

Scaling-Aware Data Selection for End-to-End Autonomous Driving Systems

Tolga Dimlioglu^{1†}, Nadine Chang², Maying Shen², Rafid Mahmood^{2,3}, Jose M. Alvarez²

¹New York University, ²NVIDIA, ³University of Ottawa

td2249@nyu.edu, {nadinec, mshen, rmahmood, josea}@nvidia.com

Abstract

Large-scale deep learning models for physical AI applications depend on diverse training data collection efforts. These models and correspondingly, the training data, must address different evaluation criteria necessary for the models to be deployable in real-world environments. Data selection policies can guide the development of the training set, but current frameworks do not account for the ambiguity in how data points affect different metrics. In this work, we propose *Mixture Optimization via Scaling-Aware Iterative Collection (MOSAIC)*, a general data selection framework that operates by: (i) partitioning the dataset into domains; (ii) fitting neural scaling laws from each data domain to the evaluation metrics; and (iii) optimizing a data mixture by iteratively adding data from domains that maximize the change in metrics. We apply MOSAIC to autonomous driving (AD), where an End-to-End (E2E) planner model is evaluated on the Extended Predictive Driver Model Score (EPDMS), an aggregate of driving rule compliance metrics. Here, MOSAIC outperforms a diverse set of baselines on EPDMS with up to 80% less data.

1. Introduction

Large-scale deep learning models are fueled by diverse data collection efforts [35, 41]. This practice is particularly prominent in physical artificial intelligence (AI) applications such as autonomous driving (AD), where video clips are collected over different locations, weather, and traffic conditions [12, 18]. It is computationally inefficient to train models on all collected data, which in physical AI can scale to hundreds of millions of hours of clips. This necessitates data mixture selection policies to construct and grow training sets of diverse and influential samples that maximize desired performance metrics.

Dataset selection and optimization has been broadly studied from various perspectives. For instance, influence (or duplicate) estimation techniques use feature information to select useful data samples [1, 5, 49], while active

[†]work done during internship at NVIDIA.

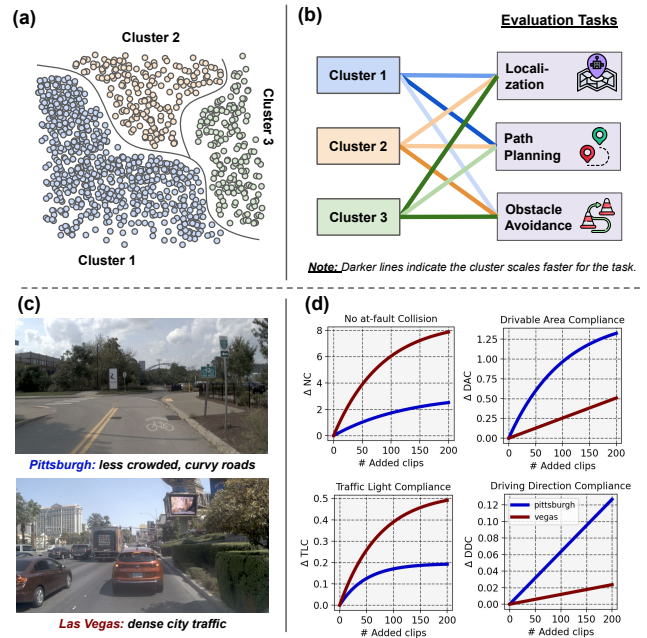


Figure 1. (a-b) The data pool is partitioned into a set of discrete domains which may each contribute to performance improvement of different evaluation tasks at varying rates. (c-d) Example application in autonomous driving: two clusters representing different driving contexts—Pittsburgh (curvy suburban roads) and Las Vegas (dense urban traffic). Data from separate contexts influence different rule-compliance metrics at distinct rates.

learning strategies optimize over this feature space [38, 46]. Large language models (LLMs) and their multi-modal extensions have successfully leveraged scaling laws to forecast how model performance improves with dataset size [21, 26, 58]; this premise has expanded to other applications including AD [3]. Further, as data collection becomes increasingly complex, scaling laws are used to determine optimal mixtures of data from explicit domains (e.g., different languages, math, coding) [23, 36, 56]. Although these methods present a general opportunity for physical AI systems, they are not immediately usable for applications that require both understanding and interacting with diverse real-world scenarios for three main reasons. First, physical AI systems are evaluated over a set of potentially compet-

ing metrics [14, 57]. Second, different data samples can influence different combinations of these metrics at various rates. Finally, the data pool is not necessarily immediately separable into subsets that have consistent, predictable influence on the metrics. Existing data mixture methods assume well-defined and homogeneous domains. However, they overlook the heterogeneous and metric-dependent improvement rates that arise when data sources influence different aspects of performance at varying rates [20, 54]. For example, a physical AI system such as an autonomous vehicle must progress along a route, follow driving rules, and avoid collisions [14]. High-traffic and pedestrian-heavy driving clips, when used for training, may impact certain metrics more than others. Moreover, finding such a subset of potential training data that has shared effects on the metrics requires careful selection and mining.

In this work, we develop a data selection and mixture optimization policy that addresses the present physical AI challenges of multiple competing metrics and imprecise data partitions. To address the challenge of imprecise data domains, we first partition a data pool into a set of separable clusters, within which we can rank samples on their influence to the metrics. We estimate the impact of each cluster on each metric, and correspondingly, an overall utility function that aggregates all the metrics. This impact is measured in terms of scaling laws that estimate the improvement to the metrics if more data from a specific cluster were used for training. Finally, we iteratively add new data to the training set by identifying the cluster which will maximize the expected gain to the aggregate utility with each additional data point. In this way, we optimize the mixture of data from our generated partitions. Figure 1 summarizes the challenges.

We apply our framework, Mixture Optimization via Scaling-Aware Iterative Collection (*MOSAIC*), for End-to-End (E2E) autonomous driving, where the challenges of data heterogeneity and metric competition are particularly pronounced. The goal is to optimize the Extended Predictive Driving Model Score (EPDMS), which aggregates a diverse set of rule-compliance metrics. *MOSAIC* is more data efficient than existing methods and achieves better EPDMS performance than naïve baselines with up to 82% less additional data. Our contributions are:

- We propose *MOSAIC*, a generic data mixture optimization pipeline that (i) clusters and ranks data, (ii) models domain-specific data scaling, and (iii) mines samples to maximize the expected gain over aggregate metrics.
- We apply *MOSAIC* to End-to-End Autonomous Driving (E2E AD) on the NAVSIM and OpenScene benchmarks using the challenge winning Hydra-MDP model [33], where it achieves substantially higher driving performance than existing data selection and mixture baselines, and improve data efficiency by up to 82%. Moreover, *MOSAIC* achieves the full training performance

while requiring 42% less data samples.

- We empirically demonstrate the necessity and robustness of our joint clustering and scaling procedure. First, *MOSAIC* outperforms baselines regardless of the clustering approach (e.g., semantic captions, geolocation). Second, embedding scaling laws on top of clustering significantly outperforms clustering-only strategies. This underscores the importance of our principled data selection strategy, which leverages the estimated improvement rates of different data clusters to maximize model performance under limited data budgets.

2. Related Works

Data Mixtures. Recent work has highlighted the importance of how data from different domains are combined for large-scale model training [17, 23, 36, 39, 53–56]. DoReMi [53] employs two proxy models to estimate domain weights based on excess loss, which are later used to reweigh domains when training a larger model. DOGE [17], tracks domain-specific gradients while training the proxy model to better capture inter-domain dynamics. Chameleon [54] instead leverages kernel similarity scores computed in the model’s latent space to assign adaptive weights to data from different sources. Another line of work treats data mixture optimization as a regression problem: many small proxy models are trained with varying mixtures, and a regressor is then fit to predict the optimal mixture at larger scale [36, 56]. A particularly relevant approach to ours is ADO [23], which begins with a random data mixture and fits scaling estimators on the fly during training. The gradients of these estimators are used for mixture reweighting. However, ADO does not model how performance scales with different data sources in isolation, and it requires a temporal averaging mechanism with multiple hyperparameters to maintain the precision of scaling fits. Although the aforementioned data mixture methods assign weights to samples from different domains, these weights can also be interpreted as sampling probabilities for constructing mixtures with varying domain ratios. In our experiments, we adopt Chameleon [54] as a baseline, since it has been shown to outperform other mixture algorithms.

Data Pruning & Selection. Data pruning aims to identify a compact subset of training data by removing redundant samples while preserving model performance [47]. In vision tasks, Abbas et al. [1] proposed removing visually similar samples using cosine similarity in the CLIP [43] feature space. Follow-up works extended this idea to specialized domains such as object detection [25] and fairness-aware multimodal learning [48]. It has also been shown analytically that optimal pruning strategies can improve power-law scaling behavior [49]. A closely related line of work, Active learning (AL), aims to maximize model performance improvement under a limited annotation budget [44, 59].

In this setting, the model has access to a large unlabeled data pool and, based on some selection signal the most informative samples are identified to be used for training. Early works focused on using the model’s prediction uncertainty, quantified through posterior probabilities [32], classifier margins [45], or entropy [24]. A notable method, CoreSet [46], seeks representation diversity by mining samples that maximize coverage in the latent space. Other data selection strategies quantify the sample importance using expensive signals such as influence on model updates [37], gradient-based criteria [8] or forgetting score [50].

End-to-End Autonomous Driving. This task aims to train planner models that map raw sensory inputs directly to control commands. Early approaches [4, 10, 42] learned control actions from RGB inputs via imitation learning, while later works incorporated richer input modalities such as LiDAR and navigational commands [9, 52]. Recently, conventional open-loop metrics have been shown to correlate poorly with closed-loop driving quality [13, 34]. This motivated the development of simulation benchmarks that better reflect real-world driving performance [14] and, AD models to employ probabilistic and rule-compliant trajectory planners [7, 22, 33].

3. MOSAIC

3.1. Main Problem

We want to train a Deep Neural Network (DNN) $f(\cdot; \mathcal{D})$ on a dataset $\mathcal{D}$ to perform a given task. We evaluate model performance using a set of R metrics $\mathcal{G}_r(f(\cdot; \mathcal{D}), \mathcal{D}_{val})$ for $r \in \{1, \dots, R\}$, where $\mathcal{D}_{val}$ is a held-out validation dataset. For brevity, we denote $\mathcal{G}_r(f(\cdot; \mathcal{D}), \mathcal{D}_{val})$ by $\mathcal{G}_r(\mathcal{D})$. To balance the trade-offs between the metrics, we use a utility function $U(\{\mathcal{G}_r(\mathcal{D})\}_{r=1}^R)$ that aggregates each metric into a final score. We do not assume about the structure of the utility function; for example, the simplest approach would be a summation $U(\cdot) = \sum_{r=1}^R \mathcal{G}_r(\mathcal{D})$.

We initialize with a current training dataset $\mathcal{D}_{train}$ and a data pool $\mathcal{D}_{pool}$. Given a budget B , our goal is to select a subset $\mathcal{D}_{sel} \subset \mathcal{D}_{pool}$ with $|\mathcal{D}_{sel}| = B$ that maximizes the improvement in model performance when f is retrained on the combined dataset $\mathcal{D}_{train} \cup \mathcal{D}_{sel}$. Formally, we write

$$\max_{\substack{\mathcal{D}_{sel} \subset \mathcal{D}_{pool} \\ |\mathcal{D}_{sel}| = B}} U\left(\{\mathcal{G}_r(\mathcal{D}_{train} \cup \mathcal{D}_{sel})\}_{r=1}^R\right) \quad (1)$$

To solve problem (1), we must determine how each data sample added in $\mathcal{D}_{sel}$ influences each of the metrics, while optimizing the trade-offs between these metrics to maximize $U(\cdot)$. For instance, in our AD application, $f(\cdot; \mathcal{D})$ is a planner model that maps sensory inputs to a predicted driving trajectory. Here, our goal is to identify driving clips that optimize the Extended Predictive Driving Model Score

Algorithm 1 Mixture Optimization via Scaling-Aware Iterative Selection (MOSAIC)

Require: Pool dataset $\mathcal{D}_{pool}$, number of clusters M , sample selection budget B .

Ensure: Selected dataset $\mathcal{D}_{sel}$

```

1:  $\{\mathcal{D}_{pool}^{i, ranked}\}_{i=1}^M = \text{ClusterAndRank}(\mathcal{D}_{pool}, M)$ 
2:  $\{\Delta \hat{U}_i(n)\}_{i=1}^M = \text{GetScalings}(\{\mathcal{D}_{pool}^{i, ranked}\}_{i=1}^M)$ 
3:  $\mathcal{D}_{sel} \leftarrow \{\}$ 
4:  $b_i \leftarrow 0$  for all  $i \in \{1, \dots, M\}$ 
5: while  $|\mathcal{D}_{sel}| < B$  do
6:   for  $i = 1$  to  $M$  do
7:      $\delta_i(b_i) \leftarrow \Delta \hat{U}_i(b_i + 1) - \Delta \hat{U}_i(b_i)$ 
8:   end for
9:    $j \leftarrow \arg \max_i \delta_i(b_i)$ 
10:   $\text{sample} \leftarrow \text{ReturnSample}(\mathcal{D}_{pool}^{j, ranked}, b_j)$ 
11:   $\mathcal{D}_{sel} \leftarrow \mathcal{D}_{sel} \cup \{\text{sample}\}$ 
12:   $b_j \leftarrow b_j + 1$ 
13: end while
14: return  $\mathcal{D}_{sel}$ 
```

(EPDMS) [6, 27], which is an aggregate of $R = 9$ closed-loop rule compliance scores: NC, DAC, DDC, TLC, EP, TTC, LK, HC, EC. Each driving clip can showcase only certain aspects of driving rule compliance, e.g. driving on a curvy road might improve lane keeping while degrading the comfort, however our goal is to elevate the model performance across all metrics. Consequently, solving this problem requires disentangling the relationships between the data samples and the metrics before optimizing the trade-offs between them.

3.2. Scaling-Aware Iterative Collection

We propose *Mixture Optimization via Scaling-Aware Iterative Collection (MOSAIC)*; a three-stage data selection framework: (i) first, cluster the pool $\mathcal{D}_{pool}$ into partitions that capture distinct driving scene contexts and rank the data samples based on an importance score within each cluster; (ii) estimate the scaling law of adding data from each cluster with respect to the utility U ; and (iii) iteratively mine samples from the clusters to optimize 1. Figure 2 visualizes our framework. Algorithm 1 summarizes the steps.

3.2.1. Clustering & Ranking the Data

Before solving problem (1), we first disentangle the relationships between the metrics and data samples by clustering the data pool into a set of structured domains [15, 47]. Our goal is to find subsets of the data pool that have similar influence, i.e., samples that all influence the same set of metrics. Given a feature representation, we cluster the data pool into M domains, i.e., $\mathcal{D}_{pool} = \bigcup_{i=1}^M \mathcal{D}_{pool}^i$. For example in AD, we may partition clips into clusters of highway driving, busy intersections, and calm local streets, that pri-

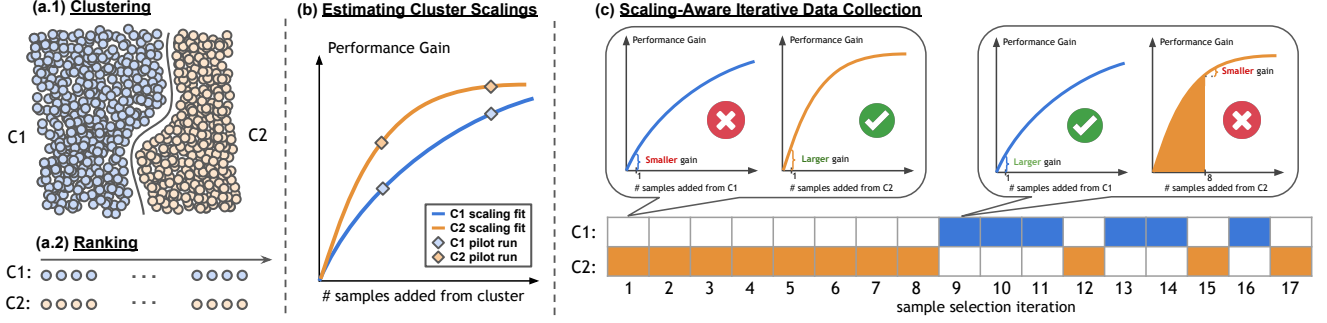


Figure 2. Overview of the proposed *MOSAIC* framework. (a) The pool $\mathcal{D}_{pool}$ is clustered and ranked by sample importance. (b) Cluster-wise scaling laws are fitted on pilot runs to estimate how performance scales with added data. (c) Samples are then iteratively mined from the cluster with the highest estimated marginal gain under the fitted scaling laws.

marily address ego progress, collision avoidance, and traffic light compliance, respectively.

Although, clustering separates the data into domains of similar influence, each domain will include samples that have stronger influence than others [28, 30]. When adding data, we should first exhaust the higher-influence samples [49]. As a result, we rank the samples x within each cluster via an importance score $\mathcal{I}(x)$. In our application, we define importance by evaluating the model on that sample $\mathcal{I}(x) := U(\{\mathcal{G}_r(f(\cdot; \mathcal{D}_{train}), x)\}_{r=1}^R)$. Later when adding data from each cluster, we first select samples with higher $\mathcal{I}(x)$ (Line 1 of Algorithm 1).

3.2.2. Selecting Data by Optimizing a Mixture

Given a set of discrete domains, problem (1) can be reformulated into a data mixture optimization problem. Let $\mathcal{D}_{sel}^i \subset \mathcal{D}_{pool}^i$ be the data added from the i -th domain and let $\mathcal{D}_{sel} = \cup_{i=1}^M \mathcal{D}_{sel}^i$. Furthermore, because each sample in $\mathcal{D}_{pool}^i$ is ranked via importance scores, it remains only to determine how many samples to draw from each domain.

Mathematically, we reformulate problem (1) to a proxy optimization problem below

$$\max_{n_1, \dots, n_M, \sum_{i=1}^M n_i = B} \Delta U_{mix}(n_1, \dots, n_M) \quad (2)$$

where $n_i := |\mathcal{D}_{sel}^i|$ is the number of samples drawn from the i -th domain and $\Delta U_{mix}(n_1, \dots, n_M)$ is

$$U\left(\{\mathcal{G}_r(\mathcal{D}_{train} \cup \{\cup_{i=1}^M \mathcal{D}_{sel}^i\})\}_{r=1}^R\right) - U(\{\mathcal{G}_r(\mathcal{D}_{train})\}_{r=1}^R)$$

the change in utility after adding n_i points from each domain. Note that the objective above is equivalent to optimizing $U(\{\mathcal{G}_r(\mathcal{D}_{train} \cup \{\cup_{i=1}^M \mathcal{D}_{sel}^i\})\}_{r=1}^R)$, but we use the above formulation since it explicitly expresses the problem in terms of performance gains from additional data.

Solving problem (2) requires quantifying how adding data samples from each domain will improve $U(\cdot)$. We estimate this by approximating ΔU_{mix} into separate effect

from each cluster and then estimating the effect of data from each domain via a scaling law. First, we apply the following linear separable approximation

$$\Delta U_{mix}(n_1, \dots, n_M) \approx \sum_{i=1}^M \Delta U_i(n_i). \quad (3)$$

where each $\Delta U_i(n)$ is the improvement in utility when adding only the data from the i -th domain

$$U\left(\{\mathcal{G}_r(\mathcal{D}_{train} \cup \mathcal{D}_{sel}^i)\}_{r=1}^R\right) - U\left(\{\mathcal{G}_r(\mathcal{D}_{train})\}_{r=1}^R\right)$$

Intuitively, the approximation in (3) assumes that each domain has an independent effect on the overall ΔU_{mix} . Moreover, this assumption allows us to estimate how model performance scales if we add data from each domain independently. We use a saturating exponential scaling law

$$\Delta U_i(n) \approx \widehat{\Delta U}_i(n) := a_i(1 - e^{-n/\tau_i}) \quad (4)$$

where a_i and τ_i are learnable parameters of the scaling law estimated from small-scale pilot-runs (we provide details in Section 9 of the Appendix.), and n denotes the number of added samples [55]. Here, a_i represents the asymptotic improvement on the total utility U when sampling from domain i , while τ_i governs the saturation rate, i.e., how quickly the marginal benefit of adding data from the domain decreases. Obtaining the scaling estimators is symbolically captured in line 2 of Algorithm 1. Then, substituting (4) into problem (2) yields a concave maximization problem that we can compute and solve.

3.2.3. Scaling-Aware Iterative Data Collection by First-Difference Steps

We propose an efficient algorithm to solve problem (2) by allocating data samples one-by-one from the domain that stands to give the highest marginal improvement to U_{mix} at any given time. Intuitively, this iterative adding of data mimics a gradient-based approach of taking small steps to optimize the data mixture.

Suppose that we have so far added b_i data points from each cluster in order to generate $\mathcal{D}_{sel}$. Then, let

$$\delta_i(b_i) := \widehat{\Delta U}_i(b_i + 1) - \widehat{\Delta U}_i(b_i), \quad (5)$$

be the marginal improvement in $\widehat{\Delta U}_i(b_i)$ if we draw one additional data point from the i -th domain. Mathematically, $\delta_i(b_i)$ is an approximate first-difference analogue of the partial derivative of U_{mix} . Furthermore, because $\widehat{\Delta U}_i(n)$ is a concave function of n , this difference decreases as b_i increases. This means that at a certain point, each domain yields diminishing value to the training dataset and we should draw from other domain.

In our algorithm, we iteratively add data from the domain with the highest marginal improvement. In each iteration, if we have so far drawn b_i samples from the i -th domain, we first identify $j = \arg \max_i \delta_i(b_i)$. We then sample a data point from $\mathcal{D}_{pool}^i$ according to the importance scores $\mathcal{I}(x)$. We then update our counts b_i , and repeat the process until we have reached the budget (lines 5-14 in Algorithm 1).

4. Experiments

We empirically evaluate *MOSAIC* on two different datasets (OpenScene and Navtrain) using a challenge-winning model Hydra-MDP, and report consistent gains in the model performance at all budgets while being up to 80% data efficiency than the baselines.

4.1. Protocols

We provide more details in Section 7 of Appendix.

Datasets. We use two train-pool configurations: the curated Navtrain [14] split and the full trainval split of OpenScene [11]. For clarity, we refer to the latter as the OpenScene experiment. In both settings, evaluation is conducted on the curated validation split *navtest* [14]. Both datasets contain driving session clips lasting from 30 seconds to 50 minutes, which has significant temporal variation over a limited number of sessions. Consequently, we segment each session into fixed-length 10-second *virtual clips* (20 frames at 2 Hz) and by doing so, we align our data handling practice with the industry standards [16]. In the experiments, each virtual clip is treated as a single sample.

For Navtrain, we use the dataset as both $\mathcal{D}_{train}$ and $\mathcal{D}_{pool}$, comprising 4,601 virtual clips. We randomly select 460 clips for $\mathcal{D}_{train}$, with the remaining 4,141 clips forming $\mathcal{D}_{pool}$. We evaluate all methods under budgets $B \in \{100, 200, 400, 800, 1600, 2400\}$. For OpenScene, we randomly select 1,000 clips as the $\mathcal{D}_{train}$ and reserving the remaining 31,539 as $\mathcal{D}_{pool}$. The sample selection budgets of this setting are $B \in \{250, 500, 1000, 2000, 4000, 8000\}$.

Model. We use the Hydra-MDP model [33], the winner of the NAVSIM Challenge in 2024 [14], with a pretrained

VoVNetV2-99 backbone [31, 40]. The trajectory vocabulary size is set to 16,384. For OpenScene experiments, rule-based distillation is disabled due to the substantial pre-processing time required to compute compliance scores.

Baselines. We compare *MOSAIC* against several baseline data selection strategies:

- *Random*: selects clips uniformly from the pool dataset under the given selection budget.
- *Uncertainty* [24]: measured via the entropy of the trajectory logits. Samples with higher entropy are prioritized.
- *Coreset* [46]: selects samples from the pool that maximize diversity over the feature space.
- *Chameleon* [54]: a data mixture framework that uses kernel ridge scores using domain embeddings in the model’s feature space to assign mixture weights to each domain.

Pseudo-codes are in Section 7 of Appendix. For Chameleon and *MOSAIC*, we cluster $\mathcal{D}_{pool}$ into domains defined by the map metadata (i.e., Boston, Pittsburgh, Singapore, Vegas). Each experiment is repeated with three random seeds. For OpenScene experiments with more than 1,000 clips, we use two seeds to reduce computational cost. Reported results are averaged over runs, and the standard deviation is shown as a subscript in the tables.

Metrics. We evaluate models using the EPDMS, an aggregate of nine rule-compliance metrics that has been shown to correlate strongly with closed-loop driving performance [6, 14]. Consequently, EPDMS has become the standard evaluation metric for AD planners, replacing conventional open-loop measures such as ADE and FDE. Formally, EPDMS is computed as:

$$\text{EPDMS} := \prod_{m \in \mathcal{M}_{\text{pen}}} m \cdot \frac{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m m}{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m}$$

where $\mathcal{M}_{\text{pen}} := \{\text{NC}, \text{DAC}, \text{DDC}, \text{TLC}\}$ denotes the set of penalty terms, and $\mathcal{M}_{\text{avg}} := \{\text{EP}, \text{TTC}, \text{LK}, \text{HC}, \text{EC}\}$ denotes the metrics combined via a weighted average, with weights $\{5, 5, 2, 2, 2\}$ respectively [6]. The glossary of the rule compliance metrics are in Section 7 of Appendix.

Similar to the relevant works [51], we also measure how each data selection policy improves EPDMS relative to the *Random* baseline to assess sample efficiency. Specifically, we report the Budget Ratio to Match Random (BRMR); the ratio of the data budget B required by each method to achieve the same EPDMS performance attained by random selection at the same budget. Formally, let B_k denote the number of samples required by selection strategy k to match the EPDMS obtained by random sampling with budget B . Then $\text{BRMR} := B_k/B$. Lower BRMR indicates greater sample efficiency, as it reflects fewer samples needed to achieve the same performance level as random selection.

Table 1. Validation EPDMS (higher is better) and BRMR (lower is better) reported in OpenScene (Section A) and Navtrain (Section B) settings. We report the results for all budgets in Section 8 of Appendix.

Budget	Method	A. Openscene		B. Navtrain	
		EPDMS ($\uparrow$)	BRMR ($\downarrow$)	EPDMS ($\uparrow$)	BRMR ($\downarrow$)
A. 250 B. 100	Random	72.84 $\pm$ 1.14	1.00	84.66 $\pm$ 0.60	1.00
	Uncertainty	70.78 $\pm$ 0.59	14.58	84.50 $\pm$ 0.48	1.47
	Coreset	76.26 $\pm$ 0.48	0.20	85.29 $\pm$ 0.47	0.53
	Chameleon	72.97 $\pm$ 1.72	0.86	84.57 $\pm$ 0.18	1.07
	MOSAIC	77.38$\pm$1.58	0.15	86.29$\pm$0.43	0.30
A. 1000 B. 400	Random	75.84 $\pm$ 0.90	1.00	86.69 $\pm$ 0.20	1.00
	Uncertainty	71.12 $\pm$ 0.38	8.00	86.07 $\pm$ 0.75	2.00
	Coreset	80.46 $\pm$ 0.02	0.22	87.09 $\pm$ 0.29	0.79
	Chameleon	79.08 $\pm$ 0.74	0.49	87.04 $\pm$ 0.60	0.82
	MOSAIC	81.68$\pm$0.52	0.18	88.21$\pm$0.03	0.38
A. 4000 B. 1600	Random	80.38 $\pm$ 0.55	1.00	88.62 $\pm$ 0.22	1.00
	Uncertainty	73.46 $\pm$ 0.19	2.00	87.75 $\pm$ 0.37	1.36
	Coreset	83.63 $\pm$ 0.36	0.25	89.30 $\pm$ 0.19	0.58
	Chameleon	82.92 $\pm$ 0.13	0.39	89.50 $\pm$ 0.20	0.62
	MOSAIC	84.25$\pm$0.14	0.18	90.18$\pm$0.25	0.37

4.2. Main Results: Openscene

Table 1 (Section A) reports the EPDMS and BRMR scores for different data selection methods. Across all clip budgets $B \in \{250, 1000, 4000\}$, *MOSAIC* consistently achieves the highest EPDMS, approximately one point higher than the next best method. This demonstrates the superior utility gains of *MOSAIC* under limited data. Moreover, *MOSAIC* requires over **80%** fewer samples to match the performance achieved by random selection (i.e., $\text{BRMR} < 0.2$).

We break down EPDMS into the individual nine metrics in Table 2. Section A corresponds to the Openscene experiments. The base model before data collection is particularly limited in DAC and EC, which impact EPDMS. *MOSAIC* achieves the largest gain in DAC, nearly 10 points higher than the base, and improves EC and EP, while maintaining balanced performance gains across other metrics. In contrast, the other methods yield less gains over the base DAC, and instead improves TTC and EC, that have less effect on the final EPDMS. On the other hand, *MOSAIC* achieves consistently Top-2 performance across all rule-compliance metrics, while strategically prioritizing DAC, the metric with the greatest room for improvement. This underscores the importance of incorporating scaling-aware collection into the data selection strategy to optimize U more effectively and achieve better trade-offs across competing metrics.

4.3. Main Results: Navtrain

Compared to Openscene, Navtrain is a curated dataset emphasizing non-trivial driving scenarios such as dense traffic and complex maneuvers. Table 1 (Section B) summarizes the EPDMS and BRMR results. Here, *MOSAIC* consistently delivers the strongest performance, achieving up to 1.1 points higher EPDMS than the next best method

across all budgets. It also attains the lowest BRMR values (< 0.4), corresponding to a 60–70% reduction in the number of samples needed to match the performance of *Random*. We conclude that *MOSAIC* remains highly effective even on the more challenging, curated Navtrain split, where each clip already carries substantial learning value.

The section B of Table 2 reports the breakdown of EPDMS. *MOSAIC* achieves consistent improvements across all metrics, with the largest gains observed in DAC, NC, and LK. Importantly, *MOSAIC* understands the trade-off between the metrics, and shifts the collection effort from saturated, less impactful metrics toward those that require more improvement. Overall, *MOSAIC* provides a more balanced and sustained improvement profile, suggesting that the scaling-aware allocation identifies data with broader generalization benefits. Ultimately, this allows *MOSAIC* to achieve a higher EPDMS than the baselines under the same clip budget.

4.4. Ablating the effectiveness of MOSAIC

Dynamics of scaling-aware data selection. In the Openscene experiments, $\mathcal{D}_{pool}$ is partitioned based on geolocation into four domains corresponding to Las Vegas, Boston, Singapore, and Pittsburgh. Figure 3 illustrates the fitted scaling curves for each city, where $\star$ markers denote the pilot-run results used to estimate the parameters of the scaling curves in Equation 4. We note that different domains scale at different rates depending on the how many clips are added. Specifically, data collected from Boston and Singapore yield the largest initial performance gains in the low-data regime (< 500 clips), while Pittsburgh maintains steadier improvements and eventually supercedes all other domains at high data budgets. In contrast, the Las Vegas cluster provides the smallest gains and saturates early. These heterogeneous scaling behaviors are later exploited by the scaling-aware selection policy of *MOSAIC* to maximize the performance gain, under varying data budgets.

Figure 4 shows how these fitted scaling laws influence the order in which samples are added to the training set. The y-axis lists data clusters, i.e. the city names, and the x-axis denotes the iteration index. Each bar indicates from which cluster the next sample is collected from at any iteration. During the early stages, only Boston and Singapore are actively mined, while Las Vegas and Pittsburgh are largely ignored. Figure 3 (top right) confirms this behavior: when the budget is 250, most selected clips originate from Boston and Singapore. As the returns from Boston and Singapore diminish, Pittsburgh’s steadier scaling curve makes it increasingly favorable between indices 500 to 3700. Figure 3 (bottom right) shows that at 4000 clips, the selected set is dominated by Pittsburgh samples. After around 3700 collection rounds, the Pittsburgh data domain is exhausted. Beyond approximately 2500 sample selections, the scaling

Table 2. Breakdown of the nine EPDMS rule-compliance metrics for the base model and the models trained with data selected by various strategies at a single budget, shown for both the OpenScene and Navtrain experiments.

Setting		NC ($\uparrow$)	DAC ($\uparrow$)	DDC ($\uparrow$)	TLC ($\uparrow$)	EP ($\uparrow$)	TTC ($\uparrow$)	LK ($\uparrow$)	HC ($\uparrow$)	EC ($\uparrow$)	EPDMS ($\uparrow$)
A. Openscene	Base	94.05	83.9	96.28	99.6	85.96	92.95	93.26	98.25	81.88	72.0
Budget 4000 Clips	Random	96.32 $\pm$ 0.59	90.53 $\pm$ 0.06	99.06 $\pm$ 0.07	99.79 $\pm$ 0.05	86.36 $\pm$ 0.48	95.66 $\pm$ 0.52	95.68 $\pm$ 0.09	98.30 $\pm$ 0.01	84.46 $\pm$ 0.14	80.38 $\pm$ 0.55
	Uncertainty	94.67 $\pm$ 0.28	85.11 $\pm$ 0.51	97.15 $\pm$ 0.54	99.71 $\pm$ 0.04	84.26 $\pm$ 0.69	93.72 $\pm$ 0.40	93.26 $\pm$ 0.09	98.28 $\pm$ 0.02	81.34 $\pm$ 1.06	73.46 $\pm$ 0.19
	Coreset	97.11$\pm$0.18	92.93 $\pm$ 0.60	99.44 $\pm$ 0.06	99.82$\pm$0.02	86.65 $\pm$ 0.55	96.42$\pm$0.19	96.66$\pm$0.30	98.16 $\pm$ 0.12	85.10 $\pm$ 0.06	83.63 $\pm$ 0.36
	Chameleon	96.76 $\pm$ 0.24	92.32 $\pm$ 0.02	99.51 $\pm$ 0.01	99.77 $\pm$ 0.01	86.98 $\pm$ 0.17	95.91 $\pm$ 0.31	96.49 $\pm$ 0.12	98.32$\pm$0.01	85.51$\pm$0.11	82.92 $\pm$ 0.13
	MOSAIC	96.97 $\pm$ 0.32	93.59$\pm$0.11	99.59$\pm$0.04	99.80 $\pm$ 0.01	87.14$\pm$0.98	96.18 $\pm$ 0.45	96.62 $\pm$ 0.08	98.28 $\pm$ 0.01	85.06 $\pm$ 0.34	84.25$\pm$0.14
B. Navtrain	Base	95.3	95.94	99.09	99.6	88.09	94.55	94.49	98.25	82.39	83.97
Budget 1600 Clips	Random	97.17 $\pm$ 0.07	98.19 $\pm$ 0.43	99.42 $\pm$ 0.05	99.69 $\pm$ 0.02	89.36 $\pm$ 0.12	96.50 $\pm$ 0.14	96.45 $\pm$ 0.25	98.31$\pm$0.03	83.17 $\pm$ 0.76	88.62 $\pm$ 0.22
	Uncertainty	96.92 $\pm$ 0.38	97.66 $\pm$ 0.08	99.22 $\pm$ 0.10	99.77$\pm$0.02	89.02 $\pm$ 0.28	96.24 $\pm$ 0.40	96.10 $\pm$ 0.07	98.30 $\pm$ 0.01	82.92 $\pm$ 0.38	87.75 $\pm$ 0.37
	Coreset	97.50 $\pm$ 0.10	98.31 $\pm$ 0.34	99.59 $\pm$ 0.03	99.72 $\pm$ 0.05	89.27 $\pm$ 0.21	96.86 $\pm$ 0.07	96.75 $\pm$ 0.22	98.30 $\pm$ 0.03	83.88$\pm$0.50	89.30 $\pm$ 0.19
	Chameleon	97.43 $\pm$ 0.22	98.46 $\pm$ 0.17	99.60 $\pm$ 0.05	99.75 $\pm$ 0.03	89.60$\pm$0.19	96.83 $\pm$ 0.30	96.89 $\pm$ 0.07	98.30 $\pm$ 0.03	83.87 $\pm$ 0.34	89.50 $\pm$ 0.20
	MOSAIC	98.04$\pm$0.24	98.61$\pm$0.32	99.63$\pm$0.06	99.73 $\pm$ 0.02	89.28 $\pm$ 0.19	97.50$\pm$0.32	97.07$\pm$0.06	98.28 $\pm$ 0.04	83.70 $\pm$ 0.41	90.18$\pm$0.25

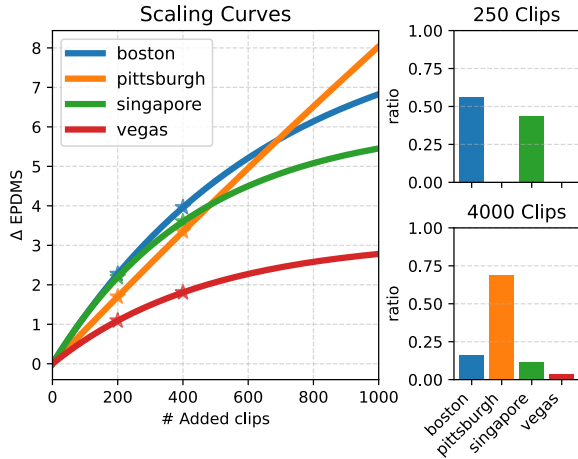


Figure 3. (Left) Performance scalings of different clusters, obtained by fitting the estimator in Equation 4 on 2 pilot runs, denoted by $\star$. (Right) Geolocation distributions at different budgets as a result of scaling-aware iterative selection.

curves of Boston, Singapore, and Pittsburgh approach saturation, causing their marginal gains to diminish. As a result, the expected improvement from the initial Las Vegas samples becomes comparable to those of the other regions, leading *MOSAIC* to mine from Vegas.

MOSAIC is optimal under different clustering mechanisms. Instead of partitioning the data pool into domains separated by geolocation, we cluster the Navtrain data pool using captions generated for each clip. We use the Qwen-2.5-VL-32B-Instruct model [2] to generate captions for all clips in $\mathcal{D}_{pool}$. We form six clusters that capture distinct driving and scene contexts using the TF-IDF feature vectors of the generated captions. The top uni-grams and bi-grams characterizing each cluster are provided in Table 3.

Figure 5 reports the validation EPDMS as a function of the data selection budget for all strategies. We also indicate the base performance and the full-training performance, corresponding to the model trained by including all

Table 3. Top uni-grams and bi-grams of different clusters.

Uni-grams & Bi-grams	
Cluster 1	calm, day, street, trees, signs, yellow
Cluster 2	signals, crossing, crosswalks, pedestrians
Cluster 3	highway, vehicles, busy urban, palm trees
Cluster 4	building, area, large, paved, parking
Cluster 5	city street, major city, moderate
Cluster 6	precipitation, potential rain, overcast, cloudy

4,141 clips in $\mathcal{D}_{pool}$ for training. (We provide full table with subscores in Section 8 of Appendix) *MOSAIC* consistently outperforms all baselines, including *Chameleon* (i.e., the other data mixture optimization method); requires 61% and 52% fewer samples than random selection to match its performance at the highest budgets of 1,600 and 2,400 clips, respectively. Moreover, *MOSAIC* reaches the full performance of training with all data samples using only 2,400 clips, i.e., 42% fewer samples. Interestingly, *Chameleon* degrades under caption-based clustering, despite being the strongest baseline in the previous setting. This indicates that its kernel ridge weighting is highly sensitive to the structure of the clustered domains. Also, since the clustering choice only affects *Chameleon* and *MOSAIC*, the other strategies have the same performance as before.

Combining clustering and ranking yields the best data selection policy. We ablate the effects of both clustering and ranking components by individually disabling them. First, we use a “w/o Clustering” variant where we simply rank $\mathcal{D}_{pool}$ by the EPDMS importance scores $\mathcal{I}(x)$ and greedily add samples with the lowest scores until reaching the collection budget. Second, we use a “w/o Ranking” variant where we disable the ranking step, while retaining clustering and the scaling-aware estimation of how many samples to collect. Here, we simply sample data points from each domain randomly to satisfy the budget. Moreover, the scaling laws are also estimated on unranked domains.

Figure 6 visualizes these baselines to show that perfor-

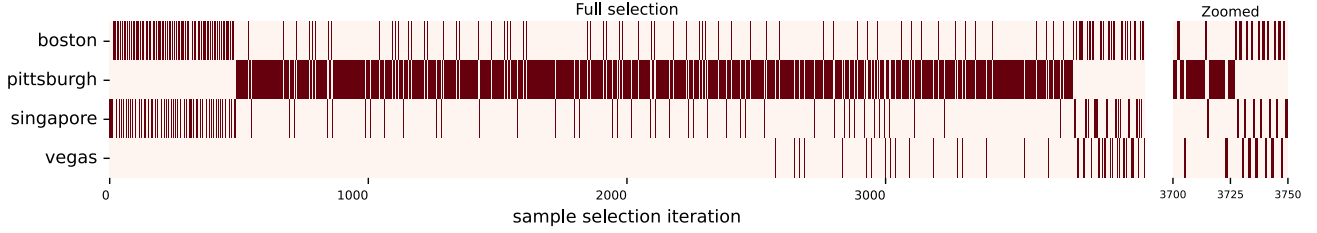


Figure 4. Visualization of the scaling-aware iterative data selection process. The x-axis denotes the sample selection iterations, and the y-axis lists the cluster names. Each vertical bar indicates from which cluster the next sample is mined at a given iteration, based on the estimated cluster-wise scaling fits. The left panel shows the complete selection process up to 4,000 clips, and the right panel zooms into iterations 3,700–3,750 for clarity.

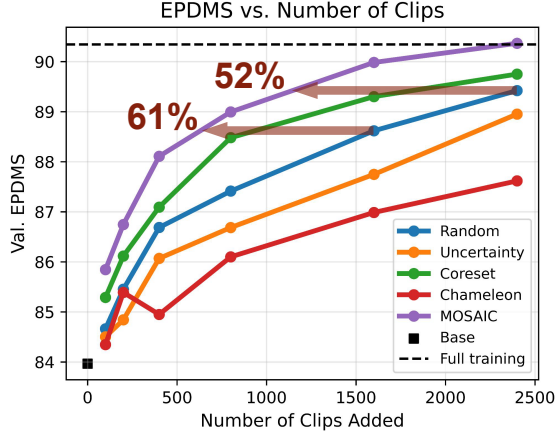


Figure 5. Validation EPDMS for various budgets obtained by different strategies when Navtrain data pool is clustered using clip captions. *MOSAIC* requires 61% and 52% fewer clips than *Random* selection to match its performance at the 1,600- and 2,400-clip budgets, respectively.

mance improvements in the low-data regime (up to a collection budget of 800 clips) can largely be attributed to ranking. The *MOSAIC* and *w/o Clustering* variants achieve competitive EPDMS in this region. However, in the higher data regime, merely adding clips with low EPDMS scores becomes less effective, as the performance of *w/o Clustering* begins to lag behind *MOSAIC*. Finally, we note that both of these disabled variants still outperform random collection by a large margin.

5. Limitations

We note two key limitations of *MOSAIC*. First, the relaxation from Equation 2 to Equation 3 assumes that each cluster’s contribution is well captured by its own scaling curve $\Delta U_i(n)$, with limited cross-cluster interactions. Consequently, if the clustering fails to produce well-separated groups, this assumption may be violated, leading *MOSAIC* to suboptimal allocation.

Second, *MOSAIC* relies on pilot runs to estimate cluster-specific scaling curves, which introduces additional computational cost. However, as shown in Section 9 of the Appendix, accurate scaling fits can be obtained efficiently us-

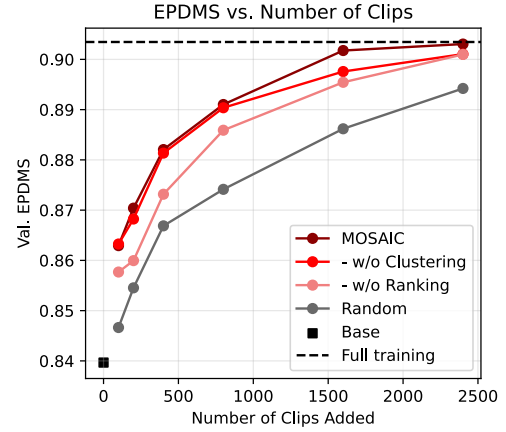


Figure 6. Analyzing the contribution of different components of the *MOSAIC* framework.

ing small pilot subsets or through continual training. Thus, despite the initial overhead for the pilot runs to obtain cluster scalings, *MOSAIC* ultimately requires less total compute to achieve superior performance in the large-data regime.

6. Conclusion

We introduce *MOSAIC*, a scaling-aware data selection framework that jointly leverages clustering, ranking, and scaling-law modeling to maximize the performance of a model defined by multiple competing metrics, under a limited data budget. We apply *MOSAIC* to E2E AD, where a planner model uses a diverse data pool to optimize a utility function that aggregates competing rule compliance metrics. Empirically, *MOSAIC* consistently outperforms existing data selection and mixture baselines on both the Openscene and Navtrain datasets by achieving substantial gains in EPDMS and sample efficiency. Ablation studies further highlight the framework’s mechanisms, analyze the necessity of the individual components components, and demonstrate robustness to clustering choices as long as semantic consistency is maintained. Overall, *MOSAIC* offers a general and principled blueprint for identifying influential data in large-scale, heterogeneous learning systems.

References

- [1] Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*, 2023. 1, 2
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhao-hai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 7
- [3] Mustafa Baniodeh, Kratarth Goel, Scott Ettinger, Carlos Fuertes, Ari Seff, Tim Shen, Cole Gulino, Chenjie Yang, Ghassen Jerfel, Dokook Choe, et al. Scaling laws of motion forecasting and planning—a technical report. *arXiv preprint arXiv:2506.08228*, 2025. 1
- [4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zeiba. End to end learning for self-driving cars, 2016. Available at <https://arxiv.org/abs/1604.07316>. 3
- [5] Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997. 1
- [6] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning (CoRL)*, 2025. 3, 5
- [7] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024. 3
- [8] Anshuman Chhabra, Peizhao Li, Prasant Mohapatra, and Hongfu Liu. "what data benefits my classifier?" enhancing model performance and interpretability through influence-based data selection. In *The Twelfth International Conference on Learning Representations*, 2024. 3
- [9] 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. 3
- [10] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4693–4700. IEEE, 2018. 3
- [11] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Vancouver, Canada*, pages 18–22, 2023. 5, 1
- [12] Michael J. Coren. Tesla has 780 million miles of driving data, and adds another million every 10 hours, 2025. Accessed: YYYY-MM-DD. 1
- [13] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conf. on Robot Learning*, 2023. 3
- [14] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024. 2, 3, 5, 1
- [15] Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, et al. Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *arXiv preprint arXiv:2504.13161*, 2025. 3
- [16] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *IEEE Int. Conf. on Computer Vision*, 2021. 5, 1
- [17] Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation. *arXiv preprint arXiv:2310.15393*, 2023. 2
- [18] Adam Grzywaczewski. Training ai for self-driving vehicles: the challenge of scale, 2017. Accessed: YYYY-MM-DD. 1
- [19] K. Tan et al. H. Caesar, J. Kabzan. NuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *CVPR ADP3 workshop*, 2021. 1
- [20] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. *arXiv preprint arXiv:2405.18392*, 2024. 2
- [21] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. 1
- [22] 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 (ICCV)*, pages 8340–8350, 2023. 3
- [23] Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and J Zico Kolter. Adaptive data optimization: Dynamic sample selection with scaling laws. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2
- [24] 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*, pages 2372–2379. IEEE, 2009. 3, 5, 1
- [25] Feiyang Kang, Nadine Chang, Maying Shen, Marc T Law, Rafid Mahmood, Ruoxi Jia, and Jose M Alvarez. Adadedup: Adaptive hybrid data pruning for efficient large-scale object detection training. *arXiv preprint arXiv:2507.00049*, 2025. 2
- [26] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1
- [27] Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 629–636. IEEE, 2024. 3
- [28] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018. 4
- [29] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *Int. Conf. on Learning Representations*, 2015. 1
- [30] Agata Lapedriza, Hamed Pirsiavash, Zoya Bylinskii, and Antonio Torralba. Are all training examples equally valuable? *arXiv preprint arXiv:1311.6510*, 2013. 4
- [31] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 5, 1
- [32] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994. 3
- [33] 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*, 2024. 2, 3, 5, 1
- [34] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahua 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*, pages 14864–14873, 2024. 3
- [35] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C Knoll. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. *IEEE Transactions on Intelligent Vehicles*, 2024. 1
- [36] Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2
- [37] Zhuoming Liu, Hao Ding, Huaping Zhong, Weijia Li, Jifeng Dai, and Conghui He. Influence selection for active learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9274–9283, 2021. 3
- [38] Rafid Mahmood, Sanja Fidler, and Marc T Law. Low-budget active learning via wasserstein distance: An integer programming approach. In *International Conference on Learning Representations*, 2022. 1
- [39] 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. 2
- [40] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3142–3152, 2021. 5, 1
- [41] Priyaranjan Pattnayak, Hitesh Laxmichand Patel, Bhargava Kumar, Amit Agarwal, Ishan Banerjee, Srikant Panda, and Tejaswini Kumar. Survey of large multimodal model datasets, application categories and taxonomy. *arXiv preprint arXiv:2412.17759*, 2024. 1
- [42] Dean Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pages 305–313. Morgan Kaufmann, 1988. 3
- [43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2
- [44] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021. 2
- [45] Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European conference on machine learning*, pages 413–424. Springer, 2006. 3
- [46] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 1, 3, 5, 2
- [47] 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*, pages 2525–2535, 2025. 2, 3
- [48] Eric Slyman, Stefan Lee, Scott Cohen, and Kushal Kafle. Fairdedup: Detecting and mitigating vision-language fairness disparities in semantic dataset deduplication. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13905–13916, 2024. 2
- [49] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022. 1, 2, 4
- [50] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J.

- Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. [3](#)
- [51] Ziting Wen, Oscar Pizarro, and Stefan B. Williams. Feature alignment: Rethinking efficient active learning via proxy in the context of pre-trained models. *Transactions on Machine Learning Research*, 2024. [5](#)
- [52] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022. [3](#)
- [53] 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. [2](#)
- [54] Wanyun Xie, Francesco Tonin, and Volkan Cevher. Chameleon: A flexible data-mixing framework for language model pretraining and finetuning. In *Forty-second International Conference on Machine Learning*, 2025. [2](#), [5](#)
- [55] Chengyin Xu, Kaiyuan Chen, Xiao Li, Ke Shen, and Chenggang Li. Unveiling downstream performance scaling of llms: A clustering-based perspective. *arXiv preprint arXiv:2502.17262*, 2025. [4](#)
- [56] Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations*, 2025. [1](#), [2](#)
- [57] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. [2](#)
- [58] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022. [1](#)
- [59] Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B Chan. A comparative survey of deep active learning. *arXiv preprint arXiv:2203.13450*, 2022. [2](#)