking scheme built on a motion-constrained scene representation. The ranking uses both planner performance and ground-truth trajectory quality, allowing the selector to favor informative, reliable supervision over low-quality or redundant examples.
- We propose a parameter-shared, overlap-aware cluster scaling law for budget allocation. Cluster value and structural dispersion condition the main-effect curves, while a pairwise term discounts overlapping coverage. Using only seven low-budget probes to fit the scaling model, our selector achieves the highest mean Predictive Driver Model Score (PDMS) on NAVSIM (Dauner et al., 2024) across the evaluated training budgets, including budgets beyond the probe range.

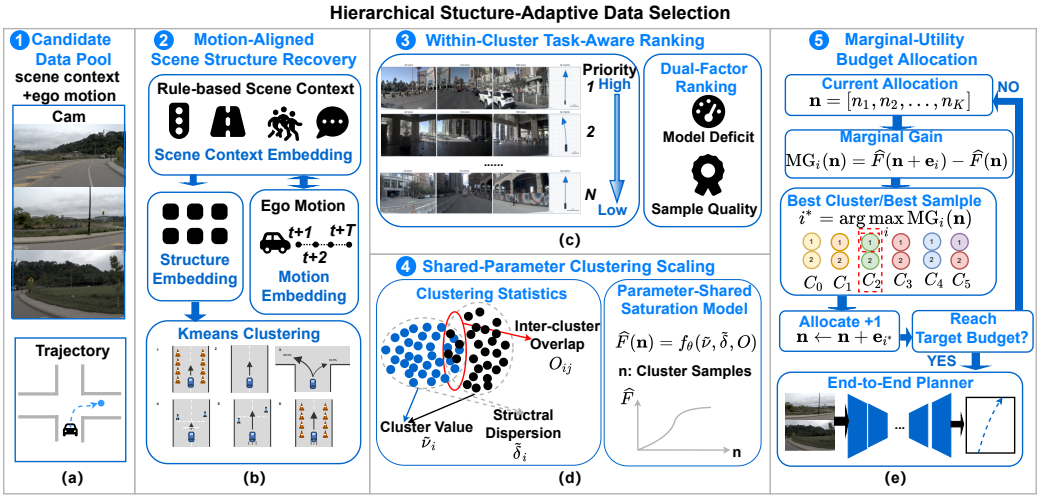

Figure 1: Overview of CASA. (a) Each candidate provides scene context and an ego trajectory. (b) Scene context is aligned with ego motion to recover scene clusters. (c) Candidates are ranked within each cluster using planner performance and supervision quality. (d) Cluster value, structural dispersion, and inter-cluster overlap inform a shared-parameter scaling surrogate. (e) Predicted marginal gains guide budget allocation and produce the selected training set for the end-to-end planner.

2 Related Work

2.1 Data Selection for Autonomous Driving

Data selection for autonomous driving aims to construct effective training sets from large candidate pools under limited training budgets. Existing methods primarily operate at the sample level, estimating the value of individual examples using planning uncertainty, task feedback, learned representations, feature coverage, or information content (Lu et al., 2026; Jiang et al., 2025; Tan et al., 2025). These approaches are effective at distinguishing the utility of individual samples, but their main decision variable remains the sample itself, without explicitly modeling how the overall training budget should be distributed across different types of driving scenes.

A complementary line of work organizes large driving corpora through semantic representations, clustering, or coverage modeling before selection (Vo et al., 2024; Shen et al., 2025; Zhao et al., 2021). Such methods highlight the importance of scene structure for improving diversity and representativeness, but clustering is typically used as a mechanism for data organization or coverage control. In contrast, CASA treats scene clusters as intermediate structural units for hierarchical selection: sample-level ranking is retained within each cluster, while the training budget is further allocated across clusters. Our focus is therefore not on optimizing either sample selection or scene composition in isolation, but on coupling the two decisions within a unified framework.

2.2 Scaling Laws and Data Allocation

Scaling laws provide a compact description of how model performance changes with training-data scale. In autonomous driving, empirical studies have reported systematic performance trends as the amount of training data increases (Naumann et al., 2025; Zheng et al., 2024). Building on this observation, scaling-aware allocation methods use a small number of pilot runs to estimate response behavior and allocate additional data according to predicted marginal gains. MOSAIC, for example, extends this idea to autonomous-driving data composition by fitting domain-specific scaling curves for data allocation (Dimlioglu et al., 2026).

Related data-mixture approaches have also been studied extensively in language-model pretraining, where data composition is treated as an optimization variable and domain value is estimated through proxy models, scaling laws, or adaptive mixture updates (Xie et al., 2023; Shukor et al., 2026; Kang et al., 2024; Chen et al., 2025; Xie et al., 2025; Wang et al., 2026). These studies provide important foundations for performance-driven data allocation, but typically use domains or clusters as the primary units of decision. For autonomous-driving data, this granularity alone does not capture the substantial variation among samples within the same scene cluster. Moreover, when driving scenes are divided into many fine-grained clusters, fitting a separate scaling law for each cluster would require a large number of pilot training runs. CASA therefore adopts a hierarchical formulation: it retains task-aware sample selection within clusters, performs budget allocation across clusters using a parameter-shared scaling model, and preserves shared structure between related clusters. This formulation jointly addresses two complementary questions under limited probing budgets: which samples should be selected, and how much data should be allocated to each scene type?

3 Method

Let $\mathcal{D} = \{x_m\}_{m=1}^{N}$ be the driving corpus. As illustrated in Fig. 1(a), each sample provides scene context, an ego future trajectory, and the information needed to train and evaluate an end-to-end planner. Given a representative seed set $\mathcal{D}_0 \subset \mathcal{D}$, the candidate pool is $\mathcal{P} = \mathcal{D} \setminus \mathcal{D}_0$. For a target training size N_{tgt}, we select $\mathcal{S} \subseteq \mathcal{P}$ with $|\mathcal{S}| = N_{\text{tgt}} - |\mathcal{D}_0| =: B_{\text{add}}$, yielding $\mathcal{D}_{\text{train}} = \mathcal{D}_0 \cup \mathcal{S}$. CASA jointly determines which samples to select and how to allocate this additional budget across scene clusters, as outlined in Fig. 1(b)–(e).

3.1 Motion-aligned scene structure recovery

CASA first organizes the driving corpus into scene clusters, providing the structural units for subsequent within-cluster ranking and budget allocation. To obtain clusters that reflect driving behavior as well as scene context, we use ego motion to guide the recovery of a context-based scene

taxonomy, as illustrated in Fig. 1(b). For each sample x_m, let $\mathbf{a}_m$ denote structured ground-truth scene information and let $\mathbf{y}_m^{\text{ego}}$ denote the ego future trajectory. A deterministic rule function $\mathcal{R}(\cdot)$ converts $\mathbf{a}_m$ into a short scene description,

$$d_m = \mathcal{R}(\mathbf{a}_m), \tag{1}$$

covering road context, nearby vehicles and pedestrians, construction elements, ego state, and coarse interaction cues. We encode d_m with a frozen `all-MiniLM-L6-v2` sentence encoder (Reimers & Gurevych, 2019) and map the resulting 384-dimensional vector $\mathbf{u}_m^{\text{sent}}$ into a learned structural space:

$$\mathbf{h}_m^{\text{ctx}} = \text{LN}\big(\text{GELU}(W_{\text{ctx}}\mathbf{u}_m^{\text{sent}} + b_{\text{ctx}})\big). \tag{2}$$

Here, W_{ctx} and b_{ctx} are trainable parameters; LN denotes layer normalization and GELU denotes the Gaussian error linear unit.

The motion branch takes the ego future trajectory $\mathbf{y}_m^{\text{ego}} = [\mathbf{r}_{m,1}, \ldots, \mathbf{r}_{m,T}]$, where $\mathbf{r}_{m,t} = [x_{m,t}, y_{m,t}, \theta_{m,t}]$ contains the ego vehicle's planar coordinates $(x_{m,t}, y_{m,t})$ and heading angle $\theta_{m,t}$ at future step t, with $T = 8$. After linear projection and positional encoding, a Transformer encoder uses a learnable `[CLS]` token to summarize the trajectory as $\mathbf{h}_m^{\text{mot}}$.

We align context and motion with a symmetric InfoNCE objective. Let $\tilde{\mathbf{h}}_m^{\text{ctx}} = \mathbf{h}_m^{\text{ctx}}/\|\mathbf{h}_m^{\text{ctx}}\|_2$ and $\tilde{\mathbf{h}}_m^{\text{mot}} = \mathbf{h}_m^{\text{mot}}/\|\mathbf{h}_m^{\text{mot}}\|_2$ denote the ℓ_2-normalized context and motion representations. The context-to-motion term is

$$\mathcal{L}_{\text{ctx}\to\text{mot}} = -\frac{1}{N_b}\sum_{m=1}^{N_b} \log \frac{\exp\Big(\text{sim}(\tilde{\mathbf{h}}_m^{\text{ctx}}, \tilde{\mathbf{h}}_m^{\text{mot}})/\tau_{\text{nce}}\Big)}{\sum_{r=1}^{N_b}\exp\Big(\text{sim}(\tilde{\mathbf{h}}_m^{\text{ctx}}, \tilde{\mathbf{h}}_r^{\text{mot}})/\tau_{\text{nce}}\Big)}. \tag{3}$$

Here, N_b is the batch size and τ_{nce} is the InfoNCE temperature, fixed at 0.1. with a symmetric $\mathcal{L}_{\text{mot}\to\text{ctx}}$ term and $\mathcal{L}_{\text{align}} = (\mathcal{L}_{\text{ctx}\to\text{mot}} + \mathcal{L}_{\text{mot}\to\text{ctx}})/2$.

The final structural representation is obtained by a learnable linear projection of the context feature:

$$\mathbf{h}_m = \text{LN}\big(W_\phi \mathbf{h}_m^{\text{ctx}} + b_\phi\big), \tag{4}$$

where W_ϕ and b_ϕ are trainable parameters.

During representation learning, K learnable prototypes provide an auxiliary clustering signal. Let Y denote the resulting soft prototype assignment and $\mathbf{h}$ the corresponding structural representation. We combine an information-maximization objective $\mathcal{L}_{\text{cluster}} = H(Y \mid \mathbf{h}) - H(Y)$ with an orthogonality penalty on normalized prototypes, $\mathcal{L}_{\text{ortho}}$. The overall objective is

$$\mathcal{L}_{\text{structure}} = \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{cluster}}\mathcal{L}_{\text{cluster}} + \lambda_{\text{ortho}}\mathcal{L}_{\text{ortho}}. \tag{5}$$

After training, we normalize $\mathbf{h}_m$ and run KMeans to obtain the fixed scene taxonomy used by all later stages.

3.2 Within-cluster task-aware ranking

Given the recovered scene clusters, CASA ranks candidates within each cluster (Fig. 1(c)). A large planner error may indicate either a difficult but valuable example or an unreliable ground-truth trajectory. We therefore use PDMS to assess both planner performance and supervision quality.

For $x_m \in C_i$, let ρ_m^{human} be the PDMS of the recorded human trajectory and ρ_m^{pred} the PDMS of the trajectory predicted by a planner trained on $\mathcal{D}_0$. Define the planner–human PDMS difference

$$\Delta\rho_m = \rho_m^{\text{pred}} - \rho_m^{\text{human}}. \tag{6}$$

Because clusters have different intrinsic difficulty, we standardize this quantity within each cluster:

$$\gamma_{m|i} = \frac{\Delta\rho_m - \mu_{\Delta\rho,i}}{\sigma_{\Delta\rho,i}}. \tag{7}$$

A negative $\gamma_{m|i}$ means that the planner performs relatively poorly on this sample compared with other samples in the same cluster. We also compute the first quartile ϑ_i of ρ_m^{human} within the candidate portion of C_i as a local quality threshold.

The CASA sample score is

$$\psi_m^{\mathrm{CASA}} = \begin{cases} \gamma_{m|i} \left(\vartheta_i - \rho_m^{\mathrm{human}} \right), & \gamma_{m|i} > 0, \\ \gamma_{m|i} \left(\rho_m^{\mathrm{human}} - \vartheta_i \right), & \gamma_{m|i} \leq 0. \end{cases} \tag{8}$$

Sorting samples in each cluster by ascending CASA score gives the ranking π_i for cluster C_i, with smaller scores receiving higher priority. In particular, a sample on which the planner is relatively weak ($\gamma_{m|i} < 0$) but whose recorded trajectory is above the cluster's low-quality quartile receives a negative score and is promoted. Samples with ground-truth quality below ϑ_i are de-prioritized regardless of the sign of $\gamma_{m|i}$.

3.3 Shared-parameter cluster scaling

Once candidates are ranked within each cluster, the remaining question is how to allocate additional data across clusters. We compute cluster value prior, structural dispersion, and inter-cluster overlap from the clustered data and rankings, and use them as descriptors in a shared-parameter scaling surrogate (Fig. 1(d)).

Cluster value prior. Let $M_i = |C_i \cap \mathcal{P}|$ denote the candidate count in cluster i. The set $\mathcal{Q}_i$ contains the top $L_i = \max(1, \lceil 0.2 M_i \rceil)$ candidates according to the ranking in Section 3.2. We define

$$\tilde{\nu}_i = -\frac{1}{L_i} \sum_{x_m \in \mathcal{Q}_i} \psi_m^{\mathrm{CASA}}, \tag{9}$$

Standardizing $\tilde{\nu}_i$ across clusters gives the cluster value prior ν_i. A larger ν_i indicates a greater concentration of high-priority candidates in cluster C_i.

Structural dispersion. Let $\bar{\mathbf{h}}_m$ be the normalized structural embedding and $\bar{\mathbf{c}}_i$ the normalized KMeans center of C_i. The raw dispersion is

$$\tilde{\delta}_i = \frac{1}{|C_i|} \sum_{x_m \in C_i} \|\bar{\mathbf{h}}_m - \bar{\mathbf{c}}_i\|_2, \tag{10}$$

Standardizing $\tilde{\delta}_i$ across clusters gives the structural dispersion δ_i.

Inter-cluster overlap. KMeans assigns each sample to a single cluster, although a driving scene may share characteristics with neighboring clusters. To capture this overlap, we compute an approximate soft assignment

$$\omega_{m,i} = \frac{\exp(-\|\mathbf{h}_m - \mathbf{c}_i\|_2^2 / \tau_{\mathrm{soft}})}{\sum_{j=1}^{K} \exp(-\|\mathbf{h}_m - \mathbf{c}_j\|_2^2 / \tau_{\mathrm{soft}})} \tag{11}$$

Here, $\omega_{m,i}$ is the soft membership weight of sample x_m in cluster C_i, with $\sum_{i=1}^{K} \omega_{m,i} = 1$. τ_{soft} controls the smoothness of the distance-based assignments: larger values distribute weight more evenly across clusters. We fix $\tau_{\mathrm{soft}} = 20$ throughout.

Collecting the soft membership weights of all N_i samples for cluster C_i, we define $\boldsymbol{\omega}_i = [\omega_{1,i}, \ldots, \omega_{N_i,i}]^\top$. For $i \neq j$, the overlap coefficient is

$$O_{ij} = \max\left(0, \mathrm{Corr}(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j)\right), \tag{12}$$

with $O_{ii} = 1$ in the stored overlap matrix $\mathbf{O}$. Negative correlations are not treated as overlap and are clipped to zero.

Parameter-shared saturation model. We use a Hill-type curve to capture diminishing returns as more samples are added to a cluster. Its asymptotic main-effect scale is

$$G_i^{\infty} = A \exp(\kappa_\nu \nu_i). \tag{13}$$

Here, $A > 0$ is a shared amplitude fitted from the probes, while κ_ν controls how the standardized cluster value prior ν_i affects this scale. We fix $\kappa_\nu = 0.25$ throughout. The half-saturation point of cluster i is

$$n_{1/2,i} = n_{1/2,0} \exp(\kappa_\delta \delta_i). \tag{14}$$

Here, $n_{1/2,0} > 0$ is a shared baseline, and the fitted coefficient κ_δ determines how structural dispersion δ_i shifts this count. At $n_i = n_{1/2,i}$, the cluster reaches half of its asymptotic contribution. Its normalized coverage is

$$g_i(n_i) = \frac{n_i^\alpha}{n_{1/2,i}^\alpha + n_i^\alpha}, \tag{15}$$

where n_i is the number of selected samples from cluster i, and the shared, fitted exponent $0.05 \leq \alpha \leq 1$ controls the shape of the saturation curve.

Let $\mathbf{g}(\mathbf{n}) = (g_i(n_i))_{i=1}^K$ and $\mathbf{G}^\infty = (G_i^\infty)_{i=1}^K$. Combining the cluster contributions with an overlap discount gives the performance surrogate

$$\widehat{F}(\mathbf{n}) = (\mathbf{G}^\infty)^\top \mathbf{g}(\mathbf{n}) - \frac{\eta}{2} \sum_{i \neq j} O_{ij} g_i(n_i) g_j(n_j). \tag{16}$$

The first term aggregates the clusters' saturating contributions. The second discounts coverage shared by overlapping clusters, with the fitted coefficient $\eta \geq 0$ controlling its strength; the factor $1/2$ accounts for each cluster pair appearing twice in the sum.

Given N_{probe} probe configurations $\{(\mathbf{n}^{(r)}, \rho_{\text{obs}}^{(r)})\}_{r=1}^{N_{\text{probe}}}$, where $\rho_{\text{obs}}^{(r)}$ is the observed probe PDMS, we fit the five shared parameters

$$\boldsymbol{\theta} = \{A, \alpha, n_{1/2,0}, \kappa_\delta, \eta\} \tag{17}$$

by bounded nonlinear least squares:

$$\mathcal{L}_{\text{fit}} = \sum_{r=1}^{N_{\text{probe}}} \left(\widehat{F}(\mathbf{n}^{(r)}; \boldsymbol{\theta}) - \rho_{\text{obs}}^{(r)} \right)^2 + (\lambda_\delta \kappa_\delta)^2 + (\lambda_\eta \eta)^2. \tag{18}$$

The last two terms regularize the dispersion and overlap coefficients, with $\lambda_\delta = \lambda_\eta = 0.002$. The search ranges are $A \in [0.01, 1.50]$, $\alpha \in [0.05, 1]$, $n_{1/2,0} \in [100, 20000]$, $\kappa_\delta \in [-2, 2]$, and $\eta \in [0, 2]$. The descriptors ν_i, δ_i, and $\mathbf{O}$ are computed from the clustered data and remain fixed during fitting.

3.4 Marginal-utility budget allocation

After fitting the cluster-composition surrogate, we use it to decide which cluster should receive each additional sample (Fig. 1(e)). The predicted gain from adding one sample to cluster i is

$$\mathrm{MG}_i(\mathbf{n}) = \widehat{F}(\mathbf{n} + \mathbf{e}_i) - \widehat{F}(\mathbf{n}), \tag{19}$$

Here, $\mathbf{e}_i$ adds one to the i-th entry of $\mathbf{n}$. Starting from the seed set, we repeatedly choose the cluster with the largest MG_i among those with remaining candidates and add its next-ranked sample from π_i. This produces nested training sets across budgets.

4 Experiments

4.1 Setup

Dataset and metric. Training candidates are drawn from `navtrain`, and PDMS is the primary evaluation metric. A 2,000-sample subset of `NavTest`, denoted Quick2000, is used only to evaluate the fitting probes. The remaining 10,146 `NavTest` samples are used for the main comparisons, ablations, interpolation results, and cross-backbone evaluation.

Planner and scene representation. We use DiffusionDrive (Liao et al., 2025) as the downstream planner. All selection methods use the same planner architecture and optimization settings; only their selected training data differ. The scene representation combines semantic context with the ego future trajectory. We use six scene clusters, selected by comparing geometric separation and maneuver consistency across candidate cluster counts. Planner settings, representation details, and the cluster-count comparison are provided in Appendix B.1.

Seed set and probes. For each cluster, we choose the 80 samples closest to the KMeans center, giving a balanced seed set of 480 samples. A base planner trained on this set is used to score the remaining candidates. We construct seven low-budget probes with different cluster compositions, measure their PDMS on Quick2000, and use these measurements to fit the scaling law. The largest probe contains 700 training samples. Probe compositions and construction details are given in Appendix Table 6 and Section B.4.

Baselines. We compare against five alternatives. **Random** shuffles the candidate pool and selects samples in that order. **Uncertainty**, adapted from uncertainty-based active learning (Joshi et al., 2009), ranks candidates by the Shannon entropy of DiffusionDrive's trajectory-mode distribution. **Core-set** (Sener & Savarese, 2018) applies greedy k-center selection in our learned scene-representation space. **MOSAIC** (Dimlioglu et al., 2026) allocates samples according to predicted per-cluster marginal gains. For **Chameleon** (Xie et al., 2025), we apply its kernel-ridge leverage-score-based weighting to our cluster representations to obtain allocation weights. All methods start from the same 480-sample seed set. For the main experiments, every method uses planner-training random seeds 0, 1, and 2; we report the mean and standard deviation of PDMS across these three runs.

4.2 Main results: selection beyond the probe range

Figure 2 shows planning performance on the evaluation set across training budgets, separating the 600–700-sample probe-size range from the larger evaluation budgets. Fitted from seven low-budget probes, the same five-parameter scaling law guides selection at all eight unseen budgets from 800 to 3000 samples. Table 1 shows that CASA achieves the highest mean PDMS at every budget. It leads the strongest baseline by 0.0438 PDMS at 800 samples and 0.0404 at 1300 samples, and retains an advantage at 3000 samples (0.6612 versus 0.6488), more than four times the largest probe size.

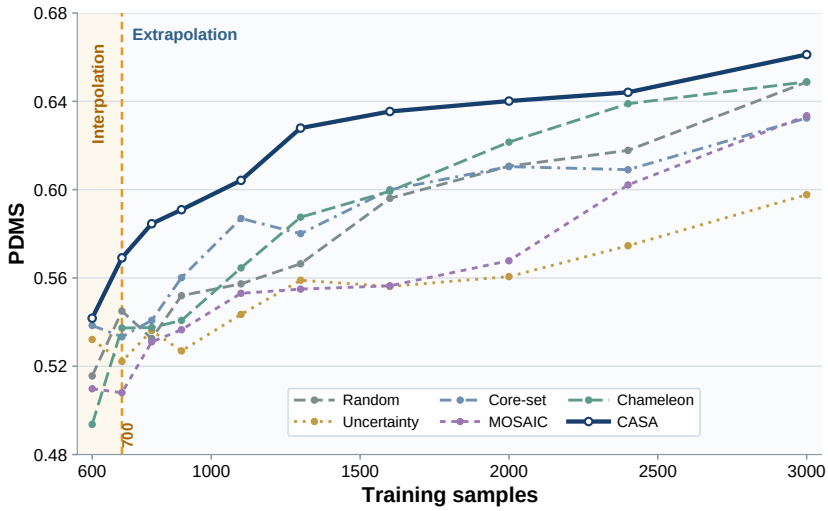

Figure 2: Planning performance on the evaluation set across training-data budgets. The 600–700-sample range covers the probe sizes used to fit CASA's scaling law; larger budgets are unseen during fitting. Curves show mean PDMS for CASA and the five baselines.

The cross-budget comparisons make the sample-efficiency gain especially clear. CASA reaches 0.5909 PDMS with 900 samples, exceeding the strongest baseline at 1300 samples (0.5875); with 1300 samples, it reaches 0.6279, exceeding the strongest baseline at 2000 samples (0.6215). These results show that CASA selects training sets that achieve stronger planning performance with fewer samples across the evaluated budgets.

4.3 Ablation study: complementary contributions

Table 2 shows that every component contributes to CASA's performance at the 1300-sample budget. The complete framework achieves 0.6279 mean PDMS, compared with 0.5947 using only trajectory

Table 1: NAVSIM planning performance across data budgets. The scaling law is fitted only using probes at 600 and 700 samples; budgets of 800 samples and above are unseen during fitting. Values are mean PDMS ± standard deviation over three planner-training seeds on the evaluation set. Best mean in bold.

	Within probe-size range		*Unseen budgets*		
Method	600	700	800	900	1100
Random	0.5156±0.0092	0.5450±0.0072	0.5325±0.0054	0.5520±0.0149	0.5573±0.0131
Uncertainty	0.5321±0.0012	0.5222±0.0089	0.5361±0.0090	0.5269±0.0064	0.5435±0.0098
Core-set	0.5385±0.0010	0.5333±0.0069	0.5408±0.0056	0.5601±0.0079	0.5869±0.0128
MOSAIC	0.5098±0.0067	0.5080±0.0089	0.5310±0.0052	0.5365±0.0043	0.5530±0.0115
Chameleon	0.4937±0.0115	0.5373±0.0101	0.5375±0.0160	0.5408±0.0146	0.5646±0.0152
CASA	**0.5418**±0.0055	**0.5691**±0.0127	**0.5846**±0.0167	**0.5909**±0.0037	**0.6042**±0.0136

	Unseen budgets				
Method	1300	1600	2000	2400	3000
Random	0.5664±0.0155	0.5960±0.0090	0.6106±0.0008	0.6178±0.0119	0.6487±0.0067
Uncertainty	0.5590±0.0041	0.5562±0.0055	0.5606±0.0058	0.5746±0.0033	0.5977±0.0193
Core-set	0.5800±0.0154	0.5999±0.0090	0.6104±0.0132	0.6090±0.0168	0.6325±0.0141
MOSAIC	0.5549±0.0086	0.5564±0.0065	0.5678±0.0101	0.6021±0.0015	0.6335±0.0068
Chameleon	0.5875±0.0147	0.5993±0.0066	0.6215±0.0123	0.6389±0.0198	0.6488±0.0108
CASA	**0.6279**±0.0038	**0.6354**±0.0033	**0.6401**±0.0054	**0.6441**±0.0078	**0.6612**±0.0098

information and 0.6099 using only semantic information. Their combination provides scene clusters that reflect both driving behavior and scene context. On this structure, task-aware ranking identifies which samples to select: removing CASA ranking produces the largest drop, to 0.5835.

The allocation components make distinct contributions as well. Removing the inter-cluster overlap matrix $\mathbf{O}$ reduces PDMS to 0.6174, while removing the cluster value prior ν_i or structural dispersion δ_i yields 0.6163 and 0.6182, respectively. Together, the ablations support the design of the full pipeline: complementary scene cues establish the clusters, task-aware ranking prioritizes samples within them, and cluster descriptors and overlap guide how the budget is distributed.

Table 2: Ablation study at the 1300-sample training budget. Values are mean PDMS ± standard deviation over three planner-training seeds on the evaluation set. The best mean is shown in bold.

Configuration	PDMS
Trajectory only	0.5947 ± 0.0195
Semantic only	0.6099 ± 0.0114
Without CASA ranking	0.5835 ± 0.0067
Without $\mathbf{O}$	0.6174 ± 0.0035
Without ν_i	0.6163 ± 0.0050
Without δ_i	0.6182 ± 0.0128
CASA	$\mathbf{0.6279} \pm 0.0039$

4.4 Adaptive cluster allocation

CASA's cluster-level allocation considers cluster value, structural dispersion, and inter-cluster overlap jointly, rather than ranking clusters by any single statistic. In Table 3, C_0 has the highest value prior and receives the most samples (344), whereas C_5 has a low value prior and remains at its initial 80 samples. These two clusters also have the only retained nonzero overlap in the table, so the model accounts for their shared coverage when allocating additional samples to either cluster. Structural dispersion affects how each cluster's fitted gain saturates as more samples are added. Although C_2 has a higher value prior than C_3, their dispersion values differ substantially, and C_2 ultimately receives fewer samples (216 versus 261). These results illustrate how the three statistics jointly inform budget

allocation; the ablations in Table 2 further show that removing any one of them reduces planning performance.

Table 3: Cluster statistics and CASA's allocation at the 1300-sample training budget. The cluster value prior ν_i and structural dispersion δ_i are standardized across clusters; mean overlap $\bar{O}_i$ averages the five off-diagonal entries of $\mathbf{O}$ for each cluster.

Cluster	Selected n_i	Cluster value prior ν_i	Structural dispersion δ_i	Mean overlap $\bar{O}_i$
C_0	344	1.5637	−0.8925	0.0965
C_1	205	0.0185	1.4556	0
C_2	216	0.7953	1.3309	0
C_3	261	0.0389	−0.3542	0
C_4	194	−1.2294	−0.7363	0
C_5	80	−1.1871	−0.8036	0.0965

5 Conclusion

In this work, we present a hierarchical data-selection framework for end-to-end autonomous driving that jointly considers sample-level quality and cluster-level allocation. By combining fine-grained sample ranking with scaling-aware budget allocation across scene clusters, our method consistently improves data efficiency under constrained training budgets. The experimental results demonstrate that effective data selection should not only identify informative individual samples, but also account for how training data are distributed across heterogeneous and structurally related driving scenarios.

Beyond the specific selection strategy, the broader value of our framework lies in bridging two previously separate perspectives: sample-level selection and cluster-level scaling. Rather than treating clusters as completely independent domains, we further introduce a parameter-sharing mechanism across clusters, allowing statistically related scene groups to jointly constrain the estimation of their scaling behavior. This is particularly important for autonomous driving, where increasingly fine-grained scene partitioning can easily lead to insufficient observations within individual clusters. By sharing structural information across clusters, the proposed framework can recover useful scaling trends from only a limited number of probe experiments, substantially reducing the experimental cost required to characterize data utility. This provides a practical way to apply scaling-law-based data allocation to large and finely structured autonomous-driving datasets.

The framework is also modular and interpretable. Its main components—scene clustering, within-cluster sample scoring, cluster-level scaling estimation, and budget allocation—can each be optimized independently. Future work will explore stronger methods for discovering scene structure, more reliable task-aware sample scoring functions, and cluster-level parameters that better characterize data value and transferability, with the aim of further improving data selection.

AI use statement

Generative AI tools were used to assist with language editing, code debugging and figure refinement. The authors reviewed and verified the material and take full responsibility for the mathematical claims, experimental results, citations, and final manuscript.

Reproducibility statement

The full selection rule, cluster statistics, scaling-law parameterization, parameter bounds, probe configurations, planner-training settings, and random-seed protocol are specified in Sections 3–4 and in the appendix.

REFERENCES

Mayee Chen, Michael Hu, Nicholas Lourie, Kyunghyun Cho, and Christopher Ré. Aioli: A unified optimization framework for language model data mixing. In *International Conference on Learning Representations*, volume 2025, pp. 46089–46132, 2025.

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, 2022.

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.

Tolga Dimlioglu, Nadine Chang, Maying Shen, Rafid Mahmood, and Jose M Alvarez. Scaling-aware data selection for end-to-end autonomous driving systems. *arXiv preprint arXiv:2604.08366*, 2026.

Wenhao Jiang, Duo Li, Menghan Hu, Chao Ma, Ke Wang, and Zhipeng Zhang. Active learning from scene embeddings for end-to-end autonomous driving. *arXiv preprint arXiv:2503.11062*, 2025.

Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pp. 2372–2379. IEEE, 2009.

Feiyang Kang, Yifan Sun, Bingbing Wen, Si Chen, Dawn Song, Rafid Mahmood, and Ruoxi Jia. Autoscale: Scale-aware data mixing for pre-training llms. *arXiv preprint arXiv:2407.20177*, 2024.

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

Han Lu, Xiaosong Jia, Yichen Xie, Siyu Sun, Wenlong Liao, Xiaokang Yang, and Junchi Yan. Activead: Planning-oriented active learning for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3656–3666, 2026.

Rafid Mahmood, James Lucas, Jose M Alvarez, Sanja Fidler, and Marc T Law. Optimizing data collection for machine learning. *Journal of Machine Learning Research*, 26(38):1–52, 2025.

Alexander Naumann, Xunjiang Gu, Tolga Dimlioglu, Mariusz Bojarski, Alperen Degirmenci, Alexander Popov, Devansh Bisla, Marco Pavone, Urs Muller, and Boris Ivanovic. Data scaling laws for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2596–2607, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: Acore-set approach. *stat*, 1050:21, 2018.

Maying Shen, Nadine Chang, Sifei Liu, and Jose M Alvarez. Sse: Multimodal semantic data selection and enrichment for industrial-scale data assimilation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2525–2535, 2025.

Mustafa Shukor, Louis Bethune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. Scaling laws for optimal data mixtures. *Advances in Neural Information Processing Systems*, 38:129554–129579, 2026.

Haoru Tan, Sitong Wu, Wei Huang, Shizhen Zhao, and Xiaojuan Qi. Data pruning by information maximization. In *International Conference on Learning Representations*, volume 2025, pp. 50133–50155, 2025.

Huy V Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, Camille Couprie, Maxime Oquab, Armand Joulin, et al. Automatic data curation for self-supervised learning: A clustering-based approach. *arXiv preprint arXiv:2405.15613*, 2024.

Yifan Wang, Fengze Liu, Yuanfan Guo, Jiyao Deng, Xuecheng Wu, Weidong Zhou, Xiaohuan Zhou, Taifeng Wang, et al. Tikmix: Efficient semi-dynamic data mixture via data influence for llm pre-training. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5777–5793, 2026.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

Wanyun Xie, Francesco Tonin, and Volkan Cevher. Chameleon: A flexible data-mixing framework for language model pretraining and finetuning. *arXiv preprint arXiv:2505.24844*, 2025.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3709–3718, 2026.

Jinxin Zhao, Jin Fang, Zhixian Ye, and Liangjun Zhang. Large scale autonomous driving scenarios clustering with self-supervised feature extraction. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 473–480. IEEE, 2021.

Yupeng Zheng, Pengxuan Yang, Zhongpu Xia, Qichao Zhang, Yuhang Zheng, Songen Gu, Bu Jin, Teng Zhang, Ben Lu, Chao Han, et al. Data scaling laws for imitation learning-based end-to-end autonomous driving. *arXiv preprint arXiv:2412.02689*, 2024.

A Additional Experimental Results

Table 4: Final number of training samples allocated to each scene cluster for the NAVSIM results evaluated on NavTest excluding Quick2000. Cluster labels C_0–C_5 follow the taxonomy used throughout the paper.

Training size	C_0	C_1	C_2	C_3	C_4	C_5
600	106	118	124	92	80	80
700	146	133	140	121	80	80
800	187	148	155	150	80	80
900	218	160	168	173	101	80
1100	281	182	192	217	148	80
1300	344	205	216	261	194	80

Table 4 shows that CASA does not simply balance the six clusters. Cluster C_2 is largest at 600 samples, after which C_0 receives the largest allocation at every reported size from 700 to 1300; C_5 never grows beyond its initial 80 examples. C_1 and C_2 receive substantial data early, C_3 continues to grow at larger sizes, and C_4 begins to receive additional examples only after the higher-value clusters have partially saturated.

This behavior follows directly from Eq. 16. A cluster can be attractive because it has a strong task-value prior, because its inferred saturation scale leaves room for further coverage, or because it overlaps little with already-covered clusters. Conversely, a low-value cluster that overlaps a better-covered region can have little marginal utility even if its raw sample count is small. The allocation therefore reflects a changing marginal-return calculation rather than a fixed cluster ranking.

B Additional Method Details

B.1 Experimental setup and evaluation splits

The training candidate pool is drawn from `navtrain`; training and evaluation sets do not share frames. NavTest contains 12,146 evaluation samples. A fixed 2,000-sample subset, Quick2000, is used to measure PDMS for all seven scaling-law probes. All reported NAVSIM planning results, including the main selection comparison, ablation study, 600–700-sample interpolation results, and TransFuser comparison, use the remaining 10,146 NavTest samples.The allocation table uses display labels C_0–C_5, corresponding to the theoretical indices C_0–C_5 in Section 3.

DiffusionDrive planner training uses batch size 32, an initial learning rate of 1.5×10^{-4}, cosine decay to 10^{-6}, five warm-up epochs, and 75 total epochs. The scene representation maps the frozen 384-dimensional sentence embedding to 64 dimensions and uses a two-layer Transformer for the eight-step ego trajectory. It is trained for 50 epochs with AdamW, batch size 256, learning rate 10^{-3}, weight decay 10^{-4}, InfoNCE temperature 0.1, and gradient clipping at 1.0.

Cluster-count sensitivity. We compare $K \in \{6, 8, 10, 12, 16\}$ using 14,951 scenes. For each K, we independently train the representation model for 50 epochs with seed 0, ℓ_2-normalize the resulting structural embeddings, and apply KMeans. We assess cluster compactness and separation with cosine Silhouette, and compare the clusters with ego-maneuver labels using adjusted Rand index (ARI) and normalized mutual information (NMI). ARI measures agreement between the two partitions, while NMI measures how much maneuver information the clusters retain.

Table 5: Sensitivity to the number of scene clusters. Higher values indicate better geometric separation or stronger agreement with ego-maneuver labels. Each representation model is trained independently for its corresponding K.

K	Cosine Silhouette ↑	Maneuver ARI ↑	Maneuver NMI ↑
6	**0.925**	**0.202**	**0.494**
8	0.869	0.157	0.456
10	0.790	0.147	0.397
12	0.885	0.168	0.392
16	0.737	0.113	0.423

As shown in Table 5, $K = 6$ achieves the highest value on all three metrics. The six-cluster representation therefore provides both clear geometric separation and the strongest retained maneuver information among the tested choices. It also gives the 480-example balanced seed set 80 examples per cluster, providing a common starting point for subsequent cluster-level statistics. We fix $K = 6$ for the remaining experiments. All methods start from the same seed set. Planner-training randomness is evaluated with seeds 0, 1, and 2; the single fixed shuffle used for the Random baseline is separate from those planner-training seeds.

Table 6: Low-budget probe configurations used to fit the five shared scaling-law parameters.

Run	Size	Composition strategy
0	480	balanced seed set (80 per cluster)
1	600	uniform expansion
2	600	geometry-prior nonuniform expansion
3	700	uniform expansion
4	700	geometry-prior nonuniform expansion
5	700	random nonuniform expansion
6	700	high-overlap-pair expansion

B.2 Local statistics for the CASA score

For cluster C_i, the local mean and population standard deviation of the planner–human PDMS difference are

$$\mu_{\Delta\rho,i} = \frac{1}{|C_i|} \sum_{x_m \in C_i} \Delta\rho_m, \tag{20}$$

$$\sigma_{\Delta\rho,i} = \left[\frac{1}{|C_i|} \sum_{x_m \in C_i} (\Delta\rho_m - \mu_{\Delta\rho,i})^2 \right]^{1/2}. \tag{21}$$

If $\sigma_{\Delta\rho,i}$ falls below a numerical tolerance, all standardized values in that cluster are set to zero. The quality threshold is computed only from the candidate portion of the cluster,

$$\vartheta_i = \mathrm{Quantile}_{0.25} \left(\{ \rho_m^{\mathrm{human}} : x_m \in C_i \cap \mathcal{P} \} \right). \tag{22}$$

This local normalization avoids systematically treating intrinsically difficult scene clusters as low-quality simply because their absolute PDMS distribution is lower.

B.3 Cluster-prior normalization

The raw cluster value $\tilde{\nu}_i$ and raw dispersion $\tilde{\delta}_i$ are standardized across the K clusters using population standard scores, yielding ν_i and δ_i, respectively. If the cross-cluster standard deviation of either statistic falls below a numerical tolerance, the corresponding normalized statistic is set to zero for every cluster. These safeguards prevent unstable scaling parameters when the derived priors contain little between-cluster variation.

B.4 Probe construction

The uniform 600-sample probe expands each cluster from 80 to 100 examples. The 700-sample uniform probe uses the composition $[117, 117, 117, 117, 116, 116]^\top$. Geometry-prior probes compute

$$\chi_i = \nu_i + \delta_i - \bar{O}_i, \qquad \bar{O}_i = \frac{1}{K-1} \sum_{j \neq i} O_{ij}, \tag{23}$$

and convert χ_i into allocation proportions with

$$\varpi_i = \frac{\exp(\chi_i / \tau_{\mathrm{alloc}})}{\sum_j \exp(\chi_j / \tau_{\mathrm{alloc}})}, \qquad \tau_{\mathrm{alloc}} = 3.0. \tag{24}$$

Integer allocations use a largest-remainder correction while respecting candidate-pool capacity. The random nonuniform probe is generated from a fixed six-dimensional uniform random vector and normalized in the same way. These two nonuniform families are included to avoid fitting the scaling law from a single perturbation direction.

For the high-overlap-pair probe, 70% of the added budget is directed to the pair with the largest O_{ij}, split equally between its two clusters, and the remaining 30% is distributed across the other four clusters. The probe configurations are used to fit the scaling law; the final selection comparisons use the methods and budgets reported in the main results.

C Interpolation Results at Observed Data Sizes

The 600- and 700-sample evaluations on NavTest excluding Quick2000 are within the total-size range used by the probes and are therefore not evidence of budget extrapolation. They are included here for completeness. Importantly, the CASA compositions at these sizes are re-optimized after fitting the scaling law and are not identical to the pre-defined probe compositions.

D Baseline Details

Random. A single fixed shuffle of $\mathcal{P}$ defines the selection sequence. Data-selection randomness is kept separate from the three planner-training seeds, so uncertainty from the two sources is not conflated.

Table 7: PDMS on NavTest excluding Quick2000 at 600 and 700 training samples. These budgets lie within the probe-size range and are reported as interpolation diagnostics.

Method	600	700
Random	0.5156 ± 0.0092	0.5450 ± 0.0072
Uncertainty	0.5321 ± 0.0012	0.5222 ± 0.0089
Core-set	0.5385 ± 0.0010	0.5333 ± 0.0069
MOSAIC	0.5098 ± 0.0067	0.5080 ± 0.0089
Chameleon	0.4937 ± 0.0115	0.5373 ± 0.0101
CASA	$\mathbf{0.5418} \pm 0.0055$	$\mathbf{0.5691} \pm 0.0127$

Uncertainty. For planner trajectory-mode probabilities $\boldsymbol{\phi}_m = [\phi_{m,1}, \ldots, \phi_{m,J}]$, we use

$$\mathcal{H}_m = -\sum_{\ell=1}^{J} \phi_{m,\ell} \log(\phi_{m,\ell} + \epsilon) \tag{25}$$

and select candidates in descending $\mathcal{H}_m$.

Core-set. Starting from $\mathcal{D}_0$, greedy k-center repeatedly selects

$$x_t^* = \arg\max_{x \in \mathcal{P}} \min_{x_j \in \mathcal{D}^{(t)}} \|\bar{\mathbf{h}}_x - \bar{\mathbf{h}}_{x_j}\|_2. \tag{26}$$

MOSAIC. We use the learned scene clusters as domains. For each cluster, we add 100 and 200 samples to the common seed set and fine-tune the planner for five epochs from the same seed checkpoint. Let n denote the number of added samples and $\Delta U_i(n)$ the observed performance gain over the seed model. We fit β_i and ζ_i to the two pilot observations using

$$\Delta U_i(n) = \beta_i\big(1 - \exp(-n/\zeta_i)\big). \tag{27}$$

Allocation then proceeds according to the predicted gain from adding one sample to each cluster.

Chameleon. We use the six K-means centers as domain embeddings and stack them as the rows of $\mathbf{E}_D$. With $K = 6$ clusters, we compute $\Omega_D = \mathbf{E}_D \mathbf{E}_D^\top$ and the kernel-ridge leverage score for cluster i as

$$\ell_{\mathrm{kr}}(D_i) = \big[\Omega_D(\Omega_D + K\lambda_{\mathrm{kr}} I)^{-1}\big]_{ii}, \qquad \lambda_{\mathrm{kr}} = 1. \tag{28}$$

We convert the reciprocal scores into mixture weights by applying a softmax to $\{1/\ell_{\mathrm{kr}}(D_i)\}_{i=1}^K$. The existing cluster centers serve as domain embeddings, without training an additional proxy representation.

E Generalization Experiments

We test whether the selection principle transfers beyond the main DiffusionDrive–NAVSIM setting. The cross-dataset experiment re-learns the scene structure and task-aware ranking signals on WaymoE2E, whereas the cross-backbone experiment retains the NAVSIM structure but recomputes every planner-dependent quantity with TransFuser. Both experiments use three independent training seeds.

E.1 WOD-E2E Adaptation

For WOD-E2E (Xu et al., 2026), the scene structure is re-learned rather than transferring the NAVSIM cluster centers. Since per-sample PDMS is not available in the same form, CASA ranking replaces planner mastery with a cluster-normalized combination of ADE and FDE, while trajectory quality is estimated using an ego-trajectory smoothness score based on jerk. All scaling-law parameters are re-fit on WaymoE2E. We evaluate the selected subsets on a fixed Rater Feedback Score (RFS) evaluation set at two representative training sizes, 787 and 1311 samples. As shown in Table 8, CASA achieves the best mean RFS score at both sizes. Its margin over the strongest baseline is 0.0942 at 787 samples and 0.0046 at 1311 samples. The smaller margin at the larger size suggests that the baselines become more competitive as coverage increases, while the adapted selector does not lose its advantage.

Table 8: WaymoE2E results on the fixed RFS evaluation set. Values are mean ± standard deviation over three training seeds. Best mean in bold.

Method	787	1311
Random	6.0999 ± 0.1834	6.4416 ± 0.0888
Core-set	5.9349 ± 0.1611	6.3692 ± 0.0537
Chameleon	5.8383 ± 0.2807	6.3140 ± 0.0658
CASA	$\mathbf{6.1941} \pm 0.1699$	$\mathbf{6.4462} \pm 0.0532$

E.2 TransFuser adaptation

For the cross-backbone test on NAVSIM, the structural clusters, dispersion statistics, and overlap matrix are reused because the dataset is unchanged. Planner-dependent CASA scores and cluster value priors are recomputed with a TransFuser (Chitta et al., 2022) seed model, and the five scaling-law parameters are re-fit using the same seven probe compositions. Table 9 reports results on NavTest excluding Quick2000 at 800 and 1300 samples. CASA attains the best mean PDMS at both sizes, exceeding the strongest baseline by 0.0141 and 0.0030, respectively. These results indicate that the allocation framework is not tied to DiffusionDrive, although the smaller 1300-sample margin warrants evaluation with additional backbones and seeds.

Table 9: Cross-backbone results with TransFuser on NavTest excluding Quick2000. Values are mean ± standard deviation over three training seeds. Best mean in bold.

Method	800	1300
Random	0.5605 ± 0.0172	0.5531 ± 0.0344
Core-set	0.5430 ± 0.0275	0.5872 ± 0.0082
Chameleon	0.5683 ± 0.0184	0.5772 ± 0.0189
CASA	$\mathbf{0.5824} \pm 0.0177$	$\mathbf{0.5902} \pm 0.0371$