WHICH SYNTHETIC DRIVING DATA ARE WORTH TRAINING ON? DYNAMIC DATA VALUATION BASED ON THEIR EFFECTS ON MODEL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models can now produce synthetic driving data at scale, providing a promising way to improve the performance of end-to-end driving models beyond costly real-world data collection. However, not all synthetic data are equally valuable for training, making it important to accurately identify which data are worth training on. Existing methods typically value synthetic data using their properties, such as visual realism and temporal consistency, thereby treating data value as an invariant property of the data. This overlooks two key issues: (i) these data properties do not directly reveal whether synthetic data actually improve model performance; and (ii) the value of synthetic data is not fixed, but varies across training stages and can be sensitive to the data used for valuation. To address these issues, we propose DriveVal, which evaluates synthetic driving data based on their training effects on end-to-end driving model performance and uses the resulting values for data selection. With a small set of real driving data as references, data value is quantified by the dot product between the real-data gradient and the synthetic-data gradient under evaluation. In addition, DriveVal periodically re-evaluates data throughout training while accounting for the current optimizer state at each stage, allowing the valuation to adapt as training progresses. Finally, DriveVal penalizes variations in data value across different real references and synthetic-data combinations, preventing the valuation from being dominated by a particular data choice. Across five end-to-end driving models on NAVSIM, DriveVal selects synthetic training data matched in size to navtrain throughout training. The resulting models achieve 95.2% of the mean PDMS of their corresponding real-data baselines, while improving mean PDMS by 13.6% over size-matched random selection and by 9.1% over training on the full synthetic set. Further experiments demonstrate that DriveVal remains effective across different generation sources and visual conditions.

1 INTRODUCTION

Recent advances in generative models have enabled synthetic driving data to be produced at scale, offering a promising complement to costly real-world data collection for training end-to-end driving models (Gao et al., 2024; Wang et al., 2024; Yang et al., 2025; Tian et al., 2026). However, synthetic data do not contribute equally to improving model performance in real driving scenes. Indiscriminately incorporating all synthetic data into training does not guarantee corresponding performance gains. As synthetic data continue to scale, the key question is therefore not simply how to generate more data, but which synthetic driving data are truly worth training on.

Existing methods typically evaluate synthetic data using properties such as visual realism, temporal consistency, controllability, and diversity (Gao et al., 2024; Wang et al., 2024; Yang et al., 2025; Ge et al., 2025; Tian et al., 2026; Kim et al., 2025). These properties describe different aspects of synthetic data, such as cross-frame coherence, whether the data follow specified conditions, and the driving scenes they cover. However, existing methods still evaluate synthetic data based on properties of the data themselves and typically treat data value as invariant throughout training.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

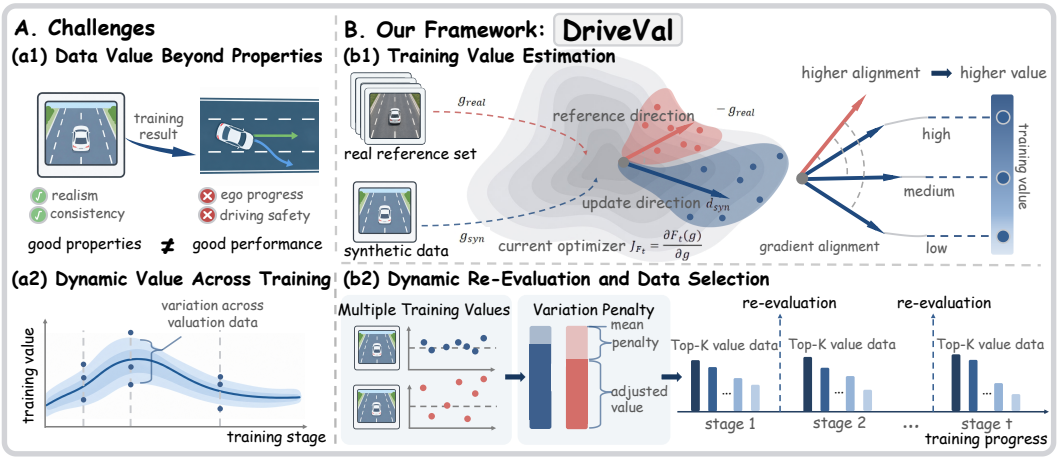

Figure 1: Overview of DriveVal. Data properties do not directly reflect training value, which changes during training. DriveVal evaluates synthetic data by their training effects, accounts for the current optimizer state and value variation, and periodically updates data valuation.

Applying existing valuation methods to synthetic data for end-to-end driving model training poses two fundamental challenges. (i) Data properties do not directly reflect training effects. Visual realism and diversity can describe synthetic data, but do not reveal whether training on them will actually improve end-to-end driving model performance. (ii) Data value is not fixed. The value of the same synthetic data can vary across training stages and can also be sensitive to the data used for valuation. A one-time valuation may therefore no longer reflect the training value of the data as training progresses.

To address these challenges, we propose DriveVal, which evaluates and selects synthetic driving data based on their training effects on end-to-end driving model performance rather than on data properties. DriveVal uses gradients computed from a small set of real driving data to define the reference optimization direction, and quantifies the training value of synthetic data using the dot product between the real-data gradient and the synthetic-data gradient under evaluation, indicating whether training on the synthetic data is consistent with reducing loss on the real data. Since data value varies as training progresses, DriveVal periodically re-evaluates synthetic data instead of relying on a single valuation throughout training. The valuation also accounts for the current optimizer state, allowing data value to reflect the current stage of training. To reduce sensitivity to a particular data choice, DriveVal evaluates the same synthetic data with multiple real references and synthetic-data combinations and penalizes variation across the resulting data values. These values are then used to select synthetic data for subsequent training. Figure 1 provides an overview of the two challenges and the overall framework of DriveVal.

Extensive experiments on NAVSIM demonstrate that DriveVal consistently improves end-to-end driving performance across diverse models trained with synthetic data. Additional experiments further confirm that DriveVal remains effective across different data generation sources and visual conditions, while ablation studies verify the contribution of each component.

In summary, our contributions are as follows:

- We propose DriveVal, a synthetic driving data valuation framework that evaluates data based on their training effects on end-to-end driving model performance and uses the resulting values to guide data selection.
- We quantify the training value of synthetic data through the dot product between real-data and synthetic-data gradients, allowing the resulting value to directly reflect training effects on real driving data.
- We periodically re-evaluate data throughout training, accounting for the current optimizer state, and penalize variation in data value across different real references and synthetic-

data combinations, allowing the valuation to adapt as training progresses while reducing sensitivity to particular data choices.

- Across five end-to-end driving models on NAVSIM, DriveVal achieves 95.2% of the mean PDMS of the corresponding real-data baselines and improves mean PDMS by 13.6% over size-matched random selection and 9.1% over training on the full synthetic set.

2 Related Work

Synthetic Driving Data Evaluation. Recent advances in controllable generation and driving world models have expanded the availability of synthetic driving data (Gao et al., 2024; Wang et al., 2024; Yang et al., 2025; Ge et al., 2025; Tian et al., 2026; Kim et al., 2025). Existing methods evaluate synthetic data from multiple dimensions. Some methods compare synthetic and real data to evaluate visual realism, while others evaluate temporal consistency through cross-frame coherence (Gao et al., 2024; Wang et al., 2024; Yang et al., 2025). Controllability and diversity further describe whether synthetic data follow specified conditions and cover different driving scenes (Ge et al., 2025; Tian et al., 2026; Kim et al., 2025). These methods evaluate synthetic data based on the properties of the data themselves, but do not directly reveal their training value for improving end-to-end driving model performance in real driving scenes.

Training-Effect-Based Data Valuation. Data valuation based on training effects evaluates data according to their contribution to model training. Gradients and held-out objectives are commonly used to determine whether particular data help reduce training or validation losses, whereas influence functions and attribution methods evaluate the effect of individual data on model predictions or parameter updates (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a;b; Mindermann et al., 2022; Potts & Everaert, 2024; Xia et al., 2024; Ghorbani & Zou, 2019; Koh & Liang, 2017; Basu et al., 2020; Pruthi et al., 2020). Compared with valuation based on data properties, these methods relate data value more directly to model training. However, many of these methods evaluate data at a fixed model state, overlooking how data value varies as training progresses.

Dynamic Data Valuation. Dynamic methods update data values throughout training instead of evaluating data at only one training state. Recent methods use changes in validation loss after each training update, gradient alignment with a direction expected to reduce training loss, or parameter updates produced during training to evaluate individual data (Wang et al., 2025; Gu et al., 2025; Wang et al., 2026b). These methods account for the evolving model state during training. However, the value of the same synthetic driving data can also vary with the current optimizer state during training, as well as with the real driving data used as references and the synthetic data evaluated together. This introduces an additional challenge that existing methods do not explicitly address.

3 Problem Definition

We consider a collection of synthetic driving data $\mathcal{G} = \{\mathcal{G}_i\}_{i=1}^{N}$, where each $\mathcal{G}_i$ contains temporally ordered training samples from one source sequence. Let $\mathcal{R} = \{\mathcal{R}_b\}_{b=1}^{B}$ denote a collection of real reference sets used to evaluate the training effects of synthetic data.

For each $\mathcal{G}_i$, let $\mathcal{C}_{i,c}$ denote the c-th synthetic-data combination containing $\mathcal{G}_i$. At training stage t, the model parameters and optimizer state are denoted by (θ_t, s_t). The training value of $\mathcal{G}_i$ under real reference $\mathcal{R}_b$ and synthetic-data combination $\mathcal{C}_{i,c}$ at training stage t is denoted by $V_{i,b,c}(t)$. It reflects the local effect of $\mathcal{G}_i$ on the loss evaluated on $\mathcal{R}_b$ under the current training state.

The value of the same synthetic data can vary as the model and optimizer states evolve during training and with the real references and synthetic-data combinations used for valuation. Synthetic driving data valuation should therefore account for both sources of variation. The resulting values can then guide which synthetic data participate in subsequent training.

4 METHODOLOGY

DriveVal evaluates synthetic driving data according to their training effects on the loss defined by real driving references. At a given training stage, it accounts for the current optimizer state when evaluating the training value of synthetic data. Since data value changes as training progresses, DriveVal periodically re-evaluates the data as the training state evolves. It further evaluates the same synthetic data across different real references and synthetic-data combinations and penalizes variation in the resulting values before using them for data selection.

4.1 TRAINING VALUE ESTIMATION

The training value defined in Section 3 reflects whether training on synthetic data helps reduce the loss on real driving references. We use gradients to describe this local training effect: the gradient on a real reference provides the direction associated with reducing its loss, while the gradient of synthetic data describes its contribution to model training.

Let $L_b^r(\theta)$ denote the loss evaluated on real reference $\mathcal{R}_b$, and let $L_{i,c}^s(\theta)$ denote the loss contribution of $\mathcal{G}_i$ when it is evaluated together with the synthetic data in $\mathcal{C}_{i,c}$. At training stage t, their gradients are

$$g_b^r(t) = \nabla_\theta L_b^r(\theta_t), \qquad g_{i,c}^s(t) = \nabla_\theta L_{i,c}^s(\theta_t). \tag{1}$$

Gradient alignment reflects training value based on whether the contribution of synthetic data to the parameter update is consistent with reducing the loss on the real reference. A direct gradient dot product evaluates this consistency, but the actual parameter update also depends on the current optimizer state. A gradient dot product alone therefore does not fully reflect the update produced by the optimizer.

To account for this, let $F_t(g)$ denote the parameter update produced by one step of the current optimizer at (θ_t, s_t) when supplied with gradient g. The optimizer state s_t includes the current learning rate, step counter, and moment or momentum buffers maintained by the optimizer. Let $g_{\mathrm{sh}}^s(t)$ denote a shared training gradient computed from a fixed set of synthetic training data. It provides a common training update for evaluating different synthetic-data groups at the same training stage. We define

$$\Delta(t) = F_t(g_{\mathrm{sh}}^s(t)) - F_t(0), \qquad d_{i,c}(t) = J_{F_t}(g_{\mathrm{sh}}^s(t))\, g_{i,c}^s(t). \tag{2}$$

Here, J_{F_t} denotes the Jacobian of F_t, $\Delta(t)$ denotes the update produced by the shared training gradient, and $d_{i,c}(t)$ describes the local change in this update induced by $g_{i,c}^s(t)$. The same $g_{\mathrm{sh}}^s(t)$ is used across all evaluations at training stage t, so differences in $d_{i,c}(t)$ reflect the contribution of $g_{i,c}^s(t)$ under the same optimizer update. Subtracting $F_t(0)$ removes the parameter update produced by the optimizer state and regularization when no gradient is supplied. For a simple gradient step, $d_{i,c}(t)$ is directly proportional to $g_{i,c}^s(t)$, recovering the gradient-alignment form.

We use this local update change to form the first-order estimate of training value:

$$V_{i,b,c}^{(1)}(t) = -\left[g_b^r(t)\right]^\top d_{i,c}(t). \tag{3}$$

A positive $V_{i,b,c}^{(1)}(t)$ indicates that the update contribution from $\mathcal{G}_i$ is predicted to reduce the loss on $\mathcal{R}_b$.

The first-order estimate treats the loss on the real reference as locally linear. However, the contribution of $\mathcal{G}_i$ is evaluated together with the shared training update, so their interaction also depends on the local curvature of the loss on the real reference. We therefore define

$$H_b^r(t) = \nabla_\theta^2 L_b^r(\theta_t), \tag{4}$$

and, following the local second-order attribution form of In-Run Data Shapley (Wang et al., 2025), include this interaction as

$$V_{i,b,c}^{(2)}(t) = V_{i,b,c}^{(1)}(t) - \frac{1}{2} d_{i,c}(t)^\top H_b^r(t)\Delta(t). \tag{5}$$

The second-order term accounts for the local curvature through $d_{i,c}(t)^\top H_b^r(t)\Delta(t)$. We compute this term through the Hessian–vector product $H_b^r(t)\Delta(t)$, without explicitly forming the full Hessian. We therefore use $V_{i,b,c}^{(2)}(t)$ as the training value of $\mathcal{G}_i$ under real reference $\mathcal{R}_b$ and synthetic-data combination $\mathcal{C}_{i,c}$ at training stage t.

4.2 Dynamic Valuation and Data Selection

The second-order training value $V^{(2)}_{i,b,c}(t)$ depends on the current model and optimizer states. As these states evolve during training, the parameter-update contribution of the same synthetic data and its local effect on the loss on real references can also change, so the same synthetic data can have different training values across training stages. To account for these changes, DriveVal periodically re-evaluates synthetic data under the current model and optimizer states. Training then continues from the current state without resetting the model parameters, optimizer state, or learning-rate schedule.

While periodic re-evaluation accounts for changes in training value across training stages, variation can still remain within a fixed training stage because different real references and synthetic-data combinations are used for valuation. In particular, $\mathcal{G}_i$ is evaluated jointly with the other synthetic data in $\mathcal{C}_{i,c}$, so its gradient and resulting training value can change across different combinations. DriveVal therefore evaluates each $\mathcal{G}_i$ with multiple real references and synthetic-data combinations.

To account for this variation, we first aggregate these training values to obtain the mean training value of $\mathcal{G}_i$ at training stage t. For each real reference $\mathcal{R}_b$, we average the second-order training values across the synthetic-data combinations evaluated with that reference:

$$\bar{V}_{i,b}(t) = \mathbb{E}_{c|b}\left[V^{(2)}_{i,b,c}(t)\right]. \tag{6}$$

We then combine the mean values from all real references to obtain

$$\bar{V}_i(t) = \sum_{b=1}^{B} w_b\, \bar{V}_{i,b}(t), \tag{7}$$

where w_b denotes the normalized weight of real reference $\mathcal{R}_b$.

The mean training value alone does not reflect how the value changes across different real references and synthetic-data combinations. However, directly computing variation over all $V^{(2)}_{i,b,c}(t)$ would mix these two sources of variation. We therefore separate variation across synthetic-data combinations from variation across real references using $\bar{V}_{i,b}(t)$ and $\bar{V}_i(t)$.

To compute variation across synthetic-data combinations separately, we subtract $\bar{V}_{i,b}(t)$ from each value evaluated under $\mathcal{R}_b$, removing the mean difference associated with the real reference. We then add back the overall mean $\bar{V}_i(t)$ so that the adjusted values under different real references share the same mean:

$$\widetilde{V}_{i,b,c}(t) = V^{(2)}_{i,b,c}(t) - \bar{V}_{i,b}(t) + \bar{V}_i(t). \tag{8}$$

This adjustment preserves the differences among synthetic-data combinations evaluated under the same real reference while removing the mean differences across real references. The variation of $\widetilde{V}_{i,b,c}(t)$ can therefore be computed separately from the variation of $\bar{V}_{i,b}(t)$ across real references.

We compute $\mathrm{SE}_{i,\mathrm{comb}}(t)$ from the variation of $\widetilde{V}_{i,b,c}(t)$ across synthetic-data combinations, and compute $\mathrm{SE}_{i,\mathrm{ref}}(t)$ from the variation of $\bar{V}_{i,b}(t)$ across real references. The first term reflects how the training value changes with the synthetic data evaluated together, while the second reflects how it changes across real references. The detailed computation of the two terms is provided in Appendix B.3.

We combine the two terms and subtract the resulting variation penalty from the mean training value:

$$\widehat{V}_i(t) = \bar{V}_i(t) - \lambda\sqrt{\mathrm{SE}^2_{i,\mathrm{comb}}(t) + \mathrm{SE}^2_{i,\mathrm{ref}}(t)}. \tag{9}$$

Here, $\lambda \geq 0$ controls the strength of the variation penalty. A higher mean training value increases $\widehat{V}_i(t)$, while larger variation decreases it.

At training stage t, DriveVal ranks the synthetic data by $\widehat{V}_i(t)$. Let $\mathcal{S}_t$ denote the set of synthetic-data groups selected for subsequent training. We select the K groups with the highest adjusted training values:

$$\mathcal{S}_t = \mathrm{TopK}_K\left(\{\widehat{V}_i(t)\}_{i=1}^{N}\right). \tag{10}$$

The selected synthetic data are used for subsequent training until the next re-evaluation, when their training values are recomputed under the updated model and optimizer states.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate DriveVal on NAVSIM (Dauner et al., 2024). The synthetic training data are constructed from OpenScene sequences (OpenScene Contributors, 2023) excluded from navtrain, with the three front-camera observations generated by DriveArena (Yang et al., 2025). The resulting synthetic set contains 206,576 training samples. DriveVal trains on 103,288 synthetic samples per epoch, matching the number of samples in navtrain. We compare it with a random baseline that selects the same number of synthetic samples without data valuation (Random), and with training on the full 206,576-sample synthetic set. For the full synthetic set, we halve the number of training epochs to keep the number of optimizer updates and the total number of processed training samples consistent with the other synthetic-data settings. Real navtrain data provide the reference sets for valuation, while final driving performance is reported on navtest using PDMS.

We evaluate DriveVal on five end-to-end driving models: DiffusionDrive (Liao et al., 2025), Latent TransFuser (Dauner et al., 2024), MeanFuser (Wang et al., 2026a), WoTE (Li et al., 2025), and DrivoR (Kirby et al., 2026). For each model, we use its officially released checkpoint trained on real navtrain data as the real-data baseline. Each model follows its official training configuration. Apart from the synthetic data used for training and the epoch adjustment for the full synthetic set described above, the training settings are kept identical within each model.

DriveVal uses three disjoint real reference sets for valuation, with an additional disjoint real set reserved for valuation analysis. We use $\lambda = 1$ and a five-epoch re-evaluation interval across all models, generation sources, and visual conditions. Sensitivity to λ is analyzed in Appendix D. Detailed data construction, valuation settings, and model-specific training configurations are provided in the Appendix. All experiments are conducted on NVIDIA L40 GPUs.

5.2 MAIN RESULTS

Table 1 reports the final-checkpoint navtest PDMS across the five end-to-end driving models. DriveVal achieves an average PDMS of 84.24, compared with 74.17 for Random, corresponding to a 13.6% improvement. Since Random uses the same number of synthetic training samples without data valuation, the result indicates that selecting synthetic data according to their training effects improves driving performance over size-matched random selection.

Table 1: NAVSIM navtest PDMS across five end-to-end driving models. DriveVal achieves the highest PDMS among the synthetic-data configurations on all five models, increasing the model average from 74.17 with Random and 77.23 with the Full Synthetic Set to 84.24. Bold marks the best synthetic-data result.

	Synthetic Data			Real Data
Model	Random	Full Synthetic Set	DriveVal	navtrain Baseline
DiffusionDrive (Liao et al., 2025)	73.77	77.87	**83.28**	88.10
Latent TransFuser (Dauner et al., 2024)	71.42	75.46	**80.84**	83.80
MeanFuser (Wang et al., 2026a)	73.32	77.13	**83.94**	89.02
WoTE (Li et al., 2025)	75.70	76.40	**84.28**	88.30
DrivoR (Kirby et al., 2026)	76.66	79.28	**88.85**	93.13
Model average	74.17	77.23	**84.24**	88.47

Compared with the Full Synthetic Set, DriveVal improves the average PDMS from 77.23 to 84.24, corresponding to a 9.1% improvement. The two configurations use the same number of optimizer updates and total processed training samples, while the Full Synthetic Set uses all available synthetic data. This result shows that increasing the amount of synthetic data alone does not lead to better driving performance, and that selecting synthetic data according to their training effects is more effective than training on the full synthetic set.

At the model level, the gains remain consistent across all five end-to-end driving models. Relative to Random, DriveVal improves PDMS by 11.3%–15.9%, while the improvements over the Full Synthetic Set range from 6.9% to 12.1%. The real-data baselines achieve an average PDMS of 88.47, and DriveVal reaches 95.2% of this performance. Across individual models, DriveVal reaches 94.3%–96.5% of the corresponding real-data baseline. These results show that the advantage of selecting synthetic data according to their training effects remains across the evaluated models.

Table 2 further reports the individual planning metrics. DriveVal achieves higher model-average NC, DAC, EP, and TTC than both Random and the Full Synthetic Set, while Comfort remains nearly unchanged across the compared configurations. The improvement is therefore not concentrated in a single planning metric.

Table 2: Model-average NAVSIM navtest planning metrics across the five end-to-end driving models. NC denotes no-at-fault collision, DAC drivable-area compliance, EP ego progress, and TTC time-to-collision score. Bold marks the highest value in each column.

Configuration	NC ↑	DAC ↑	EP ↑	TTC ↑	Comfort ↑
Random	95.21	84.46	70.15	88.41	99.98
Full Synthetic Set	95.47	87.55	72.51	89.16	**99.99**
DriveVal	**96.93**	**94.62**	**78.86**	**91.64**	99.98

5.3 Ablation Study

We conduct the ablation study on DiffusionDrive to examine the main components of DriveVal. We first compare training-value estimation with Random. At a fixed training stage, we then separately examine the effects of accounting for the current optimizer state and penalizing variation across real references and synthetic-data combinations, while keeping the second-order training-value estimation unchanged. Finally, we add periodic re-evaluation to the configuration that includes both components to examine the effect of dynamic valuation during training.

Table 3: Ablation study on DiffusionDrive. Starting from training-value estimation, we examine the current optimizer state and variation penalty separately and together, followed by periodic re-evaluation. All valuation configurations use the second-order estimate in Equation 5. Gain is measured relative to Random; bold marks the highest PDMS and gain.

Training-value estimation	Optimizer state	Variation penalty	Periodic re-evaluation	PDMS	Gain vs. Random
–	–	–	–	73.77	–
✓	–	–	–	76.57	+3.8%
✓	–	✓	–	78.15	+5.9%
✓	✓	–	–	77.96	+5.7%
✓	✓	✓	–	79.32	+7.5%
✓	✓	✓	✓	**83.28**	**+12.9%**

Training-value estimation improves PDMS over Random. At a fixed training stage, accounting for the current optimizer state and penalizing variation each provide further improvements. When both components are included, PDMS improves by 7.5% over Random. Adding periodic re-evaluation to this configuration provides a further 5.0% improvement. These results show that accounting for the optimizer state and variation improves valuation at a fixed stage, while periodic re-evaluation provides an additional gain as training progresses.

5.4 Training-Value Correlation and Evolution

We analyze whether the estimated training values reflect the training effects of synthetic data, how the variation penalty affects data selection, and how training values and selected data change during training.

To examine the effect of the current optimizer state, we compare the estimated training values with the loss reductions on real references after the corresponding parameter updates at a fixed DiffusionDrive training state. If the estimated values reflect the training effects of synthetic data, data with higher training values should correspond to larger loss reductions. Without accounting for the current optimizer state, the second-order estimator gives an average Spearman correlation of 0.0341. Accounting for the optimizer state increases the correlation to 0.7119. The increase in correlation shows that accounting for the current optimizer state makes the estimated training values more consistent with the loss reductions on real references.

We evaluate the variation penalty by comparing the ranking based on mean training value alone with the ranking obtained after applying the penalty. We compare both rankings with the ranking based on loss reductions on the disjoint real reference set reserved for valuation analysis. Applying the variation penalty increases the Spearman correlation with this ranking by 0.1426 and increases the average loss reduction of the selected Top-K data by 0.0061. The corresponding 95% bootstrap intervals are both above zero. Synthetic data with larger variation also show larger differences between their ranks based on mean training value and loss reduction on the disjoint real reference data, with a Spearman correlation of 0.5990. Together, these results show that the variation penalty gives a ranking more consistent with the loss reductions on disjoint real data and increases the loss reduction of the selected Top-K data.

Table 4: Training-value analysis on DiffusionDrive. Accounting for the current optimizer state increases the agreement between estimated training values and loss reductions on real references. The variation penalty further improves the ranking on disjoint real data, while larger variation is associated with larger rank differences from the mean training value. Intervals are 95% percentile-bootstrap intervals.

Analysis	Statistic	Value	95% CI
Training-value estimation			
Without optimizer state	Mean Spearman ρ	0.0341	–
With optimizer state	Mean Spearman ρ	0.7119	–
Variation penalty on disjoint real data			
Rank agreement	Increase in Spearman ρ	0.1426	$[0.0770, 0.2500]$
Selected Top-K data	Increase in loss reduction	0.0061	$[0.0026, 0.0109]$
Variation vs. mean-value rank difference	Spearman ρ	0.5990	$[0.2808, 0.7362]$

At each re-evaluation, DriveVal ranks all synthetic-data groups according to their current training values and selects the Top-K groups for subsequent training. To examine how these results change during training, we compare each re-evaluation with the preceding one. We measure the similarity between the two rankings of synthetic-data groups using Spearman correlation, where a higher correlation indicates a more similar ordering of the data. We also compute the fraction of selected Top-K groups that changes between the two stages.

As shown in Figure 2, the rankings of synthetic-data groups are less similar during earlier training stages and become more similar later in training. The selected data follow the same pattern, with more data changing during earlier stages and fewer changes later. These results show that both the ordering of synthetic data and the data selected for training change as training progresses, particularly during earlier stages. Periodic re-evaluation allows DriveVal to update the selected data as their training values change.

5.5 Across Generation Sources and Visual Conditions

We further evaluate DriveVal across different generation sources and visual conditions. We use front-camera images from three generation sources, DriveArena (Yang et al., 2025), BridgeSim (Zhao et al., 2026), and WorldEngine (Li et al., 2026), and evaluate two additional visual conditions by applying fog and Gaussian noise at severity 5 to real front-camera images. For these experiments, the synthetic training set contains 103,288 samples, from which DriveVal and Random each use 51,644 samples per epoch.

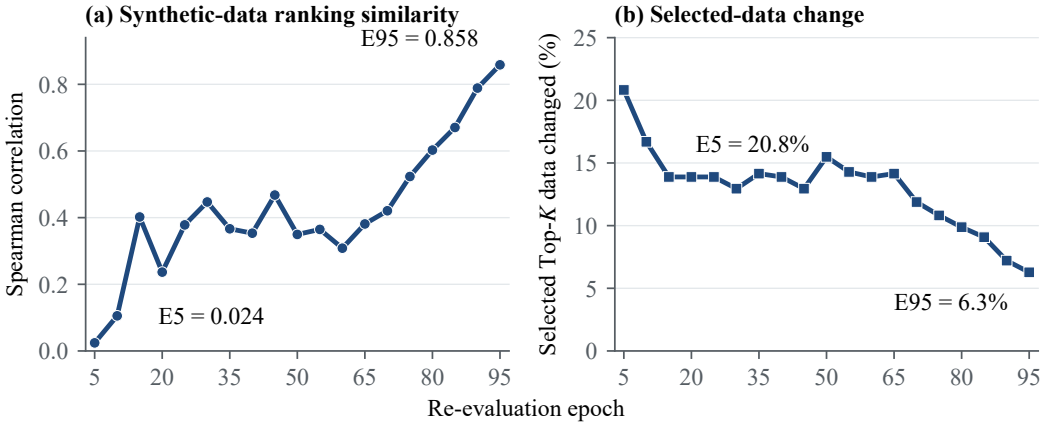

Figure 2: Changes in synthetic-data rankings and selected data across DriveVal re-evaluations on DiffusionDrive. (a) Spearman correlation between the current and preceding synthetic-data rankings. (b) Fraction of selected Top-K data changed from the preceding re-evaluation. Both become more stable later in training.

Table 5: NAVSIM navtest PDMS across three generation sources and two visual conditions. Each entry reports Random / DriveVal. DriveVal achieves higher PDMS in all 25 model–condition comparisons. The final row reports the equal-weight average across the five models; bold marks the higher value.

	Generation Sources			Visual Conditions	
Model	DriveArena	BridgeSim	WorldEngine	Fog	Gaussian Noise
DiffusionDrive	68.48 / **79.96**	64.70 / **78.16**	71.59 / **82.24**	74.12 / **83.71**	66.87 / **78.60**
Latent TransFuser	69.05 / **77.94**	62.49 / **74.98**	72.19 / **79.19**	69.27 / **79.14**	68.38 / **75.91**
MeanFuser	70.31 / **79.43**	65.06 / **76.76**	73.18 / **80.73**	72.65 / **81.73**	67.96 / **75.52**
WoTE	70.78 / **80.45**	68.40 / **76.20**	73.86 / **82.24**	75.13 / **82.68**	69.46 / **77.92**
DrivoR	70.40 / **82.35**	70.11 / **79.64**	72.56 / **84.50**	73.36 / **85.21**	69.47 / **78.62**
Model average	69.80 / **80.03**	66.15 / **77.15**	72.68 / **81.78**	72.91 / **82.49**	68.43 / **77.31**

DriveVal improves PDMS over Random in all 25 comparisons across the five models and the evaluated generation sources and visual conditions. At the model-average level, the relative improvements range from 12.5% to 16.6%, with an average improvement of 13.9%. Across the three generation sources, the average relative improvement is 14.5%, while the two visual conditions show an average improvement of 13.1%. As shown in Table 5, the improvements remain across all evaluated models, generation sources, and visual conditions. These results show that the performance gains from DriveVal remain across different generation sources and visual conditions.

6 Conclusion

In this paper, we presented DriveVal, which evaluates synthetic driving data according to their training effects on end-to-end driving model performance and uses the resulting values to guide data selection. DriveVal quantifies training value by whether training on synthetic data helps reduce the loss on real references. It periodically re-evaluates synthetic data while accounting for the current optimizer state and penalizes variation across different real references and synthetic-data combinations, allowing the valuation to adapt as training progresses while reducing sensitivity to particular data choices. Across five end-to-end driving models on NAVSIM, DriveVal improves end-to-end driving performance in real driving scenes over random selection and training on the full synthetic set, with the improvement remaining across different generation sources and visual conditions.

AI USE STATEMENT

Generative AI tools were used for language and LaTeX editing during manuscript preparation. The authors reviewed all AI-assisted text and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The experimental protocol, data-role isolation, optimizer and gradient checks, model-specific controls, and evaluation procedure are documented in Sections 3–5 and Appendices A–F.

REFERENCES

Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 715–724, 2020. URL https://proceedings.mlr.press/v119/basu20b.html.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/32768f7faf1995026ef9821c696f3404-Abstract-Datasets_and_Benchmarks_Track.html.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sBQwvucduK.

Junhao Ge, Zuhong Liu, Longteng Fan, Yifan Jiang, Jiaqi Su, Yiming Li, Zhejun Zhang, and Siheng Chen. Unraveling the effects of synthetic data on end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28859–28869, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Ge_Unraveling_the_Effects_of_Synthetic_Data_on_End-to-End_Autonomous_Driving_ICCV_2025_paper.html.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2242–2251, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9ad4891facabf17aa11580686bacfe4e-Abstract-Conference.html.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5464–5474, 2021a. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021b. doi: 10.1609/aaai.v35i9.16988.

Jongsuk Kim, Jaeyoung Lee, Gyojin Han, Dong-Jae Lee, Minki Jeong, and Junmo Kim. SynAD: Enhancing real-world end-to-end autonomous driving models through synthetic data integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 25197–25206, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Kim_

SynAD_Enhancing_Real-World_End-to-End_Autonomous_Driving_Models_through_Synthetic_Data_ICCV_2025_paper.html.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung Vu, and Matthieu Cord. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32058–32069, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Kirby_Driving_on_Registers_CVPR_2026_paper.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1885–1894, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Tianyu Li, Li Chen, Caojun Wang, Haochen Liu, Kashyap Chitta, Zhenjie Yang, Yuhang Lu, Naisheng Ye, Yihang Qiu, Yufei Wang, Luoxi Zou, Jiaxin Peng, Jin Pan, Zhaoyu Su, Andrei Bursuc, Shengbo Eben Li, Andreas Geiger, Peng Su, and Hongyang Li. World engine: Towards the era of post-training for autonomous driving. *arXiv preprint arXiv:2606.19836*, 2026. doi: 10.48550/arXiv.2606.19836. URL https://arxiv.org/abs/2606.19836.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via BEV world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27137–27146, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Li_End-to-End_Driving_with_Online_Trajectory_Evaluation_via_BEV_World_Model_ICCV_2025_paper.html.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12037–12047, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liao_DiffusionDrive_Truncated_Diffusion_Model_for_End-to-End_Autonomous_Driving_CVPR_2025_paper.html.

Sören Mindermann, Jan M. Brauner, Muhammed T. Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 15630–15649, 2022. URL https://proceedings.mlr.press/v162/mindermann22a.html.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 6950–6960, 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

OpenScene Contributors. OpenScene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. GitHub repository, 2023. URL https://github.com/OpenDriveLab/OpenScene.

Christopher Potts and Dante Everaert. GIO: Gradient information optimization for training dataset selection. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3NnfJnbJT2.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Ye, Long Chen, and Hongyang Li.

Simscale: Learning to drive via real-world simulation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36365–36374, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Tian_SimScale_Learning_to_Drive_via_Real-World_Simulation_at_Scale_CVPR_2026_paper.html.

Jiachen T. Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HD6bWcj87Y.

Junli Wang, Xueyi Liu, Yinan Zheng, Zebing Xing, Pengfei Li, Guang Li, Kun Ma, Guang Chen, Hangjun Ye, Zhongpu Xia, Long Chen, and Qichao Zhang. MeanFuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via MeanFlow for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Wang_MeanFuser_Fast_One-Step_Multi-Modal_Trajectory_Generation_and_Adaptive_Reconstruction_via_CVPR_2026_paper.pdf.

Shaobo Wang, Xuan Ouyang, Tianyi Xu, Yuzheng Hu, Jialin Liu, Guo Chen, Tianyu Zhang, Junhao Zheng, Kexin Yang, Xingzhang Ren, Dayiheng Liu, and Linfeng Zhang. OPUS: Towards efficient and principled data selection in large language model pre-training in every iteration. *arXiv preprint arXiv:2602.05400*, 2026b. URL https://arxiv.org/abs/2602.05400.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14749–14759, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wang_Driving_into_the_Future_Multiview_Visual_Forecasting_and_Planning_with_CVPR_2024_paper.html.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 54104–54132, 2024. URL https://arxiv.org/abs/2402.04333.

Xuemeng Yang, Licheng Wen, Tiantian Wei, Yukai Ma, Jianbiao Mei, Xin Li, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, Liang He, Yong Liu, Botian Shi, and Yu Qiao. Drivearena: A closed-loop generative simulation platform for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26933–26943, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Yang_DriveArena_A_Closed-loop_Generative_Simulation_Platform_for_Autonomous_Driving_ICCV_2025_paper.html.

Seth Z. Zhao, Luobin Wang, Hongwei Ruan, Yuxin Bao, Yilan Chen, Ziyang Leng, Abhijit Ravichandran, Honglin He, Zewei Zhou, Xu Han, Abhishek Peri, Zhiyu Huang, Pranav Desai, Henrik Christensen, Jiaqi Ma, and Bolei Zhou. BridgeSim: Unveiling the OL-CL gap in end-to-end autonomous driving. *arXiv preprint arXiv:2604.10856*, 2026. doi: 10.48550/arXiv.2604.10856. URL https://arxiv.org/abs/2604.10856.

A Experimental Details

This section provides additional details on data construction, model training, and the controls used to keep the experimental settings consistent across comparisons.

A.1 Data Construction and Source Isolation

The synthetic training data are constructed from OpenScene sequences excluded from NAVSIM navtrain. The three real reference sets used for valuation and the additional disjoint real set used for valuation analysis are constructed from navtrain, while navtest is used only for final driving

evaluation. We verify that the synthetic-data source sequences do not overlap with either the real reference data or navtest at the sequence or token level.

For each source sequence, the synthetic training samples pair DriveArena-generated front-camera observations with the corresponding training targets and auxiliary inputs from the source sequence. We additionally check image decoding, dimensions, camera order, token mapping, and data provenance to confirm that the synthetic images are matched to the corresponding training samples.

The main synthetic set contains 206,576 samples from 938 source sequences, where each source sequence corresponds to one synthetic-data group $\mathcal{G}_i$. DriveVal and Random each use $M = 103{,}288$ unique training samples per epoch. The selected groups are allocated training samples according to the number of available samples in each group so that the total number of training samples remains fixed at M. The sample allocation first limits each group by its available samples and then distributes the remaining samples among groups with unused samples. Within each group, training samples are taken at evenly spaced temporal positions. Random uses the same number of groups and the same number of training samples per epoch.

DriveVal and Random train for 100 epochs. The Full Synthetic Set uses all 938 source sequences and 206,576 samples per epoch for 50 epochs, resulting in the same number of optimizer updates and the same total number of processed training samples. For the experiments across generation sources and visual conditions, we use the same construction procedure with 103,288 available training samples, from which DriveVal and Random each use 51,644 samples per epoch.

A.2 Model-Specific Training Configurations

Each end-to-end driving model follows its official initialization, trainable parameter groups, optimizer, learning-rate schedule, data augmentation, and precision mode. The Full Synthetic Set uses the epoch adjustment described above. Training and valuation use AdamW for DiffusionDrive, MeanFuser, WoTE, and DrivoR, and Adam for Latent TransFuser. Accordingly, F_t represents one step of the optimizer configured for the corresponding model.

For DiffusionDrive, training uses four NVIDIA L40 GPUs with a batch size of 32 per device and gradient accumulation of 4, giving a global batch size of 512. We use AdamW with a peak learning rate of 6×10^{-4} and weight decay of 10^{-4}. Training contains 202 optimizer updates per epoch and 20,200 updates in total, with a three-epoch warmup followed by the official cosine learning-rate schedule. Training uses 16-bit mixed precision and four data workers per device. All DiffusionDrive configurations start from the same official Stage-1 checkpoint with a newly initialized optimizer state. The remaining models follow their corresponding official training configurations under the same within-model controls.

A.3 Reproducibility Controls

We use fixed manifests to record the synthetic source sequences, real reference data, synthetic-data combinations, training samples, and navtest data used by each experiment. The experiment records additionally include checkpoint identifiers, random seeds, optimizer and learning-rate scheduler settings, global batch size, number of training updates, rendering settings, data checks, and the evaluation command. Main results are reported from the predefined final checkpoint.

Within each model, all configurations use the same initialization, training procedure, and final evaluation protocol, with the epoch adjustment for the Full Synthetic Set described above.

B Valuation Implementation Details

This section provides additional implementation details for the valuation procedure described in Section 4.

B.1 Real References and Synthetic-Data Combinations

DriveVal uses three mutually disjoint real reference sets, each containing 8 navtrain sequences and 80 samples. An additional disjoint real set of the same size is reserved for the valuation analysis in Section 5.4 and is not used to compute the training values used for data selection.

The valuation schedule contains 472 synthetic-data combinations, each containing 16 synthetic-data groups. When $\mathcal{G}_i$ is evaluated in one combination, it is evaluated together with 15 other synthetic-data groups. Each $\mathcal{G}_i$ appears in 8–10 combinations and is evaluated together with 103–128 distinct other synthetic-data groups across the complete schedule, with a median of 113.

The number of available samples differs across synthetic-data groups. Let $\mathcal{I}_i$ contain the c_i physical samples in $\mathcal{G}_i$. Groups with $c_i \geq 128$ use 128 unique samples selected at temporally spaced positions, groups with $33 \leq c_i < 128$ use all available samples, and groups containing 32 samples use all 32 samples. When the fixed computation layout requires samples from a shorter group to be repeated, each occurrence is weighted by the inverse of its multiplicity. With u_i unique samples, computational multiset $\widetilde{\mathcal{I}}_i$, and multiplicity m_{ij}, the group loss is

$$L_i(\theta) = \frac{1}{u_i} \sum_{j \in \widetilde{\mathcal{I}}_i} \frac{1}{m_{ij}} \ell_j^{\text{native}}(\theta). \tag{11}$$

The same weighting is used when extracting the contribution of $\mathcal{G}_i$ from a synthetic-data combination. This prevents repeated tensor positions from being treated as additional physical samples.

Table 6 compares this weighting with direct computation over all unique samples for groups containing fewer than 128 samples. Across the tested group sizes, the relative loss error ranges from 0.153% to 0.710%, while the relative gradient error ranges from 0.84% to 1.91%.

Table 6: Loss and gradient errors of inverse-repeat weighting relative to direct computation over the unique samples. Multiplicity denotes the number of occurrences of each unique sample in the 128-position valuation input.

Unique samples	Multiplicity	Loss error ↓	Gradient error ↓
33	3–4	0.710%	1.91%
47	2–3	0.646%	1.55%
79	1–2	0.153%	1.37%
127	1–2	0.191%	0.84%

B.2 Shared Training Gradient and Synthetic-Data Gradients

At training stage t, the shared training gradient $g_{\text{sh}}^s(t)$ is computed from a fixed set of synthetic training data $\mathcal{B}_{\text{sh}}$:

$$g_{\text{sh}}^s(t) = \frac{1}{|\mathcal{B}_{\text{sh}}|} \sum_{x \in \mathcal{B}_{\text{sh}}} \nabla_\theta \ell(x; \theta_t). \tag{12}$$

The samples in $\mathcal{B}_{\text{sh}}$ remain unchanged throughout training, while the gradient is recomputed under the current model parameters at each re-evaluation.

All five models use the same construction procedure, with the number of samples following their model-specific training configurations. For DiffusionDrive, $\mathcal{B}_{\text{sh}}$ contains 512 samples distributed over $R = 4$ GPUs, each using $A = 4$ microbatches of 32 samples. With $L_{r,a}^{\text{native}}$ denoting the native training loss on GPU r and microbatch a, we compute

$$g_{\text{sh}}^s(t) = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{A} \sum_{a=1}^{A} \nabla_\theta L_{r,a}^{\text{native}}(\theta_t). \tag{13}$$

The losses are accumulated across microbatches, synchronized across GPUs, and unscaled before the gradients are read from the optimizer. No optimizer update is performed when computing $g_{\text{sh}}^s(t)$.

For each synthetic-data combination $\mathcal{C}_{i,c}$, the model's native training loss is computed jointly over the data in the combination. The loss is decomposed into per-sample terms, and the contribution of $\mathcal{G}_i$ is extracted using the inverse-repeat weights in Equation 11, giving $g^s_{i,c}(t)$. The resulting gradients are averaged across GPUs using the same reduction convention as $g^s_{\mathrm{sh}}(t)$. Therefore, $g^s_{i,c}(t)$ and $g^s_{\mathrm{sh}}(t)$ use the same loss reduction and gradient normalization.

Evaluating the synthetic data jointly preserves the corresponding computation in the model's original training procedure. Consequently, $g^s_{i,c}(t)$ can change when the other synthetic data in $\mathcal{C}_{i,c}$ change, while $g^s_{\mathrm{sh}}(t)$ remains shared across all evaluations at the same training stage.

As defined in Section 4.1,

$$\Delta(t) = F_t(g^s_{\mathrm{sh}}(t)) - F_t(0), \qquad d_{i,c}(t) = J_{F_t}(g^s_{\mathrm{sh}}(t))\, g^s_{i,c}(t). \tag{14}$$

Evaluating F_t and its Jacobian–vector product does not modify the saved model or optimizer state.

B.3 Variation-Penalty Computation

At a fixed training stage t, each valuation result for synthetic-data group $\mathcal{G}_i$ is associated with a real reference and a synthetic-data combination. Let $\mathcal{O}_{i,b}$ denote the valuation results of $\mathcal{G}_i$ obtained with real reference $\mathcal{R}_b$. We compute the mean second-order training value under this reference as

$$\bar{V}_{i,b}(t) = \frac{1}{|\mathcal{O}_{i,b}|} \sum_{o \in \mathcal{O}_{i,b}} V^{(2)}_{i,b,c(o)}(t), \tag{15}$$

where $c(o)$ denotes the synthetic-data combination associated with valuation result o. The mean training value across real references is

$$\bar{V}_i(t) = \sum_{b=1}^{B} w_b\, \bar{V}_{i,b}(t), \tag{16}$$

where w_b denotes the normalized weight of real reference $\mathcal{R}_b$. The three real reference sets used in our experiments have equal weights.

We next compute variation across synthetic-data combinations while separating it from variation across real references. Let $\mathcal{C}_i$ denote the set of distinct synthetic-data combinations in which $\mathcal{G}_i$ is evaluated, and let $n_i = |\mathcal{C}_i|$. For each combination $c \in \mathcal{C}_i$, let $\mathcal{O}_{i,c}$ denote its corresponding valuation results and let $b(c)$ denote the real reference used with that combination. We first compute the mean training value for the combination:

$$\bar{V}_{i,c}(t) = \frac{1}{|\mathcal{O}_{i,c}|} \sum_{o \in \mathcal{O}_{i,c}} V^{(2)}_{i,b(c),c}(t). \tag{17}$$

We then remove the mean associated with its real reference and add back the overall mean training value:

$$\widetilde{V}_{i,c}(t) = \bar{V}_{i,c}(t) - \bar{V}_{i,b(c)}(t) + \bar{V}_i(t). \tag{18}$$

This removes differences associated with the real reference before variation across synthetic-data combinations is computed.

Let

$$\overline{\widetilde{V}}_i(t) = \frac{1}{n_i} \sum_{c \in \mathcal{C}_i} \widetilde{V}_{i,c}(t). \tag{19}$$

The standard-error term across synthetic-data combinations is

$$\mathrm{SE}^2_{i,\mathrm{comb}}(t) = \frac{1}{n_i(n_i-1)} \sum_{c \in \mathcal{C}_i} \left(\widetilde{V}_{i,c}(t) - \overline{\widetilde{V}}_i(t) \right)^2. \tag{20}$$

Variation across real references is computed directly from the reference-level means. We first compute

$$s^2_{i,\mathrm{ref}}(t) = \frac{\sum_{b=1}^{B} w_b \left(\bar{V}_{i,b}(t) - \bar{V}_i(t) \right)^2}{1 - \sum_{b=1}^{B} w_b^2}, \tag{21}$$

and obtain the corresponding standard-error term as

$$\mathrm{SE}^2_{i,\mathrm{ref}}(t) = s^2_{i,\mathrm{ref}}(t) \sum_{b=1}^{B} w_b^2. \tag{22}$$

The two terms are combined through the variation penalty in Section 4.2, giving the adjusted training value

$$\widehat{V}_i(t) = \bar{V}_i(t) - \lambda \sqrt{\mathrm{SE}^2_{i,\mathrm{comb}}(t) + \mathrm{SE}^2_{i,\mathrm{ref}}(t)}. \tag{23}$$

If fewer than two distinct units are available for either source of variation, the corresponding standard-error term is set to zero. Unequal numbers of valuation results are handled through the reference-level and combination-level means above and do not change the real-reference weights.

DriveVal uses the second-order training values in Equation 5 for data selection. The same aggregation is applied to the first-order training values in the additional comparisons reported in the Appendix.

B.4 Second-Order Computation and Optimizer Verification

The second-order training value in Equation 5 is computed through the Hessian–vector product $H_b^r(t)\Delta(t)$, obtained by automatic differentiation of the loss on the real reference. Within each real reference, the same normalized weights are used to aggregate the losses for both the real-reference gradient and the Hessian–vector product.

At a given training stage, $g_b^r(t)$ and $H_b^r(t)\Delta(t)$ are computed once for each real reference and reused for all synthetic-data combinations evaluated with that reference. Each $d_{i,c}(t)$ is then contracted with the corresponding Hessian–vector product to compute the second-order term in Equation 5.

All gradients, Jacobian–vector products, and Hessian–vector products are computed over the complete parameter set used by the configured optimizer. F_t is evaluated with the current model parameters, supplied gradient, and optimizer state, including the learning rate, step counter, first- and second-moment or momentum buffers, and configured weight decay. Evaluating F_t does not modify the model or optimizer state.

Before the full valuation experiments, we verify that the gradients used by DriveVal match those produced by the model's original training loss. We also compare the displacement and Jacobian–vector product produced by F_t with the corresponding Adam or AdamW update and verify the real-reference Hessian–vector product used in the second-order term.

B.5 Dynamic Re-Evaluation Protocol

DriveVal is trained for 100 epochs and re-evaluates the synthetic data every five epochs. Training values are first computed at the initial training state, and the selected data are used for the first five epochs. After each five-epoch interval, the synthetic data are re-evaluated under the current model and optimizer states, and the newly selected data are used for the next five epochs.

When the selected data change, training continues without resetting the model parameters, optimizer state, or learning-rate schedule. The optimizer moments and step counters, learning-rate scheduler and its progress, global training step, AMP scaler state where applicable, and random states are preserved across re-evaluations. Dynamic data valuation therefore proceeds along one continuous training trajectory.

The configuration without periodic re-evaluation uses the same initial training values and selected data as DriveVal but keeps these data unchanged throughout training. The two configurations therefore share the same initial valuation and the same first five epochs of training.

The re-evaluation schedule uses the same real reference sets and synthetic-data combination assignments at each training stage.

C Additional Valuation Analysis

This section provides additional analysis of training-value estimation, the variation penalty, and changes in training value during training.

C.1 Training-Value Estimation Analysis

We compare training-value estimates with loss reductions on real references obtained after parameter updates at a fixed DiffusionDrive training state. For the comparison that does not account for the optimizer state, we replace F_t with a simple gradient step using the current learning rate η_t:

$$F_t^{\mathrm{raw}}(g) = -\eta_t g, \qquad \Delta^{\mathrm{raw}}(t) = -\eta_t g_{\mathrm{sh}}^s(t), \qquad d_{i,c}^{\mathrm{raw}}(t) = -\eta_t g_{i,c}^s(t). \tag{24}$$

The corresponding first- and second-order training-value estimates are

$$V_{i,b,c}^{\mathrm{raw},(1)}(t) = -\left[g_b^r(t)\right]^\top d_{i,c}^{\mathrm{raw}}(t) = \eta_t \left[g_b^r(t)\right]^\top g_{i,c}^s(t), \tag{25}$$

and

$$V_{i,b,c}^{\mathrm{raw},(2)}(t) = V_{i,b,c}^{\mathrm{raw},(1)}(t) - \frac{1}{2}\left[d_{i,c}^{\mathrm{raw}}(t)\right]^\top H_b^r(t)\Delta^{\mathrm{raw}}(t). \tag{26}$$

The estimates that account for the current optimizer state use Equations 3 and 5. All configurations are evaluated at the same model state and use the same gradients, real references, and synthetic-data combinations.

To obtain the loss reduction used for comparison on the disjoint real reference set, let $\widetilde{g}_i^s(t)$ denote the gradient contribution of $\mathcal{G}_i$ computed from a fixed set of its samples. The same normalization used for the synthetic-data combinations is applied to $\widetilde{g}_i^s(t)$. Using the same model parameters and optimizer state, we define

$$\theta^{\mathrm{ref}} = \theta_t + \Delta(t), \qquad \theta_i^+ = \theta_t + \left[F_t(g_{\mathrm{sh}}^s(t) + \widetilde{g}_i^s(t)) - F_t(0)\right]. \tag{27}$$

For real reference sequence h in the disjoint real reference set, the loss reduction attributable to $\mathcal{G}_i$ is

$$\Delta L_{i,h}^{\mathrm{obs}} = L_h^r(\theta^{\mathrm{ref}}) - L_h^r(\theta_i^+), \qquad \Delta L_i^{\mathrm{obs}} = \sum_h a_h \Delta L_{i,h}^{\mathrm{obs}}, \tag{28}$$

where a_h denotes the normalized weight of real reference sequence h. A positive value indicates that including the gradient contribution from $\mathcal{G}_i$ produces a larger reduction in loss on the real reference data than the shared training update alone.

Within each synthetic-data combination, we compute the Spearman correlation between the training-value estimates and $\Delta L_i^{\mathrm{obs}}$ for the synthetic data in that combination and then average the correlations. As reported in Table 7, accounting for the current optimizer state increases the average correlation from -0.1541 to 0.5118 for the first-order estimate and from 0.0341 to 0.7119 for the second-order estimate. After averaging repeated second-order valuations for each $\mathcal{G}_i$, the correlation across all synthetic-data groups reaches 0.7943.

C.2 Variation-Penalty Analysis

We compare ranking synthetic data by the adjusted training value $\widehat{V}_i(t)$ with ranking them by the mean training value $\bar{V}_i(t)$ alone. This analysis uses the disjoint real reference set reserved for valuation analysis and $\lambda = 1$, consistent with the main experiments.

Applying the variation penalty increases the Spearman correlation with the ranking based on $\Delta L_i^{\mathrm{obs}}$ by 0.1426 and increases the average loss reduction of the selected Top-K data on the disjoint real reference set by 0.0061. The corresponding 95% bootstrap intervals are $[0.0770, 0.2500]$ and $[0.0026, 0.0109]$, respectively.

We additionally examine whether larger variation across real references and synthetic-data combinations is associated with larger errors in the ranking obtained from mean training value alone. Define the normalized rank error as

$$e_i = \frac{\left|\mathrm{rank}_{\bar{V}}(i) - \mathrm{rank}_{\Delta L^{\mathrm{obs}}}(i)\right|}{N-1}. \tag{29}$$

Across synthetic-data groups, the variation term in Equation 9 has a Spearman correlation of 0.599 with e_i, with a 95% bootstrap interval of $[0.2808, 0.7362]$. Thus, synthetic data with larger variation across real references and synthetic-data combinations also tend to have larger rank differences between their mean training values and the loss reductions on the disjoint real reference data.

Table 7: Additional training-value and variation-penalty analysis on DiffusionDrive. Blocks A and B compare training-value estimates with loss reductions on real references after parameter updates. Block C evaluates the variation penalty on disjoint real reference data. Intervals are 95% percentile-bootstrap intervals.

Configuration	Statistic	Value	95% interval
A. Training-value estimation within synthetic-data combinations			
Without optimizer state, first-order	Mean Spearman ρ	-0.1541	–
Without optimizer state, second-order	Mean Spearman ρ	0.0341	–
With optimizer state, first-order	Mean Spearman ρ	0.5118	–
With optimizer state, second-order	Mean Spearman ρ	$\mathbf{0.7119}$	–
B. Training-value estimation after averaging repeated valuations			
With optimizer state, second-order	Spearman ρ	0.7943	–
C. Variation penalty on disjoint real reference data			
Rank agreement	Increase in Spearman ρ	0.1426	$[0.0770, 0.2500]$
Selected Top-K data	Increase in loss reduction	0.0061	$[0.0026, 0.0109]$
Variation vs. mean-value rank difference	Spearman ρ	0.599	$[0.2808, 0.7362]$

C.3 Statistical Intervals

The confidence intervals in Table 7 are computed with 20,000 bootstrap samples. For the comparison between the variation-penalized ranking and the ranking of synthetic-data groups by mean training value, we resample the eight real reference sequences in the disjoint real reference set with replacement.

Let $\Delta L^{\mathrm{obs}}_{i,h}$ be the loss reduction of synthetic-data group $\mathcal{G}_i$ on real reference sequence h, with normalized weight a_h. For each bootstrap sample,

$$(m_1, \ldots, m_8) \sim \mathrm{Multinomial}\left(8; \frac{1}{8}, \ldots, \frac{1}{8}\right), \qquad a^*_h = \frac{m_h a_h}{\sum_j m_j a_j}, \tag{30}$$

and

$$\Delta L^{\mathrm{obs},*}_i = \sum_h a^*_h \Delta L^{\mathrm{obs}}_{i,h}. \tag{31}$$

The same bootstrap sample is applied to all synthetic-data groups and to both rankings. Let $T_{\widehat{V}}$ denote the Top-K data selected by $\widehat{V}_i(t)$ and let $T_{\bar{V}}$ denote the Top-K data selected by $\bar{V}_i(t)$. These sets remain fixed across bootstrap samples. For each sample, we compute

$$\Delta\rho^* = \rho\left(\widehat{V}, \Delta L^{\mathrm{obs},*}\right) - \rho\left(\bar{V}, \Delta L^{\mathrm{obs},*}\right), \tag{32}$$

and

$$\Delta L^*_K = \frac{1}{K}\sum_{i \in T_{\widehat{V}}} \Delta L^{\mathrm{obs},*}_i - \frac{1}{K}\sum_{i \in T_{\bar{V}}} \Delta L^{\mathrm{obs},*}_i. \tag{33}$$

The 2.5th and 97.5th percentiles of the resulting distributions give the reported 95% intervals.

For the correlation between variation and normalized rank error, we resample synthetic-data groups with replacement and recompute the Spearman correlation between the variation term in Equation 9 and e_i for every bootstrap sample. The same percentile rule is used to obtain its 95% interval.

C.4 Changes in Training Value During Training

Figure 2 compares each DriveVal re-evaluation during DiffusionDrive training with the preceding one. The first point compares the initial ranking of synthetic-data groups by training value and selected data with those obtained at the first re-evaluation. Later points compare each re-evaluation with the preceding stage.

For the rankings of synthetic-data groups by training value, we compute the Spearman correlation between the current and preceding rankings. For the selected data, we compute the fraction of Top-K groups that changes from the preceding stage:

$$\frac{K - |\mathcal{S}_t \cap \mathcal{S}_{t-1}|}{K}. \tag{34}$$

A value of zero indicates that the selected data are unchanged, while a larger value indicates that more selected data have changed.

D Sensitivity to the Variation-Penalty Coefficient

We analyze the sensitivity of DriveVal to the variation-penalty coefficient λ on DiffusionDrive with DriveArena. We vary λ while keeping all other training and valuation configurations unchanged. The main experiments use $\lambda = 1$.

Table 8: Sensitivity of final-checkpoint NAVSIM navtest PDMS to the variation-penalty coefficient λ on DiffusionDrive with DriveArena. The main experiments use $\lambda = 1$.

λ	0	0.5	1	1.5	2
PDMS	78.09	80.62	83.28	83.47	82.96

Without the variation penalty ($\lambda = 0$), DriveVal obtains a PDMS of 78.09. Increasing λ to the value used in the main experiments ($\lambda = 1$) raises PDMS to 83.28. For $\lambda \in [1, 2]$, PDMS ranges from 82.96 to 83.47, indicating that the improvement from the variation penalty is not tied to a single coefficient value.

E Additional Experimental Results

This section reports additional results across end-to-end driving models, planning metrics, generation sources, and visual conditions.

E.1 Additional Per-Model Results

Table 9 reports the per-model PDMS for the additional configurations used in the valuation analysis. The first- and second-order configurations account for the current optimizer state and use the corresponding training-value estimates in Section 4.1. The configuration without periodic re-evaluation additionally applies the variation penalty but keeps the selected data unchanged throughout training.

Table 9: Additional per-model NAVSIM navtest PDMS. The first- and second-order configurations use the corresponding training-value estimates while accounting for the current optimizer state. The configuration without periodic re-evaluation uses the second-order training value and variation penalty but keeps the selected data unchanged throughout training. Bold marks the highest value in each row.

Model	Random	First-order	Second-order	Without re-evaluation	DriveVal
DiffusionDrive	73.77	75.17	77.96	79.32	**83.28**
Latent TransFuser	71.42	73.65	74.14	77.57	**80.84**
MeanFuser	73.32	76.13	77.56	79.75	**83.94**
WoTE	75.70	76.74	80.25	80.76	**84.28**
DrivoR	76.66	78.81	81.23	83.33	**88.85**
Model average	74.17	76.10	78.23	80.15	**84.24**

Across all five models, the second-order configuration achieves higher PDMS than the first-order configuration, and applying the variation penalty without periodic re-evaluation further improves PDMS. Periodic re-evaluation provides an additional improvement on every model, increasing the model average from 80.15 without periodic re-evaluation to 84.24 with DriveVal.

E.2 Additional Planning Metrics

Table 10 provides the complete model-average planning metrics for the configurations considered in the main experiments and additional analyses, including Comfort.

Table 10: Equal-weight averages of final-checkpoint NAVSIM navtest planning metrics across the five models. Each metric is averaged independently; PDMS is averaged from the model-level scores and is not recomputed from the averaged submetrics. NC denotes no-at-fault collision, DAC drivable-area compliance, EP ego progress, and TTC time-to-collision score. All metrics are reported on a 0–100 scale, with higher values indicating better performance. Bold marks the highest value in each column, including ties at the reported precision.

Configuration	PDMS ↑	NC ↑	DAC ↑	EP ↑	TTC ↑	Comfort ↑
Random	74.17	95.21	84.46	70.15	88.41	99.98
First-order	76.10	95.57	86.18	71.85	89.18	99.98
Second-order	78.23	96.00	88.08	73.72	90.04	99.98
Without periodic re-evaluation	80.15	96.38	89.79	75.42	90.78	**99.99**
DriveVal	**84.24**	**96.93**	**94.62**	**78.86**	**91.64**	99.98
Full Synthetic Set	77.23	95.47	87.55	72.51	89.16	**99.99**

DriveVal obtains the highest model-average PDMS, NC, DAC, EP, and TTC among the compared configurations. Comfort remains between 99.98 and 99.99 across all configurations.

Table 11 provides the complete per-model planning metrics for the Full Synthetic Set.

Table 11: Per-model final-checkpoint NAVSIM navtest metrics for the Full Synthetic Set. All metrics are reported on a 0–100 scale. The final row reports the equal-weight average across the five models.

Model	PDMS ↑	NC ↑	DAC ↑	EP ↑	TTC ↑	Comfort ↑
DiffusionDrive	77.87	96.94	87.57	72.36	91.33	99.99
Latent TransFuser	75.46	94.69	86.18	71.07	87.94	99.98
MeanFuser	77.13	95.08	87.65	72.63	88.57	99.99
WoTE	76.40	94.95	86.97	71.96	88.33	99.99
DrivoR	79.28	95.68	89.40	74.54	89.62	99.99
Model average	77.23	95.47	87.55	72.51	89.16	99.99

E.3 Generation Sources and Visual Conditions

Table 12 reports the complete planning metrics for three generation sources and two visual conditions. The generation-source experiments use front-camera images from DriveArena, BridgeSim, and WorldEngine. For the visual-condition experiments, fog and Gaussian Noise are applied at severity 5 to the real front-camera images.

For the Fog and Gaussian Noise conditions, the three front-camera images are converted to RGB uint8 at their original resolution and processed using `imagecorruptions.corrupt` with severity 5. On normalized RGB intensities, Gaussian Noise uses zero-mean Gaussian noise with standard deviation 0.38. Fog uses a plasma-fractal field with amplitude 3 and decay parameter 1.4, followed by the implementation's brightness normalization and clipping. Image dimensions are preserved.

To make the stochastic image processing reproducible, we use a base seed of 42 and derive the seed for each image deterministically from the visual condition, severity, sample token, and camera identity using SHA-256. The three camera views are `CAM_L0`, `CAM_F0`, and `CAM_R0`.

Table 12: Complete final-checkpoint NAVSIM navtest planning metrics across three generation sources and two visual conditions. Each cell reports Random / DriveVal. Both methods use the same number of training samples and optimizer updates for each generation source or visual condition. S5 denotes severity 5. Bold marks the higher value in each pair; displayed ties are both bold.

	Random / DriveVal					
Source / condition	PDMS ↑	NC ↑	DAC ↑	EP ↑	TTC ↑	Comfort ↑
DiffusionDrive						
DriveArena	68.48/**79.96**	93.11/**95.77**	80.11/**91.08**	64.86/**76.92**	84.68/**88.01**	**99.98**/99.77
BridgeSim	64.70/**78.16**	92.33/**96.41**	76.82/**89.37**	61.61/**73.26**	82.92/**89.64**	**99.98**/99.86
WorldEngine	71.59/**82.24**	93.85/**98.04**	82.73/**91.17**	67.63/**76.78**	86.16/**92.80**	**99.98**/99.46
Fog (S5)	74.12/**83.71**	94.43/**97.99**	84.96/**92.93**	69.89/**77.13**	87.35/**93.82**	**99.98/99.98**
Gaussian Noise (S5)	66.87/**78.60**	92.79/**96.25**	78.70/**90.60**	63.48/**72.29**	83.91/**89.96**	99.98/**99.99**
Latent TransFuser						
DriveArena	69.05/**77.94**	93.27/**96.34**	80.57/**88.81**	65.37/**72.99**	84.99/**89.91**	99.98/**99.99**
BridgeSim	62.49/**74.98**	92.50/**96.13**	76.90/**86.70**	57.89/**69.40**	82.71/**88.65**	**99.99/99.99**
WorldEngine	72.19/**79.19**	94.00/**96.46**	83.26/**89.58**	68.19/**75.28**	86.43/**89.33**	99.98/**99.99**
Fog (S5)	69.27/**79.14**	94.20/**97.23**	80.96/**88.83**	64.56/**72.92**	85.90/**92.17**	**99.99**/99.98
Gaussian Noise (S5)	68.38/**75.91**	**94.85**/94.60	80.35/**88.10**	60.92/**71.50**	**88.36**/87.20	99.97/**99.99**
MeanFuser						
DriveArena	70.31/**79.43**	95.49/**96.34**	82.21/**90.11**	62.51/**75.09**	89.57/**89.83**	**100.00**/99.20
BridgeSim	65.06/**76.76**	92.39/**94.97**	77.15/**89.10**	61.94/**72.66**	83.06/**87.81**	**99.98**/99.77
WorldEngine	73.18/**80.73**	95.35/**97.11**	84.03/**90.79**	68.83/**76.27**	87.96/**90.51**	**100.00**/99.57
Fog (S5)	72.65/**81.73**	94.55/**97.75**	83.95/**90.47**	68.54/**75.86**	87.34/**93.35**	**100.00**/99.98
Gaussian Noise (S5)	67.96/**75.52**	**94.81**/94.44	80.65/**87.74**	59.99/**71.64**	**88.42**/86.49	99.98/**99.99**
WoTE						
DriveArena	70.78/**80.45**	95.31/**96.55**	82.48/**91.65**	63.71/**74.92**	89.20/**90.09**	99.97/**99.98**
BridgeSim	68.40/**76.20**	93.07/**96.87**	80.06/**87.53**	64.79/**68.47**	84.64/**91.78**	**99.98/99.98**
WorldEngine	73.86/**82.24**	94.38/**97.39**	84.73/**92.71**	69.66/**75.40**	87.21/**92.41**	99.98/**99.99**
Fog (S5)	75.13/**82.68**	95.27/**97.24**	85.37/**93.55**	72.20/**75.19**	87.98/**92.69**	**100.00**/99.98
Gaussian Noise (S5)	69.46/**77.92**	95.06/**97.39**	82.08/**90.93**	62.01/**67.52**	88.08/**92.61**	99.99/**100.00**
DrivoR						
DriveArena	70.40/**82.35**	93.59/**96.86**	81.68/**92.28**	66.55/**77.69**	85.69/**91.13**	99.98/**100.00**
BridgeSim	70.11/**79.64**	93.50/**96.28**	81.45/**90.78**	66.27/**75.00**	85.55/**89.45**	**99.98/99.98**
WorldEngine	72.56/**84.50**	94.11/**98.15**	83.58/**93.07**	68.52/**78.61**	86.60/**93.75**	99.98/**99.99**
Fog (S5)	73.36/**85.21**	94.30/**98.16**	84.27/**93.74**	69.23/**79.42**	87.02/**93.96**	99.98/**99.99**
Gaussian Noise (S5)	69.47/**78.62**	93.34/**96.83**	80.93/**87.99**	65.71/**74.35**	85.21/**90.78**	99.98/**99.99**

F Computational Cost

We report the total training and valuation cost of Random, the configuration without periodic re-evaluation, and DriveVal in Table 13. Random includes training cost only. The configuration without periodic re-evaluation performs valuation once before training, while DriveVal performs the initial valuation and periodically re-evaluates the synthetic data during training. All three configurations use the same number of optimizer updates within each model. The reported costs exclude data generation and final navtest evaluation.

Table 13: Total training and valuation compute on NVIDIA L40 GPUs. Each cell reports GPU-hours followed by the total compute relative to Random within the same model. All configurations use the same number of optimizer updates within each model.

	Total L40 GPU-hours ↓		
Model	Random	Without periodic re-evaluation	DriveVal
DiffusionDrive	35.2 (1.00×)	38.1 (1.08×)	99.8 (2.84×)
Latent TransFuser	75.9 (1.00×)	82.8 (1.09×)	196.2 (2.59×)
MeanFuser	104.3 (1.00×)	105.9 (1.01×)	135.3 (1.30×)
WoTE	30.1 (1.00×)	41.3 (1.37×)	103.3 (3.43×)
DrivoR	199.3 (1.00×)	217.7 (1.09×)	254.0 (1.27×)

Across the five models, the configuration without periodic re-evaluation requires 1.01–$1.37\times$ the compute of Random, while DriveVal requires 1.27–$3.43\times$. The additional computation of DriveVal comes from the periodic re-evaluation of synthetic data during training.