Test-Time Coverage: Test-Conditioned Data Curation for Deployment-Aware Learning

Nadine Chang[*]
NVIDIA
nadinec@nvidia.com

Maying Shen[*]
NVIDIA
mshen@nvidia.com

Shizhe Diao
NVIDIA
sdiao@nvidia.com

Jialiang Wang
NVIDIA
jialiangw@nvidia.com

Jingde Chen
NVIDIA
joshchen@nvidia.com

Thomas Breuel
NVIDIA
tbreueul@nvidia.com

Pavlo Molchanov
NVIDIA
pmolchanov@nvidia.com

Rafid Mahmood
NVIDIA & University of Ottawa
rmahmood@nvidia.com

Jose M. Alvarez
NVIDIA
josea@nvidia.com

[*]Equal contribution.

Abstract

Deployed AI systems are often trained from broad candidate data pools, necessitating data curation towards the deployment test distribution. However, standard data curation methods score training-side criteria rather than directly optimizing deployment match. We introduce **TTCov (Test-Time Coverage)**, a data-level test-conditioned curation method that uses test-side information before training instead of updating model weights at inference. TTCov decomposes deployment-conditioned curation into coverage and distribution. To represent coverage, it builds a task *Atlas*, a collection of LLM-based atomic propositions (LPs) describing deployment-relevant concepts, seeded from open task knowledge and expanded with unmatched LPs extracted from unlabeled deployment samples. To represent distribution, it instantiates the matched deployment LPs with their frequencies, yielding a *Knowledge Atlas* (K-Atlas) that operationalizes the deployment distribution as a curation target. TTCov then selects a budgeted training set whose deployment LPs distribution approximates this target. We apply TTCov towards autonomous driving (AD), keeping adaptation off the inference path while selecting data with greater deployment-relevant coverage, closer K-Atlas matching, and stronger downstream end-to-end driving performance than data-curation baselines, including seamless adaptability to novel domains via city-to-city expansion.

1 Introduction

Deployed AI systems often have access to broad candidate data pools for training the underlying model. For example, physical AI systems such as autonomous driving (AD) and robotics can collect large amounts of sensor data from production fleets, and large-scale benchmarks covering the breadth of user scenarios [21, 15, 6, 49, 32]. Because training on every collected example is computationally infeasible, data curation policies seek to mine the data pool for a training subset that optimizes downstream test performance [38, 33].

Data curation methods often score samples and subsets from the data pool on training-side criteria such as diversity, representativeness, uncertainty, redundancy, or utility [43, 12, 44]. These criteria can

Preprint.

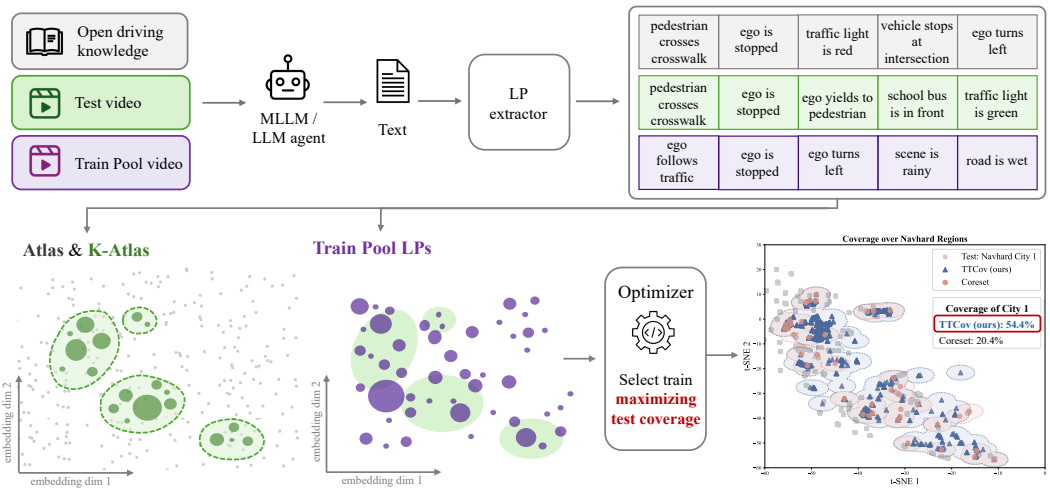


Figure 1: The Atlas consists of a collection of LLM-based atomic propositions (LPs) describing deployment-relevant concepts. [Top] It is seeded from open task knowledge (grey) and expanded with any new LPs extracted from test samples (green). [Bottom] To represent distribution, the frequency of test LPs is calculated, yielding K-Atlas. TTCov aims to curate a training dataset that contains relevant test LPs, matching the target test distribution and, consequently, maximizing test coverage, as shown in lower right plot.

produce compact and useful training subsets, but they do not directly ask whether the selected dataset matches the distribution the model will encounter after training. To mitigate the potential mismatch, unlabeled deployment-side information can be used to match the curated training distribution to the test distribution that the deployed system will encounter [8].

Statistical learning theory assumes a fixed test distribution and minimizes training error as a proxy for test error [52]. In contrast, training and testing sets for deployed AI systems are often constructed and evolved by hand with human-curated datasets that emphasize specific critical evaluation conditions [7]. However, human curation does not scale to the shifting deployment conditions for AI technology products. For these situations, transductive learning methods that condition on unlabeled deployment-side data have shown improved performance when test conditions differ from the original training distribution [25, 5, 58, 57, 4]. More recent test-time training and test-time adaptation techniques instantiate this premise by using test data online to update model weights at inference [50, 53, 34, 36, 23]. Although these methods have been shown to be effective in physical AI settings [45], online model adaptation in these applications multiplies inference latency and complicates model validation under real-time use conditions such as safety-critical human interaction [54, 36]. These methods also tune the model per test sample, which scales poorly with deployment volume.

In this work, we introduce **TTCov (Test-Time Coverage)**, a data-level test-conditioned curation method that adapts the training distribution before training rather than adapting model weights at inference (Fig. 1). TTCov breaks deployment-conditioned curation into two problems: (i) coverage, which asks what deployment-relevant content the training data should include, and (ii) distribution, which asks how frequently that content should appear. To represent coverage, TTCov builds an interpretable *task Atlas*: a collection of ***L**LM-based atomic **p**ropositions* (LPs) that describe deployment-relevant concepts such as scenarios, agents, conditions, and behaviors. The Atlas is seeded from open task knowledge and expanded with LPs extracted from unlabeled deployment-side samples when those samples contain concepts not already represented. To represent distribution, TTCov instantiates the matched deployment-side LPs with their frequencies, yielding the *Knowledge Atlas* (K-Atlas), an operational curation target for the deployment distribution. Our data curation pipeline then seeks to select a training dataset whose own K-Atlas, deployment relevant LP distribution, approximates the deployment K-Atlas.

We implement TTCov on autonomous driving, a safety-critical physical AI domain where the deployment distribution is available at training time via driving logs, target operational design domains, sensor stacks, and route specifications. Here, the Atlas is seeded from open driving knowledge such as taxonomies extracted from large language models (LLMs) [13, 2, 55], and it

expands when test-derived LPs do not match existing entries [6]. We demonstrate the effectiveness of TTCov in variable deployment conditions through city-to-city expansion, where the training set must be re-curated for a new city. Against the breadth of data curation techniques, TTCov selects data that covers more deployment-relevant LPs, better matches the target K-Atlas distribution, and improves end-to-end (E2E) autonomous driving performance. Our contributions are:

1. We propose TTCov, a data-level method for test-conditioned curation that adapts the training dataset to the unlabeled deployment distribution before model training.
2. We introduce the Atlas and K-Atlas as an interpretable coverage representation and frequency-weighted target distribution for deployment-conditioned data curation.
3. We formulate TTCov as budgeted K-Atlas matching and solve it with a greedy selection procedure over candidate training samples.
4. We provide a monotonic Atlas evolution mechanism for changing deployment conditions, instantiated through city-to-city domain expansion in autonomous driving.
5. We evaluate TTCov with coverage, distribution-matching, and downstream E2E autonomous driving metrics against data curation baselines.

2 Related Works

Data curation selects subsets of a candidate pool by training-side criteria. Coverage and diversity based methods aim to span the pool's feature or gradient space [43, 3, 12, 47]. Deduplication methods remove visually or semantically duplicate samples, often via CLIP-based similarity [1, 40, 27, 46, 42]. Uncertainty active learning selects samples on which the model is least confident [28, 26, 41]. Utility and difficulty based methods score samples by their effect on training or model behavior [39, 51, 44, 18], and scaling-aware analyses show that pruning can improve scaling-law behavior [48, 20]. All these criteria are defined entirely based on training data. They do not consider whether the selected subset matches the distribution the deployed model will encounter [8], which can cause selected data to be distributional misaligned with frequently shifting deployment distributions in safety-critical physical AI. Unlike these methods, TTCov performs test-conditioned curation method to ensure deployment distributional alignment.

Transductive Learning conditions on test-side information to improve predictions on specific test instances. Introduced in [52], specific transductive problem is argued to be easier to solve than the general inductive one. Classical methods include transductive SVMs using unlabeled test points to find low-density separators [25, 5], graph-based label propagation spreading labels through a similarity graph that has the test set [58, 57], and manifold regularization [4]. These methods assume a fixed labeled test set, work only on prediction stage, and do not scale to modern models or evolving deployment distributions. Modern works bring the transductive principle to model adaption: test-time training (TTT) adapts model weights at inference via a self-supervised auxiliary task [50, 34, 23]. Test-time adaptation (TTA) replaces the auxiliary task with a minimized objective in the test stream [53, 36, 54, 37, 56]. While modern model adaptation has shown effectiveness for E2E AD [45], their online updates present unscalable per-sample compute and latency.

End-to-end Autonomous Driving trains planning models that directly use raw sensor data to directly predict trajectories. Modern approaches use imitation learning to mimic expert ground-truth behavior [11, 10, 24, 9, 29, 30] and evaluate with both open and closed-loop metrics. However, open-loop metrics have been proven to be misaligned with true closed-loop driving quality [31, 16]. In an effort to translate to real-world driving, approaches now evaluate on closed-loops simulation benchmarks [17, 7], which we similarly evaluate in our work.

3 TTCov

In this section, we formulate TTCov, a framework for structuring knowledge relevant for deployment-conditioned data curation before model training. TTCov separates the curation problem into two stages: coverage, which asks what deployment-relevant knowledge should the training data include, and distribution matching, which asks how frequently should this knowledge appear. Below, we describe TTCov as instantiated on an AD task.

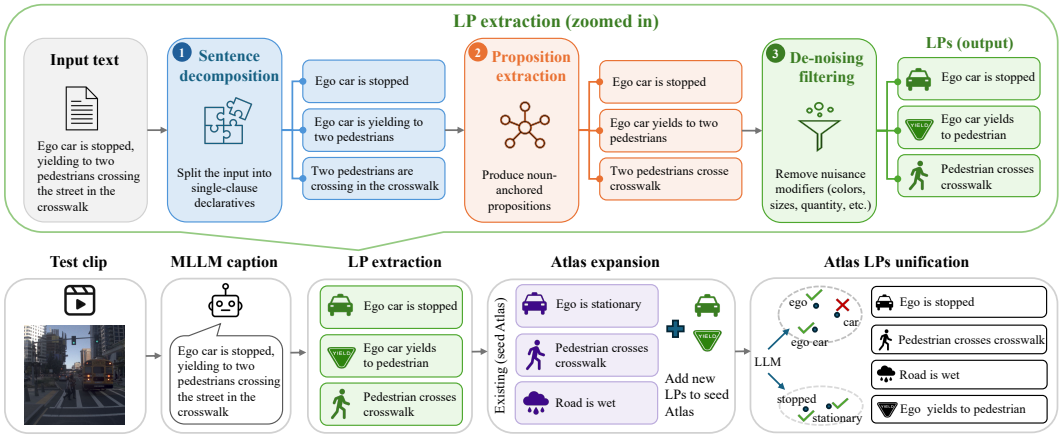


Figure 2: TTCov Overview. Top: The LP extraction process transforms complex descriptions into clean, atomic propositions and forms the list of LPs. Bottom: Workflow for generating the K-Atlas. Video test clips are captioned and converted into LPs, which are used to expand the seed Atlas. The final unification stage uses an LLM to merge similar phrases (e.g., "stationary" and "stopped"), ensuring a consistent phrasing.

3.1 Covering deployment knowledge via an *Atlas*

We define an Atlas as a set of LLM-based atomic propositions (LPs) that cover all knowledge required to complete the task. Each LP is a fundamental declarative sentence unit that describes concepts such as scenarios, agents, conditions, and behaviors (e.g., "A pedestrian crosses crosswalk", "A vehicle yields at an intersection", "The road is wet". Figure 2 summarizes the Atlas creation process.

For our AD setting, LPs are extracted from input plain text obtained from two knowledge sources. To seed an initial open-world knowledge, we note that LLMs trained on internet-scale data can generate strong structured ontologies for general tasks. We seed an initial Atlas by prompting an LLM to construct an ontology of driving conditions. We then augment this knowledge base with deployment test data. Here, an MLLM extractor translates each recorded driving video in the deployed test set into dense captions, answering a series of questions about the corresponding scenarios, agents, and behaviors. Our complete input knowledge base is the set of input texts from the initial seed Atlas and captions from the deployment data. We provide full prompt details for both extraction steps in Sec. B.

All input plain text is converted into LPs via a three stage process (see top of Fig. 2 for an example input text being parsed into LPs). First, we decompose the input text into single-clause declarative sentences. Then, we further simplify each declarative sentence by extracting the noun-anchored proposition (i.e., subject-verb-object or subject-attribute). Finally, each proposition may still contain non-essential modifiers or lexical noise. As a result, we apply de-noising filtering by removing all nuisance modifiers. This yields a list of LPs relevant to our task.

LPs obtained from multiple sources (e.g., seed Atlas, different driving videos) may have redundancy in equivalent concepts with slightly different phrasing. Consequently, we unify the list of LPs into our Atlas of unique concepts. We perform this unification in a two-stage approach. First, we cluster all the nouns and verbs using sentence embeddings. Each cluster then contains semantically similar phrases. Then for each cluster, we prompt an LLM to verify if all cluster phrases share the same meaning, identify any outlier phrases, and pick one phrase to replace all equivalent entries. This de-duplication process reveals the Atlas.

Finally, we define a LP distribution as the Knowledge-Atlas (K-Atlas). To instantiate the K-Atlas in AD, we review the set of original LPs from each video of the deployment test set. Within a given data sample, we perform pairwise matching for each LP against the set of LPs in the Atlas using Faiss [19]. Then, let $p^\star$ be the K-Atlas distribution vector, whose elements represent an importance frequency of each LP L via:

$$p^\star(L) := \frac{\#\{\text{test samples whose LPs include } L\}}{\sum_{L' \in \text{Atlas}} \#\{\text{test samples whose LPs include } L'\}}. \tag{1}$$

Algorithm 1: Greedy KL coverage for TTCov curation.

Input: Candidate pool $\mathcal{C}$, target K-Atlas $p^\star$, budget B, per-sample LP sets $\{\mathcal{L}(x)\}_{x\in\mathcal{C}}$
Output: Curated subset $S^\star$ with $|S^\star| = B$

$S \leftarrow \emptyset$
while $|S| < B$ **do**
  foreach $x \in \mathcal{C} \setminus S$ **do**
    $\hat{p}_{S\cup\{x\}} \leftarrow$ K-Atlas of $S \cup \{x\}$ over atlas LPs
    $g(x) \leftarrow D_{\mathrm{KL}}\left(p^\star \,\|\, \hat{p}_{S\cup\{x\}}\right)$
  end
  $x^\star \leftarrow \arg\min_{x\in\mathcal{C}\setminus S} g(x)$
  $S \leftarrow S \cup \{x^\star\}$
end
return $S^\star \leftarrow S$

3.2 Curating a training set by targeting the *K-Atlas* test distribution

Given the set of task-relevant knowledge, our goal is to curate a training dataset conditioned on a deployment LP distribution. Let $\mathcal{C}$ be a candidate data pool of samples, e.g. driving videos. For each sample $x \in \mathcal{C}$, we apply the same MLLM-based extraction procedure to obtain a corresponding set of LPs. To ensure that these LPs are unified in the same vocabulary as our K-Atlas, we perform pairwise matching of these LPs against the LPs in our Atlas.

We seek to construct a curated subset $\mathcal{S} \subset \mathcal{C}$ with a budget $|\mathcal{S}| = B$ samples. In TTCov, we optimize for a subset whose distribution matches the target K-Atlas. Specifically for a given subset $\mathcal{S}$, let $\hat{p}_{\mathcal{S}}$ be the distribution vector whose elements represent the importance frequency of each LP in the set $\mathcal{S}$, analogous to (1). Then, the goal fo TTCov is to optimize

$$\mathcal{S}^\star := \arg \min_{\mathcal{S}\subset\mathcal{C},\ |\mathcal{S}|=B} D_{\mathrm{KL}}(p^\star \,\|\, \hat{p}_{\mathcal{S}})\,. \tag{2}$$

The above problem (2) is a combinatorial optimization problem over a convex objective. Considering large candidate data pools increase the scale of the problem, we apply a greedy iterative algorithm by iteratively picking samples whose LPs maximally reduce the current KL gap. Algorithm 1 summarizes our approach.

4 TTCov Analysis

LLMs and MLLMs. Unless stated otherwise, we use Gemini 2.5 Pro [13] and Qwen 3 Embedding [55] for all relevant calls in Atlas and K-Atlas creation.

4.1 Atlas Ablations: Why LLM atomic propositions?

A natural starting point for our Atlas is a knowledge graph (KG), and recent work has shown that SOTA methods can generate KGs directly from large text corpus [35]. Starting from our seed Atlas, we use KG Gen to extract a knowledge graph. However, whole complex sentences collapse into a single triplet, losing shared atomic structure across examples. Consider two sentences: 1) *"The car is slowly nudging to the left and stopping because there are three people crossing Broadway street and a car is parked on the street"* contains four key parts (nudging left, stopping, three people crossing, car parked), while 2) *"The car is stopping because a person is crossing the avenue"* contains two (stopping, person crossing). These captions share key concepts — stopping, people crossing — but KG Gen treats them as entirely separate. We include a list of examples in Sec. B.

To capture shared similarity at the atomic level, we move away from KGs to atomic breakdowns of captions via LLMs. Each caption decomposes into basic atomic propositions (APs) across scenarios, actions, etc., e.g. *"Car turn left", "Person is crossing street", "Three people crossing street"*. More examples in Sec. B. However, modifiers like street names (*Broadway*), adjectives (*three*), synonyms (*avenue* vs. *street*), and plurals (*cars* vs. *car*) keep these APs separate. We note that none of these modifiers are critical for driving. We do not want to hit any number of people crossing any street, regardless of what color shirt they may be wearing. Thus, we add a final de-noising and phrasing

Table 1: TTCov's coverage metrics for multi-round city evolution. For each round, we add a new test city and report the number of selected training points, whose embedding is within 0.15 cosine dist of any test point (Num. Selected Train, NN @ 0.15), and Maximum Mean Discrepancy (MMD).

	Num. Selected Train, NN @ 0.15 (↑)				MMD(↓)			
Method	Round 1	Round 2	Round 3	Round 4	Round 1	Round 2	Round 3	Round 4
Coreset	42	36	19	62	0.13	0.10	0.20	0.13
SSE	46	49	17	68	0.13	0.11	0.20	0.13
TTCov (ours)	**732**	**355**	**47**	**228**	**0.08**	**0.06**	**0.14**	**0.10**

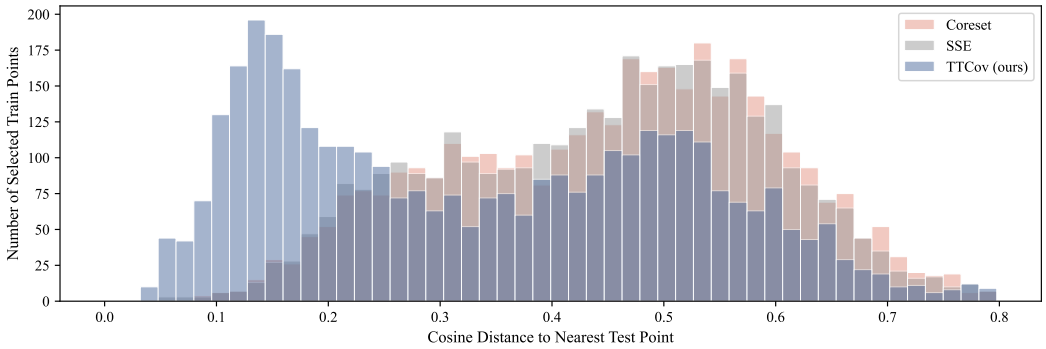


Figure 3: TTCov's selected data coverage over first test city. We show the histogram of the number of selected data points, whose embeddings are within a certain distance from any nearest test points. Given only one target test city, TTCov selects more data points closer to test points (see left of fig).

unification stage to remove distracting modifiers and unify near-identical APs. The final result is our Atlas: a compact, deduplicated representation of driving knowledge in atomic form.

4.2 Atlas Coverage

Datasets. We use OpenScene trainval [14] split as our training data pool, which provides a rich diversity of driving recordings. The dataset contains driving session clips lasting from 30 seconds to 50 minutes. To better capture the actions during driving and align with industry standards [18], we segment each session into fixed-length 10-second virtual clips (20 frames at 2Hz) with a sliding window. Adhering to the Navsim [17] framework configuration, each clip contains a 3-frame historical buffer and a 10-frame future horizon. The remaining frames are added to the training set upon virtual clip selection. Further details are available in Sec. E. We use Navhard [7] as the test dataset. Unlike standard test sets, Navhard isolates high-difficulty scenes, providing a more stringent metric for evaluating the planning stability and safety of the proposed model in edge-case environments. For our city expansion experiments, we iteratively add the cities in Navhard in the following order: Pittsburgh, Las Vegas, Singapore, and Boston.

Baselines. We compare TTCov against two baseline data selection strategies. Both methods aim to ensure that a model trained on the subset performs as closely as possible to a model trained on the entire pool. Coreset [43] selects data points from the pool that maximize diversity and coverage over the entire training feature space. SSE [44] clusters data points from the pool and maximize the diversity by removing semantically repetitive samples.

Atlas can adapt to dynamic test sets and maximize coverage. As operational task goals shift, test-conditioned curation must address dynamically evolving test sets. In AD, this manifests as a continuously growing set of corner cases existing datasets often fail to cover and is critical to identify. While some existing methods [47] attempt to address curation by matching train labels, they require discrete labels for tasks like classification. However, E2E planning in AD consists of continuous trajectories over a given time span. TTCov does not require such discrete labels. We evaluate TTCov ability to address this gap with a city-to-city domain adaptation setting, a common goal for AD.

TTCov addresses this evolving requirements by matching new test city data against its existing Atlas and updating the K-Atlas for the new city. Crucially, TTCov retains previously selected data from prior cities while adding only the most relevant samples for the new domain, ensuring efficient expansion without redundancy.

To evaluate data coverage efficacy, we use established coverage metrics, the Nearest Neighbor (NN) distance between closest two points in two datasets [23] and Maximum Mean Discrepancy (MMD) with a radial basis function. As shown in Tab. 1, TTCov's adaptation strategy maximizes test set coverage, yielding significantly more selected training data points close to existing test data points (NNs) and lower MMD than competing baselines. Notably, this advantage grows with domain evolution, where TTCov continues to gain coverage with each additional city. Fig. 3 visualizes the selected subset for the initial city, counting the amount of selecting data point within a threshold proximity of a nearest test point. Compared to other methods, TTCov selects a far denser concentration of points relevant to the test distribution. These metrics collectively highlight TTCov's flexibility, robustness, and ability for continuous adaptation across diverse domains. Additional results in Sec. B.

5 Experiments

We evaluate TTCov on Navhard, containing challenging scenarios from Nuplan, across various budgets and report E2E performance, EPDMS. Below details budgets, models, and metrics.

Budgets. We use the size of Navtrain, a manually curated dataset from OpenScene trainval split, as our baseline budget ($1\times$) for all studies. To evaluate the scalability of our approach, we further conduct a budget sweep across varying magnitudes, specifically at $0.5\times$, $0.75\times$, $1.25\times$ and $1.5\times$ the baseline budget B for our main results.

Models and Metrics. We evaluate data selection methods by training Latent Transfuser (LTF) [10], a baseline planner integrated into the NAVSIM framework [17, 7]. All models are trained for 100 epochs on eight NVIDIA A100 GPUs using the default NAVSIM training configuration. Trained checkpoints are evaluated on the NAVSIM-v2 benchmark [7], and we report EPDMS scores on the `navhard_two_stage` split. NAVSIM-v2 extends NAVSIM [17] with a two-stage pseudo-simulation protocol that augments real driving recordings with synthetic novel views to better approximate closed-loop evaluation. The `navhard` split is its most challenging curated evaluation subset, containing diverse and operationally difficult traffic and road geometries. For each data selection training split, we train models with three random seeds and the average EPDMS is reported, std. dev. in Sec. C.

Table 2: TTCov outperforms other curation baselines on Navhard. We report EPDMS and its ratio to the original manually curated dataset Navtrain (oracle), where 1 indicates parity to Navtrain.

	Budget				
	0.5x	0.75x	1x	1.25x	1.5x
Method	EPDMS(↑) | Rel.(↑)	EPDMS | Rel.	EPDMS | Rel.	EPDMS | Rel.	EPDMS | Rel.
Navtrain (oracle)	–	–	24.70 | 1.00	–	–
Random	18.95 | 0.77	18.76 | 0.76	20.15 | 0.82	21.95 | 0.89	22.95 | 0.93
Coreset	**20.77 | 0.84**	21.48 | 0.87	23.63 | 0.96	24.55 | 0.99	25.89 | 1.05
SSE	18.44 | 0.75	22.29 | 0.90	23.45 | 0.95	24.77 | 1.00	26.03 | 1.05
TTCov (ours)	20.62 | 0.83	**23.00 | 0.93**	**24.42 | 0.99**	**25.49 | 1.03**	**26.40 | 1.07**

5.1 Main Results

Navhard. To evaluate the efficacy of TTCov, we select subsets from an unlabeled training pool (OpenScene) and evaluate them on the E2E task in Navhard, a collection of the most challenging scenes from NAVSIM. Notably, the original Navtrain dataset was manually curated using ground-truth labels to filter out annotation errors and non-trivial solutions [17]. As such, we treat Navtrain as an oracle upper bound. In Tab. 2, we report both absolute EPDMS scores and the relative gains compared to the oracle across data budgets. For brevity, PDM submetrics are in Sec. C. TTCov demonstrates consistent performance gains over all baselines, including the SOTA method, SSE [44]. As the budget

Table 3: City to city performance. TTCov naturally extends Atlas to new cities without re-curation and improves EPDMS(↑) in both previous and current targeted cities.

	Round 1	Round 2		
Method	City 1	City 1	City 2	City 1+2
Coreset	15.72	20.29	29.69	22.80
SSE	15.78	19.91	31.18	22.18
TTCov (ours)	**20.23**	**21.94**	**38.11**	**26.39**

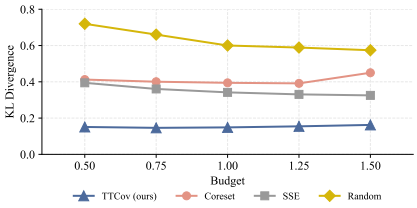


Figure 4: KL div to K-Atlas distribution.

Table 4: Distribution metrics at budget 1x.

Method	KL Div.(↓)	JS Dist.(↓)	Hell.(↓)	Cosine(↑)	Cov.(↑)
Random	0.60	0.31	0.32	0.78	0.67
Coreset	0.34	0.23	0.24	0.90	0.72
SSE	0.34	0.23	0.24	0.90	0.72
TTCov (ours)	**0.15**	**0.14**	**0.14**	**0.98**	**0.80**

approaches $B = 1$, TTCov approaches oracle performance more than other methods. Furthermore, at budgets exceeding the original Navtrain size, TTCov outperforms all baselines and the oracle itself. Crucially, at $B = 1$, TTCov automatically curates a dataset that achieves performance parity with Navtrain without requiring expensive human annotations or manual filtering.

TTCov for city evolution. Beyond the city evolution ablation study in Sec. 4.2, we train the E2E model on our selected data for each city added. Notably, TTCov proves truly adaptive; it only requires data selection for each new environment without re-selecting data for previous cities. As shown in Tab. 3, TTCov significantly outperforms all baselines for City 1 in the initial round. In Round 2, TTCov not only outperforms others in the newly introduced City 2 but also preserves its performance lead in City 1. This highlights TTCov's ability to continuously adapt to new domains without suffering from catastrophic forgetting. Ultimately, TTCov far outpaces competing methods across both cities, demonstrating its general robustness and stability in evolving environments.

TTCov selects data with closest distributional match to test. By targeting our K-Atlas test distribution comprised of relevant LPs, TTCov curates a training set through an optimizer that minimizes the distributional distance between the selected data and the target K-Atlas. This alignment is visualized in Fig. 4, which shows that TTCov's selected data distribution is significantly closer to K-Atlas than other methods. Crucially, our results confirm a correlation between distributional proximity and model performance. The closer the selected dataset aligns with the K-Atlas test distribution, the higher the resulting performance. For example, as shown in Tab. 2, TTCov achieves the highest EPDMS score alongside the lowest KL divergence, whereas Coreset and SSE exhibit both higher KL divergence and correspondingly poorer EPDMS performance. Lastly, this trend remains consistent across all several distribution metrics. As shown in Tab. 4, TTCov consistently achieves the closest distributional proximity regardless of the metric evaluated.

5.2 Ablations

Optimizer Ablations. We study the various effects of ablating different aspects of TTCov's optimizer. First, we compare our default approach against Greedy Residual Matching (GRM), which selects samples to minimize the weighted squared error between current LP coverage and the target $B \cdot p^{\star}$. However, in Tab. 5, GRM yields a dataset with a poorer distributional match to K-Atlas, resulting in lower EPDMS scores. Next, we ablate the parameters within greedy optimization in Tab. 5. We ablate the selection metric by replacing KL divergence with normalized dot product. This swap similarly leads to worse distributional alignment and EPDMS, validating KL divergence as the better objective. While greedy KL selection effectively minimizes global divergence, it can still select samples containing already overly covered, saturated LPs. So we add an over coverage penalty ρ that lowers a candidate's score by how much more it adds to saturated LPs, with $\rho = 1$ recovering our original score. We observe that both under and over penalizing lead to significantly higher KL

Table 5: Optimizer ablations with corresponding KL divergence to K-Atlas and downstream EPDMS.

Method	Metric	Reweighting	KL div.(↓)	EPDMS(↑)
Greedy Residual Matching	weighted square error	–	0.53	20.92
Greedy Metric Optimization	KL divergence	–	**0.15**	24.42
		RFS, $t = 0.001$	0.16	23.90
		RFS, $t = 0.01$	0.18	25.39
		RFS, $t = 0.1$	0.21	**26.15**
		penalty, $\rho = 0.5$	1.19	16.55
		penalty, $\rho = 2.0$	0.53	21.82
	Dot product norm	–	0.45	23.37

Figure 5: Visual samples of TTCov's selected data with diverse relevant LPs from our K-Atlas, demonstrating high-level, low-level, and safety-aware understanding.

divergence and worse EPDMS compared to our baseline, $\rho = 1$. Finally, since Navhard contains hard long-tail scenarios, we employ a standard long-tail strategy, repeat-factor-sampling (RFS) [22], to emphasize rare LPs. While RFS naturally increases the KL divergence relative to the unweighted K-Atlas distribution, it reproduces positive long-tail results by significantly improving EPDMS to 26.15, surpassing even the Navtrain oracle. We report our main results without such re-weighting strategies to keep the focus on the core TTCov framework, Atlas. However, these results suggest that further optimization techniques can push TTCov's performance even higher. For full details on all optimizer ablations, refer to Sec. D.

Atlas Ablations. We refer to Sec. 4 and Sec. B for Atlas construction ablations and further ablations.

Visualizations. In Fig. 5, we illustrate the semantic breadth and depth of TTCov's curation, which selects data whose LPs span across a variety of scenarios, agents, and behaviors. Not only do we capture high-level environmental features such as bridges, TTCov also identifies safety critical and usually rare agents like pedestrians and school buses. Finally, TTCov also captures non-nominal driving actions, ensuring the final dataset is both visually and behaviorally diverse. This demonstrates TTCov's ability beyond curation and towards safety-aware understanding of driving task.

6 Conclusion

We propose TTCov, a framework for test-conditioned curation that adapts training datasets to deployment distributions by leveraging an interpretable Atlas of LLM-based atomic propositions. Unlike traditional curation methods that rely on training-side proxies, TTCov directly optimizes for deployment coverage and distributional alignment. Our results demonstrate that this stronger distributional matching correlates with improved E2E AD performance. Furthermore, while outside of our primary focus, with further optimization improvements, TTCov can surpass even the human-curated oracle, Navtrain. Beyond performance, TTCov demonstrates its scalability and adaptability to novel conditions, common failure points in AD. Through city-to-city experiments, we illustrate the framework's unique ability to adapt to new conditions by curating data for new cities without requiring complete re-curation or suffering from catastrophic forgetting. Lastly, the flexibility of TTCov and

its underlying interpretable Atlas demonstrate that the framework can not only meaningfully curate training data, but also capture the nuanced, safety-critical aspects of the driving task.

7 Limitations

While TTCov presents a powerful framework for extracting and leveraging test-conditioned knowledge for data curation, we note two limitations. TTCov efficacy is purposely tied to an existing test set. Thus, TTCov cannot capture information for extremely rare cases that do not exist in the test set nor in open-world knowledge. Similarly, open-world knowledge is extracted via LLMs, limiting TTCov's ceiling to their knowledge. However, LLMs continue to make enormous progress, and TTCov's performance will continue to scale alongside these models.

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.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[4] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[5] Kristin P. Bennett and Ayhan Demiriz. Semi-supervised support vector machines. In *NeurIPS*, 1998.

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020.

[7] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

[8] Nadine Chang, Maying Shen, Jialiang Wang, Rafid Mahmood, and Jose M. Alvarez. Position: Stop reactively patching your model every time and start proactive test-driven ai development. In *ICML*, 2026.

[9] 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.

[10] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 45(11):12878–12895, 2023.

[11] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *ICRA*, pages 4693–4700. IEEE, 2018.

[12] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *ICLR*, 2020.

[13] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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

[15] Michael J. Coren. Tesla has 780 million miles of driving data, and adds another million every 10 hours. Quartz, 2016. Published May 28, 2016. Accessed: 2026-05-06.

[16] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *CoRL*, 2023.

[17] 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 *NeurIPS*, volume 37, 2024.

[18] Tolga Dimlioglu, Nadine Chang, Maying Shen, Rafid Mahmood, and Jose M. Alvarez. Scaling-aware data selection for end-to-end autonomous driving systems. In *CVPR*, 2026. Accepted.

[19] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. *TBD*, pages 1–17, 2025.

[20] Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *CVPR*, pages 22702–22711, 2024.

[21] Adam Grzywaczewski. Training AI for self-driving vehicles: The challenge of scale. NVIDIA Technical Blog, Oct. 2017. Accessed: 2026-05-06.

[22] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019.

[23] Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *ICLR*, 2024.

[24] 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 *ICCV*, pages 8340–8350, October 2023.

[25] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[26] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pages 2372–2379. IEEE, 2009.

[27] 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.

[28] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *ICML*, pages 148–156. Elsevier, 1994.

[29] 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.

[30] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Jingde Chen, Nadine Chang, Maying Shen, Jingyu Song, Zuxuan Wu, Shiyi Lan, et al. Ztrs: Zero-imitation end-to-end autonomous driving with trajectory scoring. *arXiv preprint arXiv:2510.24108*, 2025.

[31] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*, pages 14864–14873, 2024.

[32] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *TPAMI*, 45(3):3292–3310, 2023.

[33] 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. *TIV*, 9(11):7138–7164, 2024.

[34] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: When does self-supervised test-time training fail or thrive? In *NeurIPS*, volume 34, pages 21808–21820, 2021.

[35] Belinda Mo, Kyssen Yu, Joshua Kazdan, Proud Mpala, Lisa Yu, Charilaos I. Kanatsoulis, and Sanmi Koyejo. KGGen: Extracting knowledge graphs from plain text with language models. In *NeurIPS*, 2025.

[36] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, volume 162, pages 16888–16905, 2022.

[37] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *ICLR*, 2023.

[38] 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.

[39] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *NeurIPS*, volume 34, pages 20596–20607, 2021.

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[41] 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. *CSUR*, 54(9):180:1–180:40, 2021.

[42] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, volume 35, pages 25278–25294, 2022.

[43] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

[44] Maying Shen, Nadine Chang, Sifei Liu, and Jose M Alvarez. Sse: Multimodal semantic data selection and enrichment for industrial-scale data assimilation. In *KDD*, pages 2525–2535, 2025.

[45] Chonghao Sima, Kashyap Chitta, Zhiding Yu, Shiyi Lan, Ping Luo, Andreas Geiger, Hongyang Li, and Jose M Alvarez. Centaur: Robust end-to-end autonomous driving with test-time training. *arXiv preprint arXiv:2503.11650*, 2025.

[46] Eric Slyman, Stefan Lee, Scott Cohen, and Kushal Kafle. Fairdedup: Detecting and mitigating vision-language fairness disparities in semantic dataset deduplication. In *CVPR*, pages 13905–13916, 2024.

[47] Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented bayesian active learning. In *AISTATS*, pages 7331–7348, 2023.

[48] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *NeurIPS*, volume 35, pages 19523–19536, 2022.

[49] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020.

[50] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, pages 9229–9248. PMLR, 2020.

[51] 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 *ICLR*, 2019.

[52] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[53] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

[54] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pages 7201–7211, 2022.

[55] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[56] Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *NeurIPS*, 2022.

[57] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NeurIPS*, 2004.

[58] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, 2003.

A Broader Impact

TTCov improves autonomous driving safety by enabling interpretable, deployment-conditioned data curation. By guiding selection in an explicit Atlas of atomic propositions, TTCov enables adaptability to novel environments with potentially new edge-cases and reduces the computational cost for learning to new domains. However, we note that real-world deployment still requires rigorous closed-loop testing and oversight.

B Atlas Ablations

B.1 Atlas threshold for LP matching

To additionally measure the TTCov's robustness to LLM sensitivities, we ablate the cosine similarity threshold used by Qwen 3 in LP matching to cluster similar LPs. Within each cluster, an LLM verifies all phrases are semantically similar, finds any outliers, and merges all phrases into a single phrase. As seen in Tab. 6, TTCov is robust to this threshold and EPDMS scores remain consistent across all thresholds.

Table 6: We ablate the threshold used by Qwen 3 in LP matching to cluster similar LPs. Within each cluster, an LLM (Gemini 2.5 Pro) performs all final matches and merging. Due to compute constraints, reported numbers are from one run.

Threshold	EPDMS
0.83	24.12
0.84	24.44
0.85 (main)	24.42
0.86	24.54
0.87	24.36

B.2 Why LLM atomic propositions? Continued.

In Fig. 6, we illustrate some complex triplets in a knowledge graph extracted from a large text corpus. We observe that each triplet contains phrases that are difficult to parse and contain a multitude of information. Furthermore, triplets often end with "None". Leveraging such a complex knowledge graph representation makes it difficult to find key commonalities among test data. Next, we provide additional samples of unnecessary modifiers that are irrelevant for our task in Fig. 7. Modifiers include basic articles, colors, and clothing. In our final de-noising and unification stage LP extraction, we merge semantically similar phrases, if not for these modifiers.

Ego, approaches an intersection from the minor road that are governed by the stop signs, while the cross-traffic directions are uncontrolled, None

Ego, maintains or re-establishes a safe longitudinal gap to a vehicle that is crossing ego's path while ego is making an unprotected left turn vehicles, None

Ego, manages its motion to avoid stopping or queuing on a designated keep-clear area, keep-clear area

Ego, starts accelerating from standstill after stopping or contender in front increased its distance far enough that it is no longer relevant for ego's immediate driving decisions, None

Ego, uses, adjacent buffer space (e.g., bike lane, parking lane, bus lane, shoulder)

Ego, maintains or re-establishes a safe longitudinal gap to a vehicle on the crossroad making a right-turn while ego is making a U-turn, vehicles

Ego, maintains a safe/appropriate gap and yields to a VRU that is in, entering, or clearly intending to enter an unmarked crosswalk, None

Driver, must signal and use the left turn lane or far-left lane when making a U-turn, None

Vehicle, slow down and be ready to stop to let any vehicle, bicyclist, or pedestrian pass before you proceed, RED YIELD SIGN

Driver, must travel at a reasonable speed and not endanger the safety of, other bicyclists

BROKEN WHITE LINES, separates, traffic lanes on roads with two or more lanes in the same direction

Vehicle, yield to all traffic already in the roundabout, None

Ego, approaches, intersection where all contenders are required to come to a complete stop

Figure 6: Samples of complex triplets in knowledge graph.

"a man": "man",
"a minivan": "minivan",
"a motorcycle": "motorcycle",
"a parked utility vehicle": "parked utility vehicle",
"a person": "person",
"a pickup truck": "pickup truck",
"small red bus": "bus",
"small red car": "car",
"small red three-wheeled vehicles":
"three-wheeled vehicles",
"small shuttle bus": "shuttle bus"
"small shuttle vehicle": "shuttle vehicle",
"small silver car": "car",
"another gray sedan": "sedan",
"another gray suv": "suv",
"car with an orange roof": "car",
"car with blue and red wrap": "car",
"car with camouflage-patterned wrap": "car",
"car with colorful advertising wrap": "car"
"one person in orange top": "person",
"one person in uniform": "person",
"one person in white top": "person",

Figure 7: Samples of unnecessary modifiers, which we removed in our final de-noising phrase unification stage.

B.3 Structured ontology extracted from LLM

We prompt Gemini 2.5 Pro, trained on internet-scale data, to generate strong ontologies for autonomous driving. We illustrate a subsample of decomposed LPs prior to de-noising and unification from LLM generated ontologies in Fig. 8. Critically, we observe the ontology includes diverse ego actions as well as important driving events to note (e.g. "Ego waits for the jaywalker to finish crossing." and "Giant puddle might be a deep pothole.")

Ego has to stop for jaywalker. Ego yields to car running a red light. Ego stays clear of driver going the wrong way. Ego ignores aggressive honking from behind. Ego expects car in turn-only lane to go straight. **Ego slows down for car drifting near the line. Ego waits for pedestrian making eye contact.** Ego should watch for car doors opening in bike lanes. Ego anticipates kids running after a stray ball. Ego gives space to car looking for a parking spot. Ego doesn't enter blocked intersection. Ego creeps out to unblock its own view. Ego should expect hidden cars at a blind driveway. **Ego slows down at puddle to avoid splashing people. Ego stops behind the crosswalk line.** Ego pulls over for ambulance with sirens. Ego avoids getting sandwiched between two semis. Ego stays out of another car's blind spot. Ego covers the brake when passing a row of parked cars. Ego matches speed with highway traffic while merging. Ego treats broken traffic light as a four-way stop. **Ego switches lanes to avoid a stalled car. Ego keeps distance from truck carrying loose gravel.** Ego waits an extra second at fresh green light. Ego needs a bigger gap if road is wet. Ego feels squeezed by two big semi-trucks. Ego should stay back from car with a shaky load. Ego needs to watch out for tailgaters. Ego gives extra room to student driver. Ego creeps forward to see around a corner. Ego can't trust a green light it just saw change. Ego checks its blind spot before moving over. Ego stays out of the truck's no-zone. **Ego double-checks for bikes before turning right. Ego takes its turn at the four-way stop. Ego waits for the jaywalker to finish crossing.** Ego slows down if the car next to it hits the brakes. Ego lets the person merging into the lane. Ego doesn't just go because the guy behind is honking. Ego dodges the deep pothole. Ego steers wide around the person changing a tire. **Ego stops early for the bus with its sign out. Ego nudges over to get past the double-parked car.** Ego is ready to slam the brakes for the ball in the street. Ego picks the lane that moves the fastest. **Ego gets out of the way of the ambulance. Ego follows the detour signs. Ego matches speed with the rest of traffic.** Ego aims for the middle of the lane. Ego should never trust a wave-through without checking the lane itself. Ego treats a flashing red light exactly like a stop sign. Ego gives the merging zipper space during heavy traffic. Ego covers the brake when it sees a stale green light. Ego avoids blocking the box at an intersection even if the light is green. Ego stays back if it can't see a truck driver's mirrors. Ego should not pass a vehicle that is stopped at a crosswalk. Ego matches the speed of the flow of traffic even if it's slightly above the limit. Ego flashes hazard lights to warn cars behind of a sudden stop. **Ego performs unprotected left turn by judging gaps in oncoming traffic.** Ego needs to watch for bikes when opening its own door. **Ego should slow down when passing a stopped school bus.** Ego moves over a lane to give space to a car on the shoulder. Ego looks both ways even on a one-way street. Ego leaves space for people to pull out of driveways in heavy traffic. Ego avoids the suicide gap when a car waves it through a blind turn. Ego anticipates a lane change when the car ahead is hugging the line. Ego gives extra room to a car that is swerving or driving erratically. Ego should never pass a snowplow on the right. **Ego checks for pedestrians before making a right turn on red.** Ego treats a dark intersection as a four-way stop when power is out. Ego avoids high beams when following another car. Ego slows down before a speed bump to keep the ride smooth. Ego watches for animals near the edge of the woods at dusk. Ego doesn't speed up when another car is trying to pass it. Ego gives a thank you wave when someone lets it into traffic. Ego stays out of the intersection if it can't clear the other side. Ego assumes a parked car with exhaust coming out might move soon. Ego gives extra space to delivery bikes in the city. Ego watches for shopping carts rolling away in a parking lot. Ego expects pedestrians to be wearing dark clothes at night. **Ego slows down in a parking garage because of tight corners.** Ego keeps its wheels straight while waiting to turn left. Ego treats unmarked parking lot aisles as two-way streets. Ego yields to any car already backing out of a spot in a busy lot. Ego pulls into a driveway to let an oncoming car pass in a narrow alley. Ego assumes pedestrians in a parking lot will walk behind the car without looking. Ego avoids passing a garbage truck on a narrow street because a worker might step out. Ego follows the tire tracks of the car ahead when lanes are snowy. Ego increases following distance x3 when driving on gravel or unpaved roads. Ego watches for door-prize scenarios when driving past outdoor dining setups. **Ego yields to the car traveling uphill on a narrow mountain road.** Ego assumes a delivery bike will travel against the flow of traffic in a one-way alley. Ego turns off high beams when entering a residential dead end. Big truck hides the view of traffic light. **Row of parked cars could hide kid or dog.** Sun glare makes it hard to see red light. Foggy windows make it tough to change lanes. Blinker shows where car wants to go. Speeding car is likely to cut someone off. Driver looking at phone is probably going to drift out of lane. **Door opening on a parked car means someone is stepping out.** Double-parked delivery van forces you to nudge into oncoming traffic. **Giant puddle might be a deep pothole. Construction workers have the final say over traffic lights.** Ball rolling into street is usually followed by a kid. Trash day means garbage trucks making frequent stops. Black ice makes the road slick as a skating rink. School zone means everyone needs to crawl. Heavy rain means double the stopping distance. Parking lot is full of people backing out without looking. Emergency siren means pull over to the right immediately. Ego treats shared center turn lane as a temporary space for both directions. Suicide gap is a gap in stopped traffic that invites a dangerous turn. **Ghost pedestrian space is the area between parked vans where someone might pop out.** Stale green light is a light likely to turn yellow soon. Blind spot is a place where cars disappear. **Brake lights mean the car ahead is slowing down.** Zip-merging means taking turns like a zipper. Flashing high beams is a way to let someone go first. Limit Line is the stripe that tells the Ego where its nose should stop. Door-prize is a slang term for a car door opening into the Ego's path.

Figure 8: Samples of decomposed LPs prior to de-noising and unification from LLM generated ontologies.

B.4 Additional coverage analysis

We visualize TTCov's selected data coverage for the second test city in Fig. 9. Consistent with the patterns observed for City 1, TTCov selects a higher volume of training points in close proximity to the test distribution than baseline methods. Furthermore, Fig. 10 illustrates the ground-truth geographic distribution of the data selected by TTCov. The results show that TTCov targets the specific test city significantly better by selecting more than $3\times$ the data for City 1 and $2\times$ for City 2 compared to other methods. Additionally, we show more quantitative coverage metrics in Tab. 7, where we observe TTCov selects high volume of training points even as we increase the distance to nearest test data. We also report an additional metric, mean NN distance, an average of all distances for each train point to nearest test point. Again, TTCov selects a tighter set of points around test regions. These results further validate TTCov's ability to maximize relevant data coverage even in dynamically changing test domains.

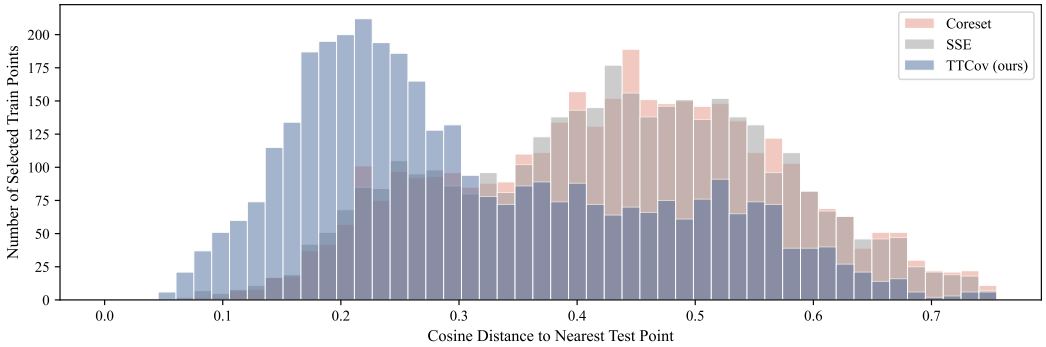
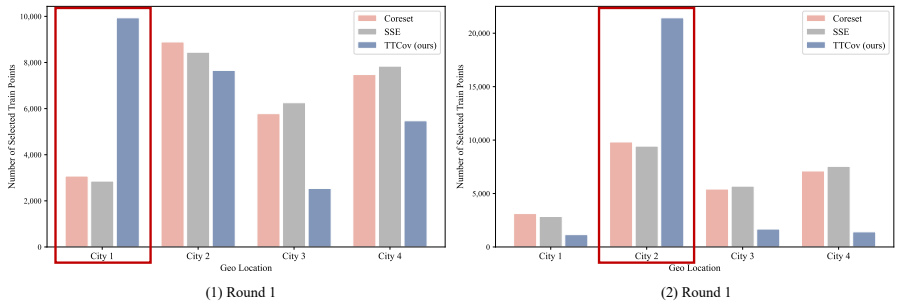


Figure 9: TTCov's selected data coverage over the 2nd test city. We show the histogram of the number of selected data points, whose embeddings are within a certain distance from any nearest test points. Given only one target test city, TTCov selects more data points closer to test points (see left of fig).



(1) Round 1 (2) Round 1

Figure 10: The geography distribution of selected data during the first and second round of city evolution. In each round, TTCov's selected data contains more data points that are from the target test city.

Table 7: TTCov's coverage metrics for multi-round city evolution.

Round	Method	Num. Selected Train(↑)			Mean NN Dist.(↓)	MMD(↓)
		NN@0.15	NN@0.30	NN@0.45		
Round 1	Coreset	42	659	1646	0.20	0.13
	SSE	46	699	1697	0.19	0.13
	TTCov (ours)	**732**	**1744**	**2436**	**0.16**	**0.08**
Round 2	Coreset	36	733	1950	0.19	0.10
	SSE	49	774	1991	0.18	0.11
	TTCov (ours)	**355**	**2083**	**2877**	**0.16**	**0.06**
Round 3	Coreset	19	357	849	0.25	0.20
	SSE	17	362	856	0.25	0.20
	TTCov (ours)	**47**	**814**	**1651**	**0.23**	**0.14**
Round 4	Coreset	62	785	2019	0.19	0.13
	SSE	68	837	2096	0.19	0.13
	TTCov (ours)	**228**	**1341**	**2532**	**0.17**	**0.10**

B.5 Prompts

We prompt Gemini 2.5 Pro for video knowledge extraction, LPs extraction, LPs de-noising and LPs unification. We provide the prompts in Fig. 11.

Video Captioning:

```
You are a driving instructor. Your student is the driver of this video. The video is presented by the provided sequence of images. The video might include discontinuities, sudden changes in the driving environment.
Please pay attention to what action your student is taking and explain why.  Please pay special attention to if the car is reacting to a traffic light intersection, or making a lane change, or keeping distance from the pedestrian/vehicle ahead, or avoiding some objects/cars on the road, or yielding or nudging to other objects/cars, or at a fork split/merge.  If the car is at a traffic light intersection, please describe the traffic light color and shape and the car's action. Describe it if objects are partially occluded by others.

Exclude the following in your output:
- UNRELATED objects
- Background buildings
- Hypothetical scenarios or actions
- Street/business names
- Vehicle makes/models

Use 'the ego car' to refer to your student in your answer.
Make sure your answers have considered everything throughout the entirety of the video.
Do not mention anything that you are certain does not exist (no 'there is/are no')! No statements about uncertain objects or events (no 'maybe' or 'might' or 'possibly').
All responses must be in English only!

Format your answers into Overall Scene Description, Key Driving Actions, Relevant Road Users and Objects, Overall Assessment.
```

(a) Prompt for video knowledge extraction.

Sentence Decomposition:

```
Role: You are an expert in English syntax, grammar, and Knowledge Graph ontology. Your task is to systematically decompose complex paragraphs into pure, atomic Subject-Verb-Object (SVO) triplets without losing any original information.

You must strictly adhere to the following rules to ensure the output is optimized for Knowledge Graph generation (Node-Edge-Node):

1. Atomic SVO/Triplet Structure: Break down every thought into basic, un-nested Subject-Verb-Object sentences. There should be no complex clauses within the subject or object.
2. Distribute Conjoined Elements (Cross-Multiplication): If a subject or action applies to a list of items (e.g., A does X, Y, or Z to target 1 or 2), distribute the relationship across all items. Explicitly write out every single combination as a standalone SVO triplet (e.g., A does X to 1; A does X to 2; A does Y to 1; etc.).
3. Chain Relationships: Flatten sentences into a logical chain of relationships linking high-level categories to actors, actors to actions, actions to targets, and targets to environments (e.g., A is B; B involves C; C does D; D is at E).
4. No Added Articles: Do not add any new articles (a, an, the) just to make sentences sound more grammatically correct. Rely strictly on the vocabulary provided in the source text.
5. Resolve Pronouns and Demonstratives: Replace all pronouns (it, they, them) and demonstrative determiners (this, that, these, those) with the actual, specific concepts or nouns they refer to.
6. Resolve Locative Adverbs: Replace spatial adverbs (like "there" or "here") with the specific object or location they reference in the context of the text.
7. Keep Negatives: Retain all negative sentences, clauses, and conditions. Do not invert them or delete them.
8. Decompose Complex Noun Phrases: If a complete subject or complete object contains embedded clauses or heavy modifiers (e.g., "people using wheelchairs"), extract that phrase and create a new, separate SVO sentence to define it using its simple subject (e.g., "people use wheelchairs").
9. Group by Source: Group the resulting basic sentences under the exact original sentence they were extracted from.
10. JSON Output: Provide your final response exclusively in JSON format. Do not include conversational filler before or after the JSON.

Use the following JSON structure:
{
  "paragraph_breakdown": [
    {
      "original_sentence": "[Insert the first original sentence here]",
      "basic_sentences": [
        "[Atomic SVO triplet 1]",
        "[Atomic SVO triplet 2]",
        "[Atomic SVO triplet 3]"
      ]
    }
  ]
}

Input Text:
```

(b) Prompt for decomposing raw knowledge to atomic propositions, LPs extraction.

LPs Denoising:

```
You are cleaning entity names from a knowledge graph about autonomous driving scenes.

Given a JSON list of entity strings, return a JSON object mapping original entities to cleaned entities, but ONLY for entities that are actually modified.

Rules:
1. Remove color, size, and shape descriptors from vehicle entities (car, truck, SUV, van, bus, motorcycle, sedan, pickup, vehicle, automobile, wagon, semi-truck, minivan, hatchback, coupe, jeep, trailer)
  - Examples: "white semi-truck" -> "semi-truck", "red SUV" -> "SUV", "large white truck" -> "truck", "small blue car" -> "car", "silver sedan" -> "sedan"
2. Remove brand/make names from vehicle entities (Toyota, Honda, BMW, Ford, Tesla, Chevrolet, Nissan, Audi, Mercedes, Hyundai, Kia, Volkswagen, etc.)
  - Examples: "toyota sedan" -> "sedan", "BMW car" -> "car", "honda vehicle" -> "vehicle"
3. Keep position and state descriptors for vehicle entities — do NOT remove these
  - Examples: "stationary car" stays "stationary car", "oncoming vehicle" stays "oncoming vehicle", "parked truck" stays "parked truck", "stopped bus" stays "stopped bus"
4. Keep colors in traffic light, traffic signal, and sign entities — do NOT remove these
  - Examples: "red traffic light" stays "red traffic light", "green signal" stays "green signal", "yellow light" stays "yellow light"
5. Remove quantity words from vehicle, pedestrian, and cyclist entities. Remove articles ("a", "an") and singular quantity ("one"). For quantities of two or more ("two", "three", "four", "several", "few", "many", "group of", etc.), replace the quantity word with "several".
  - Examples: "a car" -> "car", "one truck" -> "truck", "an SUV" -> "SUV", "a pedestrian" -> "pedestrian", "one cyclist" -> "cyclist"
  - Examples: "two cars" -> "several cars", "three pedestrians" -> "several pedestrians", "several cyclists" stays "several cyclists", "a few vehicles" -> "several vehicles", "group of pedestrians" -> "several pedestrians"
6. Remove outfit/clothing/appearance descriptions from pedestrian entities (what they are wearing, their clothing color, etc.) — do NOT remove action or state descriptors
  - Examples: "pedestrian wearing red" -> "pedestrian", "pedestrian in a blue jacket" -> "pedestrian", "woman dressed in black" -> "woman", "man with a red shirt" -> "man"
  - Keep: "crossing pedestrian" stays "crossing pedestrian", "stationary pedestrian" stays "stationary pedestrian", "pedestrian with stroller" stays "pedestrian with stroller"
7. Keep all descriptors (color, size, shape, brand, etc.) for other non-vehicle, non-pedestrian, non-cyclist entities (roads, lanes, buildings, signs, traffic lights, etc.)
8. Only include entities that are actually modified in your output — do not include unchanged entities.

Input format: a JSON list of entity strings
Output format: a JSON object {"original entity": "cleaned entity"} containing only modified entries

Input:
```

(c) Prompt for LPs de-noising.

LPs Unification:

```
You are an expert in autonomous driving knowledge graphs.

Given the following group of entity/edge names, determine which ones share \
the same meaning in the context of autonomous driving. If any entity does NOT \
share the same meaning as the others, call it out.

Return a JSON object with two keys:
- "same": a list of strings that share the same meaning
- "different": a list of strings that do NOT share the same meaning

Group:
{group_json}
```

(d) Prompt for LPs unification.

Figure 11: All relevant prompts used throughout TTCov's process.

Table 8: Additional subscores across all curation experiments corresponding to Tab. 2. Each cell reports the average score with standard deviation shown as subscript. Same overall EPDMS across Stage 1 and 2 reported in last column.

Stage 1

Budget	Method	NC	DAC	DDC	TLC	EP	TTC	LK	HC	EC	EPDMS
0.5x	Random	$92.74_{\pm 0.96}$	$63.11_{\pm 1.60}$	$91.81_{\pm 0.94}$	$99.56_{\pm 0.22}$	$80.86_{\pm 0.54}$	$91.33_{\pm 1.18}$	$86.07_{\pm 0.68}$	$97.70_{\pm 0.13}$	$77.48_{\pm 0.68}$	$18.95_{\pm 0.39}$
	Coreset	$94.74_{\pm 0.55}$	$69.56_{\pm 1.39}$	$95.78_{\pm 0.84}$	$99.33_{\pm 0.00}$	$81.97_{\pm 0.75}$	$92.44_{\pm 0.67}$	$88.22_{\pm 1.46}$	$97.70_{\pm 0.13}$	$80.00_{\pm 1.18}$	$20.77_{\pm 0.40}$
	SSE	$93.81_{\pm 0.42}$	$64.44_{\pm 2.47}$	$94.19_{\pm 1.32}$	$99.41_{\pm 0.13}$	$81.59_{\pm 0.29}$	$92.37_{\pm 0.78}$	$86.44_{\pm 0.44}$	$97.63_{\pm 0.13}$	$79.85_{\pm 2.86}$	$18.44_{\pm 1.52}$
	TTCov (ours)	$94.70_{\pm 0.23}$	$68.74_{\pm 1.28}$	$95.07_{\pm 0.45}$	$99.56_{\pm 0.00}$	$80.88_{\pm 0.34}$	$93.63_{\pm 0.13}$	$89.11_{\pm 0.89}$	$97.78_{\pm 0.00}$	$78.81_{\pm 1.85}$	$20.62_{\pm 0.52}$
0.75x	Random	$93.78_{\pm 0.51}$	$63.33_{\pm 3.01}$	$92.93_{\pm 0.64}$	$99.41_{\pm 0.13}$	$81.12_{\pm 0.86}$	$92.30_{\pm 0.78}$	$88.74_{\pm 0.34}$	$97.78_{\pm 0.00}$	$78.52_{\pm 0.93}$	$18.76_{\pm 1.36}$
	Coreset	$94.67_{\pm 0.68}$	$74.67_{\pm 2.22}$	$97.22_{\pm 0.38}$	$99.33_{\pm 0.00}$	$82.87_{\pm 0.75}$	$93.63_{\pm 0.26}$	$91.04_{\pm 1.80}$	$97.63_{\pm 0.13}$	$77.93_{\pm 3.37}$	$21.48_{\pm 1.80}$
	SSE	$94.81_{\pm 0.23}$	$72.37_{\pm 1.68}$	$96.89_{\pm 0.73}$	$99.33_{\pm 0.00}$	$81.43_{\pm 0.16}$	$92.96_{\pm 0.13}$	$88.81_{\pm 1.22}$	$97.63_{\pm 0.13}$	$78.81_{\pm 1.80}$	$22.29_{\pm 1.10}$
	TTCov (ours)	$94.96_{\pm 0.76}$	$73.85_{\pm 3.90}$	$97.70_{\pm 0.65}$	$99.63_{\pm 0.13}$	$81.39_{\pm 0.64}$	$93.70_{\pm 0.90}$	$91.56_{\pm 0.80}$	$97.78_{\pm 0.00}$	$78.37_{\pm 1.36}$	$23.00_{\pm 1.53}$
1x	Random	$94.33_{\pm 1.25}$	$68.15_{\pm 0.64}$	$94.96_{\pm 0.55}$	$99.48_{\pm 0.13}$	$81.69_{\pm 0.44}$	$92.52_{\pm 1.00}$	$89.11_{\pm 2.19}$	$97.78_{\pm 0.00}$	$79.41_{\pm 1.68}$	$20.15_{\pm 0.48}$
	Coreset	$95.04_{\pm 0.74}$	$75.41_{\pm 2.58}$	$98.11_{\pm 0.69}$	$99.33_{\pm 0.00}$	$82.56_{\pm 0.12}$	$93.63_{\pm 0.13}$	$91.78_{\pm 1.33}$	$97.70_{\pm 0.13}$	$74.81_{\pm 7.12}$	$23.63_{\pm 1.19}$
	SSE	$94.85_{\pm 0.23}$	$74.96_{\pm 2.38}$	$97.15_{\pm 0.74}$	$99.48_{\pm 0.13}$	$82.00_{\pm 0.83}$	$93.04_{\pm 0.51}$	$91.26_{\pm 1.73}$	$97.78_{\pm 0.00}$	$78.81_{\pm 3.22}$	$23.45_{\pm 0.53}$
	TTCov (ours)	$95.04_{\pm 0.97}$	$78.07_{\pm 4.14}$	$98.30_{\pm 0.51}$	$99.48_{\pm 0.13}$	$82.05_{\pm 0.13}$	$94.22_{\pm 0.80}$	$92.44_{\pm 0.67}$	$97.78_{\pm 0.00}$	$78.37_{\pm 2.96}$	$24.42_{\pm 0.66}$
1.25x	Random	$94.63_{\pm 1.05}$	$69.85_{\pm 2.36}$	$95.52_{\pm 0.90}$	$99.63_{\pm 0.13}$	$80.93_{\pm 0.01}$	$92.96_{\pm 1.26}$	$89.48_{\pm 0.26}$	$97.70_{\pm 0.13}$	$78.07_{\pm 1.03}$	$21.95_{\pm 0.42}$
	Coreset	$95.70_{\pm 0.83}$	$76.81_{\pm 0.84}$	$98.44_{\pm 0.69}$	$99.33_{\pm 0.00}$	$82.94_{\pm 0.53}$	$94.22_{\pm 0.00}$	$93.56_{\pm 0.44}$	$97.63_{\pm 0.13}$	$79.70_{\pm 0.51}$	$24.55_{\pm 1.39}$
	SSE	$95.78_{\pm 0.87}$	$76.00_{\pm 2.56}$	$97.78_{\pm 0.77}$	$99.48_{\pm 0.13}$	$81.78_{\pm 1.15}$	$94.22_{\pm 1.46}$	$91.85_{\pm 0.34}$	$97.70_{\pm 0.13}$	$79.56_{\pm 0.89}$	$24.76_{\pm 1.84}$
	TTCov (ours)	$94.89_{\pm 0.51}$	$76.59_{\pm 2.63}$	$96.89_{\pm 1.28}$	$99.41_{\pm 0.13}$	$81.43_{\pm 0.35}$	$94.30_{\pm 0.71}$	$91.56_{\pm 2.40}$	$97.78_{\pm 0.00}$	$80.15_{\pm 4.38}$	$25.48_{\pm 0.58}$
1.5x	Random	$94.74_{\pm 0.71}$	$75.04_{\pm 1.51}$	$96.52_{\pm 1.66}$	$99.48_{\pm 0.13}$	$81.47_{\pm 1.56}$	$93.63_{\pm 0.84}$	$90.22_{\pm 2.62}$	$97.78_{\pm 0.00}$	$73.78_{\pm 6.22}$	$22.95_{\pm 1.20}$
	Coreset	$95.59_{\pm 0.36}$	$78.59_{\pm 1.45}$	$98.30_{\pm 0.13}$	$99.48_{\pm 0.26}$	$82.27_{\pm 0.27}$	$94.44_{\pm 0.38}$	$92.52_{\pm 3.56}$	$97.70_{\pm 0.13}$	$80.00_{\pm 0.44}$	$25.89_{\pm 0.50}$
	SSE	$95.96_{\pm 0.45}$	$79.19_{\pm 1.89}$	$98.22_{\pm 0.69}$	$99.48_{\pm 0.13}$	$81.51_{\pm 0.62}$	$94.96_{\pm 0.46}$	$91.85_{\pm 1.30}$	$97.63_{\pm 0.13}$	$81.48_{\pm 0.26}$	$26.03_{\pm 1.22}$
	TTCov (ours)	$94.67_{\pm 0.67}$	$78.59_{\pm 2.31}$	$98.70_{\pm 0.57}$	$99.33_{\pm 0.00}$	$82.85_{\pm 0.09}$	$93.48_{\pm 0.90}$	$93.85_{\pm 1.14}$	$97.78_{\pm 0.00}$	$80.59_{\pm 1.85}$	$26.40_{\pm 1.26}$
–	Navtrain (oracle)	$95.56_{\pm 0.79}$	$77.78_{\pm 2.83}$	$97.61_{\pm 0.39}$	$99.44_{\pm 0.16}$	$84.29_{\pm 0.33}$	$94.56_{\pm 0.16}$	$92.67_{\pm 2.83}$	$97.78_{\pm 0.00}$	$77.78_{\pm 1.89}$	$24.49_{\pm 1.41}$

Stage 2

Budget	Method	NC	DAC	DDC	TLC	EP	TTC	LK	HC	EC	EPDMS
0.5x	Random	$79.86_{\pm 0.87}$	$58.74_{\pm 0.22}$	$75.97_{\pm 1.81}$	$97.99_{\pm 0.10}$	$77.64_{\pm 0.38}$	$77.43_{\pm 1.46}$	$44.58_{\pm 2.02}$	$97.59_{\pm 0.02}$	$81.26_{\pm 1.62}$	$18.95_{\pm 0.39}$
	Coreset	$79.63_{\pm 1.02}$	$62.92_{\pm 0.65}$	$77.70_{\pm 1.70}$	$98.02_{\pm 0.18}$	$80.28_{\pm 1.83}$	$77.01_{\pm 0.43}$	$42.89_{\pm 0.44}$	$97.71_{\pm 0.41}$	$79.91_{\pm 1.05}$	$20.77_{\pm 0.40}$
	SSE	$80.04_{\pm 0.84}$	$60.03_{\pm 1.06}$	$75.94_{\pm 2.59}$	$97.96_{\pm 0.22}$	$78.52_{\pm 0.22}$	$78.01_{\pm 1.00}$	$44.28_{\pm 1.51}$	$97.01_{\pm 0.66}$	$81.59_{\pm 2.53}$	$18.44_{\pm 1.52}$
	TTCov (ours)	$81.21_{\pm 0.63}$	$61.64_{\pm 1.05}$	$78.07_{\pm 0.93}$	$97.79_{\pm 0.48}$	$78.56_{\pm 1.19}$	$78.84_{\pm 0.62}$	$44.29_{\pm 0.16}$	$97.55_{\pm 0.37}$	$80.97_{\pm 2.16}$	$20.62_{\pm 0.52}$
0.75x	Random	$80.54_{\pm 2.17}$	$61.38_{\pm 1.56}$	$76.55_{\pm 1.22}$	$98.16_{\pm 0.24}$	$77.70_{\pm 2.28}$	$77.97_{\pm 1.76}$	$43.41_{\pm 1.26}$	$97.35_{\pm 0.29}$	$80.72_{\pm 2.83}$	$18.76_{\pm 1.36}$
	Coreset	$79.15_{\pm 0.66}$	$65.37_{\pm 1.81}$	$79.88_{\pm 1.80}$	$97.72_{\pm 0.49}$	$82.11_{\pm 0.38}$	$76.08_{\pm 1.27}$	$45.52_{\pm 0.35}$	$97.03_{\pm 0.32}$	$77.89_{\pm 1.40}$	$21.48_{\pm 1.80}$
	SSE	$81.39_{\pm 1.15}$	$62.97_{\pm 0.73}$	$78.23_{\pm 0.85}$	$97.85_{\pm 0.22}$	$80.23_{\pm 0.89}$	$78.59_{\pm 1.39}$	$44.29_{\pm 0.73}$	$97.12_{\pm 0.30}$	$80.18_{\pm 0.73}$	$22.29_{\pm 1.10}$
	TTCov (ours)	$80.05_{\pm 0.44}$	$66.78_{\pm 2.05}$	$82.08_{\pm 1.93}$	$98.14_{\pm 0.11}$	$80.39_{\pm 0.89}$	$77.28_{\pm 0.45}$	$45.42_{\pm 2.75}$	$97.56_{\pm 0.67}$	$78.75_{\pm 0.84}$	$23.00_{\pm 1.53}$
1x	Random	$80.86_{\pm 1.28}$	$62.52_{\pm 1.91}$	$78.41_{\pm 2.39}$	$97.89_{\pm 0.58}$	$80.19_{\pm 0.99}$	$78.53_{\pm 1.76}$	$44.91_{\pm 3.43}$	$97.04_{\pm 0.40}$	$80.29_{\pm 2.24}$	$20.15_{\pm 0.48}$
	Coreset	$80.29_{\pm 0.91}$	$67.33_{\pm 2.68}$	$81.43_{\pm 0.93}$	$97.94_{\pm 0.19}$	$81.63_{\pm 0.18}$	$78.13_{\pm 1.54}$	$46.04_{\pm 1.44}$	$97.03_{\pm 0.33}$	$71.79_{\pm 5.99}$	$23.63_{\pm 1.19}$
	SSE	$81.24_{\pm 1.06}$	$67.08_{\pm 0.27}$	$80.65_{\pm 1.32}$	$98.19_{\pm 0.20}$	$80.51_{\pm 0.94}$	$79.24_{\pm 0.86}$	$45.10_{\pm 1.17}$	$97.18_{\pm 0.37}$	$77.69_{\pm 0.60}$	$23.45_{\pm 0.53}$
	TTCov (ours)	$80.80_{\pm 0.20}$	$69.03_{\pm 0.95}$	$81.85_{\pm 0.58}$	$97.73_{\pm 0.37}$	$81.36_{\pm 0.50}$	$77.76_{\pm 1.58}$	$43.94_{\pm 0.05}$	$97.03_{\pm 0.35}$	$78.79_{\pm 1.11}$	$24.42_{\pm 0.66}$
1.25x	Random	$80.94_{\pm 0.81}$	$66.51_{\pm 1.13}$	$80.86_{\pm 1.22}$	$98.08_{\pm 0.23}$	$78.61_{\pm 1.16}$	$78.83_{\pm 1.66}$	$44.70_{\pm 1.49}$	$96.88_{\pm 0.07}$	$79.00_{\pm 3.13}$	$21.95_{\pm 0.42}$
	Coreset	$80.05_{\pm 0.88}$	$70.54_{\pm 0.63}$	$83.41_{\pm 0.63}$	$98.05_{\pm 0.53}$	$82.53_{\pm 0.87}$	$77.10_{\pm 1.81}$	$48.22_{\pm 1.46}$	$97.21_{\pm 0.13}$	$74.45_{\pm 1.74}$	$24.55_{\pm 1.39}$
	SSE	$80.44_{\pm 0.62}$	$70.51_{\pm 2.50}$	$82.80_{\pm 1.53}$	$98.30_{\pm 0.24}$	$81.26_{\pm 0.72}$	$78.17_{\pm 1.03}$	$47.05_{\pm 0.59}$	$96.58_{\pm 0.37}$	$77.51_{\pm 2.21}$	$24.76_{\pm 1.84}$
	TTCov (ours)	$80.90_{\pm 1.74}$	$69.60_{\pm 1.53}$	$83.24_{\pm 0.90}$	$98.29_{\pm 0.49}$	$80.65_{\pm 0.81}$	$78.37_{\pm 1.82}$	$47.58_{\pm 0.70}$	$97.43_{\pm 0.24}$	$77.11_{\pm 0.91}$	$25.48_{\pm 0.58}$
1.5x	Random	$80.13_{\pm 0.91}$	$67.63_{\pm 2.03}$	$80.83_{\pm 0.59}$	$97.90_{\pm 0.36}$	$80.22_{\pm 2.38}$	$78.18_{\pm 1.08}$	$45.25_{\pm 1.00}$	$97.08_{\pm 0.10}$	$73.95_{\pm 1.89}$	$22.95_{\pm 1.20}$
	Coreset	$81.53_{\pm 0.40}$	$70.53_{\pm 0.26}$	$82.53_{\pm 1.25}$	$98.09_{\pm 0.14}$	$82.35_{\pm 0.57}$	$78.76_{\pm 0.72}$	$47.77_{\pm 0.61}$	$97.04_{\pm 0.20}$	$74.86_{\pm 0.75}$	$25.89_{\pm 0.50}$
	SSE	$80.35_{\pm 1.84}$	$69.23_{\pm 0.37}$	$82.63_{\pm 0.39}$	$98.46_{\pm 0.34}$	$82.11_{\pm 1.23}$	$78.77_{\pm 1.16}$	$45.14_{\pm 1.02}$	$96.76_{\pm 0.35}$	$76.89_{\pm 1.82}$	$26.03_{\pm 1.22}$
	TTCov (ours)	$80.19_{\pm 0.29}$	$69.33_{\pm 1.03}$	$83.08_{\pm 0.41}$	$98.13_{\pm 0.57}$	$82.32_{\pm 0.13}$	$77.64_{\pm 0.58}$	$46.06_{\pm 0.75}$	$96.77_{\pm 0.14}$	$76.29_{\pm 1.07}$	$26.40_{\pm 1.26}$
–	Navtrain (oracle)	$80.20_{\pm 1.25}$	$66.93_{\pm 1.28}$	$80.76_{\pm 0.04}$	$98.58_{\pm 0.10}$	$85.45_{\pm 0.21}$	$77.95_{\pm 1.60}$	$44.46_{\pm 0.08}$	$96.39_{\pm 0.85}$	$72.65_{\pm 0.55}$	$24.49_{\pm 1.41}$

Abbreviations: NC = no at-fault collisions; DAC = drivable area compliance; DDC = driving direction compliance; TLC = traffic light compliance; EP = ego progress; TTC = time-to-collision within bound; LK = lane keeping; HC = history comfort; EC = extended comfort.

C EPDMS Submetrics

Additional Results. In Tab. 8, we report the full set of submetrics corresponding to Tab. 2.

D Optimizer Ablations

D.1 Greedy Residual Matching

A natural alternative to KL distribution matching is to match LP counts directly. Let $\mathbf{n}(x)$ denote the LP-count vector of sample x, with entries $n(x, L)$. Define the desired LP totals at budget B as

$$\mathbf{m} \;=\; B\,p^{\star}. \tag{3}$$

If $\mathcal{S}^{(t)}$ is the selected set after t iterations, define the residual

$$\mathbf{r}^{(t)} \;=\; \mathbf{m} \;-\; \sum_{x \in \mathcal{S}^{(t)}} \mathbf{n}(x). \tag{4}$$

Using weighted squared error,

$$L(\mathcal{S}) \;=\; \left\| \mathbf{m} \;-\; \sum_{x \in \mathcal{S}} \mathbf{n}(x) \right\|_W^2, \qquad W = \mathrm{diag}(w_1, \ldots). \tag{5}$$

If candidate sample x is added, the new residual is $\mathbf{r}^{(t)} - \mathbf{n}(x)$, and the improvement is

$$\|\mathbf{r}^{(t)}\|_W^2 \;-\; \|\mathbf{r}^{(t)} - \mathbf{n}(x)\|_W^2 \;=\; 2\,\mathbf{n}(x)^\top W \mathbf{r}^{(t)} \;-\; \mathbf{n}(x)^\top W \mathbf{n}(x). \tag{6}$$

Therefore the greedy step is

$$x^{(t+1)} \;=\; \arg\max_{x \,\in\, \mathcal{C} \setminus \mathcal{S}^{(t)}} \Big\{ 2\,\mathbf{n}(x)^\top W \mathbf{r}^{(t)} \;-\; \mathbf{n}(x)^\top W \mathbf{n}(x) \Big\}. \tag{7}$$

The first term rewards samples that fill LPs we still need. The second penalizes any LP added, so as the residual decreases, we naturally stops adding samples that would overshoot. We use the smoothed weighting $w_L = 1/(p^\star(L) + \tau)$ with $\tau = 1/B$ as the default, which uplifts rare LPs without diverging as $p^\star(L) \to 0$. Setting $W = I$ recovers the unweighted variant.

D.2 Repeat Factor Sampling

The proportional K-Atlas target $p^\star$ in (1) contains many common LPs, so a budget constrained optimization can under select rare LPs. We adapt repeat-factor sampling (RFS) [22], originally introduced for long-tailed instance segmentation, to uplift rare LPs in the target before optimization. Let $c(L)$ denote the raw Atlas count for LP L and N_{test} the number of test samples, so the total LP frequency is $f(L) = c(L)/N_{\text{test}}$. We compute a repeat factor per LP

$$r(L) \;=\; \max\!\Big(1,\; \sqrt{t \,/\, f(L)}\Big), \tag{8}$$

where t is the threshold hyperparameter (default $t = 10^{-2}$). LPs with $f(L) \geq t$ are unchanged ($r(L) = 1$), and rarer LPs are amplified inversely to the square root of their frequency. The reweighted target distribution is

$$p_{\text{RFS}}(L) \;=\; \frac{c(L)\, r(L)}{\sum_{L'} c(L')\, r(L')}, \tag{9}$$

which substitutes for $p^\star$ in (2). Intuitively, t controls how aggressively rare LPs get amplified. At low t (e.g. 10^{-3}) only the rarest LPs are boosted and the target stays close to proportional, while at high t (e.g. 10^{-1}) most LPs count as rare and end up taking a larger share of the budget than they did under proportional weighting.

D.3 Penalty

To approach (2) greedily, we add one sample at a time. Let $\mathcal{S}^{(t)}$ be the curated set after t picks, $n(x, L)$ the count of LP L in sample x, $n(x) = \sum_L n(x, L)$, and $q^{(t)}(L) = \sum_{x \in \mathcal{S}^{(t)}} n(x, L)$ with $S^{(t)} = \sum_L q^{(t)}(L)$ the running coverage counts and total mass. Expanding the marginal KL gain $D_{\text{KL}}(p^\star \,\|\, \hat{p}_{\mathcal{S}^{(t)}}) - D_{\text{KL}}(p^\star \,\|\, \hat{p}_{\mathcal{S}^{(t+1)}})$ gives the per-candidate score

$$\text{score}(x) \;=\; \sum_L p^\star(L)\, \log\!\left(\frac{q^{(t)}(L) + n(x, L)}{q^{(t)}(L)} \right) \;-\; \log\!\left(\frac{S^{(t)} + n(x)}{S^{(t)}} \right), \tag{10}$$

and we pick $x^{(t+1)} = \arg\max_{x \in \mathcal{C} \setminus \mathcal{S}^{(t)}} \text{score}(x)$.

Equation (10) treats every LP symmetrically, so a sample whose LPs would only added to saturated K-Atlas LPs $B\,p^\star(L)$ is scored only through its KL gain, not its overshoot. We introduce a penalty $\rho \geq 0$ that penalizes candidates by how much they add to already saturated LPs:

$$\text{score}^{(\rho)}(x) \;=\; \text{score}(x) \;-\; (\rho - 1)\, \frac{\sum_L n(x, L)\, \big[q^{(t)}(L) - B\, p^\star(L) \big]_+}{S^{(t)}}, \tag{11}$$

with $[z]_+ = \max(0, z)$. Setting $\rho = 1$ zeroes the second term, recovering our original greedy score (10). $\rho > 1$ penalizes over-coverage and $\rho < 1$ rewards it.

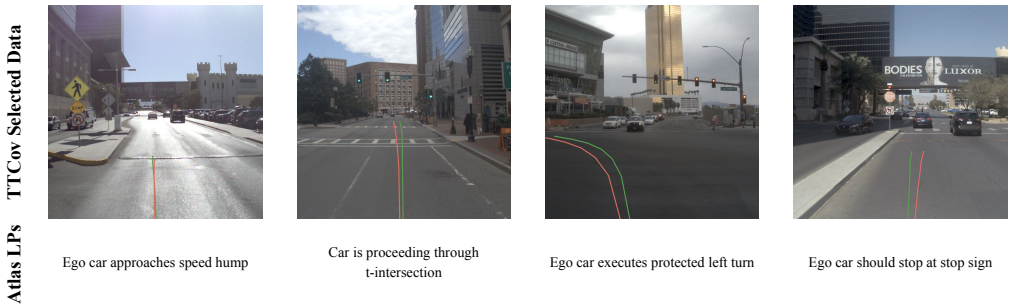


Figure 12: Additional visualization examples of TTCov selected data.

E Datasets and Virtual Clip Creation

In this section, we provide details of the datasets and the virtual clip strategy we use for the experiments.

The OpenScene dataset consists of continuous driving logs sampled at 2Hz. Frameworks such as Navsim typically process these logs using a sliding window to create individual "scenes". In Navsim, by default configuration, a scene is a 14-frame window consisting of a 3-frame history, 1 frame for the current state and a 10-frame future horizon. To better capture the actions during driving and align with both academic and industry standards [18], we define a 20-frame virtual clip (10 seconds) as our fundamental unit for captioning and curation. Each 20-frame virtual clip therefore encapsulates 7 overlapping scenes. To ensure that every possible scene in the OpenScene trainval split is covered by one virtual clip, we employ a 7-frame stride to get the virtual clips. Such a virtual clip strategy makes sure all the scenes in OpenScene trainval are covered without the massive overhead of frame-by-frame sliding window. During the data curation process, we use the virtual clip as the fundamental unit, such that when a virtual clip is selected, all its 7 encapsulated scenes are added into the train dataset.

F Additional Visualizations

In Fig. 12, we provide additional visualizations of TTCov's selected data, demonstrating diversity in visual, semantic, and behavioral space.

G Code Release

All code will be released alongside the final version of this work.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide thorough analyses, experimental results, ablations, and visualizations in Sec. 4 and Sec. 5

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Yes, limitations discussed in Sec. 7.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: We do not have any theoretical proofs. Equations used are all provided in Secs. 3 and D.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed explanation of our methods are in Sec. 3 and Secs. B, D and E.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All code will be released alongside the final version of this work. Datasets used are already publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed training and evaluation details are in Secs. 3 and 5 and Secs. B, D and E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report averaged performance across three runs for all model evaluation performances in the main table. For brevity, submetrics and standard deviations are in Sec. C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Resources reported in Sec. 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We do not have any human subject experiments. All data, LLMs, and MLLMs used are publicly available. All code will be released alongside the final version of this work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impact discussion located in Sec. A.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We use existing public models and datasets, requiring no such safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All used datasets, model, and code are properly cited in Secs. 3 and 5.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Any code, prompts, data, etc. are documented throughout paper and appendix. We do not create any new data.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: We do not have any such experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: We do not have any research requiring human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We used LLMs as part of our method, which we state clearly in our method description along with the types of models used.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.