CURASCENE: INTERACTION-AWARE SCENARIO CURATION FOR DATA-EFFICIENT AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory forecasting models for autonomous driving are trained on scenario corpora that now reach hundreds of thousands of samples, yet the marginal value of an additional scenario is far from uniform. We argue that much of this redundancy is *structural*: what makes a driving scenario informative is largely the pattern of *predictive influence* among its agents—whose past states help forecast whose futures—and large corpora repeat a small number of interaction structures while varying mostly in surface trajectory geometry. We present CuraScene, which makes this structure explicit and selects on it. For each scenario, CuraScene infers—without interaction supervision—a directed graph over the agents and, where annotated, the traffic signals that control them. An edge $i \to j$ indicates that the lagged state of i improves prediction of j. The graph is learned with an encoder–decoder trained under a sparsity penalty and a physically admissible edge mask. Scenarios are then selected greedily by the topological novelty of their ego-centric subgraph relative to the already-selected set, so a fixed budget covers interaction structures rather than raw samples. We do not claim a new graph structure learning algorithm; our claim is that inferred interaction structure is a useful and inexpensive selection signal for driving data. At a matched 50% budget on Waymo Open Motion and Argoverse 2, two forecasting models trained on CuraScene-selected scenarios reduce minADE_6 by 10.7–14.6% relative to budget-matched random subsampling, and an ablation shows that the learned graph and its admissibility prior are both required in order to recover the agents Waymo annotates as interacting.

1 INTRODUCTION

Learning-based autonomous driving has followed the familiar recipe of scaling data. Motion forecasting corpora have grown from a few thousand tracked sequences to the roughly 250k scenarios of Argoverse 2 (Wilson et al., 2021) and the multi-stage pretraining corpora behind recent driving foundation models (Wang et al., 2025). But the returns on raw volume are sublinear (Kaplan et al., 2020), while the cost of storing, transforming, and training on a scenario is not. For a practitioner with a fixed compute or annotation budget the question is therefore not how much data exists, but *which* scenarios to train on. Coreset selection and data pruning made this question precise for image classification (Sener & Savarese, 2018; Paul et al., 2021; Sorscher et al., 2022), but the signals they rely on—embedding geometry, training loss, gradient norms—are agnostic to what makes a driving scenario hard.

Driving scenarios are redundant in interaction structure, not in appearance. A forecasting model fails on a scenario when it mispredicts how one agent responds to another: a vehicle that yields at an unprotected turn, a merge that forces the ego vehicle to decelerate, a pedestrian whose crossing blocks a right turn. What determines this difficulty is the pattern of predictive influence among agents—whose past states help forecast whose futures—and a scene of sixty tracked objects of which fifty-seven are parked or distant is, for prediction purposes, a three-agent scene. Two scenarios recorded in different cities can induce exactly the same pattern, so the second adds little the first has not already supplied; conversely, two scenarios nearly identical in trajectory space can

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

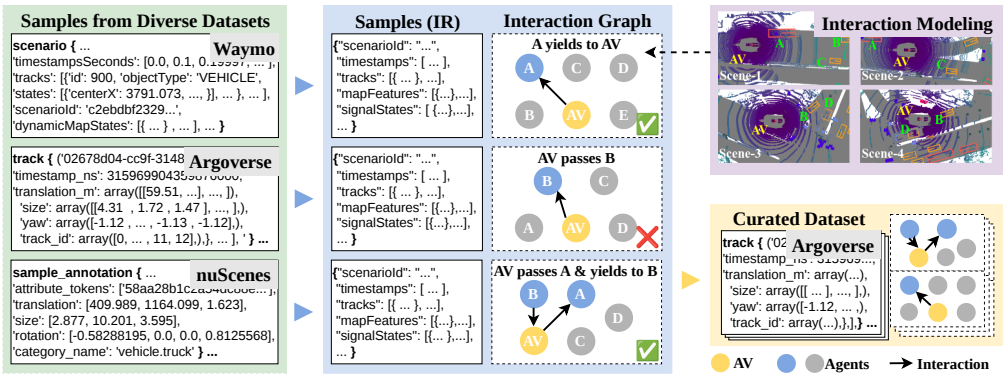


Figure 1: Scenarios drawn from heterogeneous sources are normalized into a common representation, converted into directed interaction graphs, and then selected by the novelty of their ego-centric structure. The third scenario is rejected not because its trajectories resemble an earlier one, but because its predictive-influence topology does.

be structurally opposite, depending on whether the ego vehicle yields or is yielded to. Signals computed from trajectory features or from model loss do not expose this distinction, because it is a property of the relational structure and not of any individual trajectory (Figure 1).

The structure is not labeled. Acting on interaction structure requires knowing it, and the major forecasting corpora do not provide it. Waymo Open Motion (Ettinger et al., 2021) flags only a handful of agents per scenario as interacting, out of more than sixty tracked on average, and does so as an unordered set rather than a relation, so neither the direction of influence nor its semantics is recorded; Argoverse 2 (Wilson et al., 2021) and nuScenes (Caesar et al., 2019) provide no interaction annotation at all (Appendix B). Annotating influence manually at the scale of 10^5 scenarios is not viable, so the structure has to be inferred from the trajectories themselves, without supervision, and cheaply enough that inference over the full pool costs less than training on it.

CuraScene. We propose CuraScene, an interaction-aware curation method with two components. The first treats a scenario as a latent directed graph over its agents, together with the traffic signals that control them where a corpus annotates them, and infers it under a structural-equation view: an edge $i \to j$ is retained when the lagged state of i improves prediction of j's future state, a nonlinear Granger criterion (Tank et al., 2021). An encoder proposes a sample-specific adjacency, a decoder must roll the scenario forward given only that adjacency, and the two are trained jointly with an ℓ_1 sparsity penalty under a hard mask forbidding physically implausible edges; no interaction labels are used at any point. The second component extracts the ego-centric subgraph, measures its topological novelty against the already-selected set with a normalized graph edit distance, and admits scenarios greedily under a budget, so that the budget covers interaction structures rather than samples. Both components run behind a schema-unification layer that reconciles the differing formats, field names, units, and coordinate conventions of the source corpora; this is infrastructure rather than contribution, and we defer it to Appendix A.

Claims and results. We do not propose a new graph structure learning algorithm: the encoder–decoder, the sparsity penalty, and the Granger reading of an edge are all established, and the inference component is deliberately assembled from them. Our claim is narrower, that interaction structure—even when inferred imperfectly and without supervision—is a selection signal that outperforms the budget-matched random sampling which remains the default in driving research and a strong baseline in coreset selection generally (Guo et al., 2022). The contribution is therefore the formulation of structural coverage as a selection criterion (Section 3), its instantiation without interaction labels (Section 4), and the matched-budget evidence together with an honest account of where that evidence is thin (Sections 5 and 6). At a matched 50% budget, the generative SeNeVa (Lu et al., 2024) and the query-decoupled DeMo (Zhang et al., 2024) improve by 10.7–14.6% in minADE_6 on Waymo Open Motion and Argoverse 2, and removing the admissibility prior or the

learned graph drops the F1 of interacting-agent identification from 0.80 to 0.42 and 0.60, indicating that the inferred structure and not the surrounding pipeline carries the signal.

2 Related Work

Selecting and curating training data. Subset selection for deep models has a mature toolkit: geometric methods cover a feature space by farthest-point traversal or a submodular coverage function (Sener & Savarese, 2018; Wei et al., 2015; Welling, 2009), difficulty-based methods rank examples by forgetting events or early-training error and prune the easy tail (Toneva et al., 2019; Paul et al., 2021; Sorscher et al., 2022), and matching methods choose subsets whose training signal approximates the full set's (Mirzasoleiman et al., 2020; Killamsetty et al., 2021b;a), with proxy models making any of these affordable at scale (Coleman et al., 2020). Random selection nonetheless remains a strong baseline (Guo et al., 2022), which is part of why we treat it as our reference point. These criteria differ in whether they score a sample on its own or a subset jointly, but all of them measure difficulty or diversity in an embedding, loss, or gradient space rather than in the predictive-influence topology a scenario induces among its agents. Within driving, the usual responses to redundant data are to generate what is missing—safety-critical scenario generation searches for configurations that induce failure, and its output is synthetic by construction (Ding et al., 2023; Rempe et al., 2022; Huai et al., 2023; Cheng et al., 2023)—or to reweight the loss so that long-tail cases share information in embedding space (Makansi et al., 2021). Selection from recorded driving data has instead used semantic surface features for test reduction and prioritization (Deng et al., 2022), post-hoc analysis of recorded failures (Sun et al., 2024; Feng et al., 2024), and active learning for perception annotation (Luo et al., 2023; Liu et al., 2024; Acuna et al., 2018; Liu et al., 2022). None of these selects on inferred influence topology or evaluates the selection by its effect on downstream forecasting accuracy at a matched budget.

Interaction structure in multi-agent forecasting. Most forecasting models treat interaction as a graph that they either assume complete and attend over (Gao et al., 2020; Zhou et al., 2022; 2023) or fix in advance by heuristic relations (Salzmann et al., 2020; Sun et al., 2022; Rowe et al., 2023). Recovering the graph instead is the subject of graph structure learning: neural relational inference treats edges as latent variables in a variational autoencoder (Kipf et al., 2018), amortized causal discovery shares an inference network across samples so that a sample-specific graph comes out of one forward pass (Löwe et al., 2022), and neural Granger causality reads edges off the sparsity pattern of a predictive model (Tank et al., 2021; Li et al., 2020). Our inference stage is an instance of this family rather than an addition to it, and we say so plainly; the departure is in what the graph is for. In the work surveyed above the structure is consumed by the model that produced it, in service of a prediction, whereas here it becomes a statistic over the dataset, used to decide which samples a different model should be trained on.

3 Interaction-Aware Scenario Selection

Budgeted selection. Let $\mathcal{D} = \{x_1, \dots, x_N\}$ be a pool of driving scenarios, where each x_i contains the tracked states of a set of agents over a fixed horizon together with map context and, where the source provides it, traffic-signal context. A forecasting model f_θ trained on a subset $S \subseteq \mathcal{D}$ incurs risk $R(S) = \mathbb{E}_{x \sim P_{\text{test}}}\big[\ell\big(f_{\theta(S)}(x), y(x)\big)\big]$, and the selection problem under a budget $B \ll N$ is $S^\star = \arg\min_{|S|=B} R(S)$. This is intractable: evaluating one candidate subset requires training a model, and the search space is combinatorial. Practical methods therefore replace R with a surrogate computable on the pool before any downstream training, typically a geometric surrogate in some feature or gradient space (Sener & Savarese, 2018; Mirzasoleiman et al., 2020; Killamsetty et al., 2021a). We choose a structural one instead.

Scenarios as directed interaction graphs. We associate with each scenario x_i a latent directed graph $G_i = (V_i, E_i)$, in which a directed edge $(u \to v) \in E_i$ asserts *predictive influence*: the past of u carries information about the future of v beyond what v's own past carries. We write *influence* for this relation throughout, always in that predictive sense and never as a claim about intervention. Nodes V_i are the agents present in the scenario, and, where a source publishes signal states together with the lanes each signal controls, the controlling traffic signals as well. Every

source we use therefore supports agent-only inference, and Waymo, the one source that publishes both fields, additionally permits signal enrichment (Appendix B); nothing else in the formulation changes between the two cases. Direction matters and is not a technicality: a scenario in which the ego vehicle yields and a scenario in which it is yielded to have the same undirected structure and opposite behavioral content. The full graph mixes interactions that shape the scenario with others confined to its periphery, so we work with a local view: the *ego-centric subgraph* $g_i \subseteq G_i$, obtained by keeping the ego node, the nodes within r hops of it in G_i, and the induced edges. We take the ego node to be the recording vehicle, which is the one entity every corpus identifies explicitly and around which each recording was collected; it serves as a fixed reference point for describing a scenario's structure and is not in general the agent whose forecasting error is later measured, a distinction Section 6 returns to. Figure 4 in Appendix E shows such a subgraph, with agents separated into those that influence the ego vehicle directly, those that do so through an intermediary, and those that do not influence it at all.

Structural coverage as the surrogate. The hypothesis this paper tests is that $R(S)$ is governed largely by how well the ego-centric structures appearing at test time are represented among the structures in S; if this holds, a budget is spent well when the selected subset covers the space of ego-centric structures rather than the space of raw scenarios. Given a distance d between ego-centric subgraphs, we make this operational by defining the novelty of a candidate with respect to an existing selection S as

$$\Delta(g_i, S) \;=\; \min_{g_j \in S} d(g_i, g_j), \qquad d(g_i, g_j) \;=\; \frac{\mathrm{GED}(g_i, g_j)}{|V_i| + |E_i|}, \tag{1}$$

where GED is graph edit distance and the denominator normalizes for subgraph size so that large scenarios are not admitted merely for being large. The normalization makes d asymmetric, so we treat it as a novelty score rather than a metric; this costs us the approximation guarantees that hold for metric dispersion, a point we return to in Section 6.

Selecting to maximize coverage under this score is the max–min dispersion problem, $\max_{|S|=B} \min_{i \neq j \in S} d(g_i, g_j)$, for which the standard greedy farthest-point traversal is also the algorithm underlying k-center coreset selection (Sener & Savarese, 2018). CuraScene uses the streaming form of that greedy rule: scenarios are visited in the order the corpus supplies them, and a candidate is admitted when $\Delta(g_i, S) > \tau_c$. We calibrate τ_c by bisection on a random subsample of the pool, taking the smallest value whose admission rate meets the target $|S| = B$ and applying it unchanged to the full pass. The rule is order-dependent by construction: a different permutation of the stream admits a different subset, and the batch variant, which repeatedly admits the pool element of maximum Δ, need not agree with either. We fix one visiting order per corpus, do not average over permutations, and leave that sensitivity untested.

Neither the objective nor its greedy solution is new; what differs from prior coreset work is the space in which distance is measured—not an embedding of the sample, but the topology of the predictive-influence relation it induces among its agents. Selection therefore becomes invariant to surface properties that leave that topology unchanged, such as absolute position, map geometry, or recording city, and sensitive to the one property ordinary trajectory features do not express, the direction of influence.

4 METHOD

Section 3 defined selection as structural coverage but assumed the structure was available; this section supplies it. Figure 2 shows the pipeline: Section 4.1 recovers a directed interaction graph per scenario without interaction labels, and Section 4.2 turns those graphs into a budgeted selection. Both stages read one intermediate representation into which every source is normalized beforehand by a language-model-assisted schema layer; that layer is infrastructure rather than contribution, nothing downstream depends on how a scenario reached the representation, and Appendix A documents it in full.

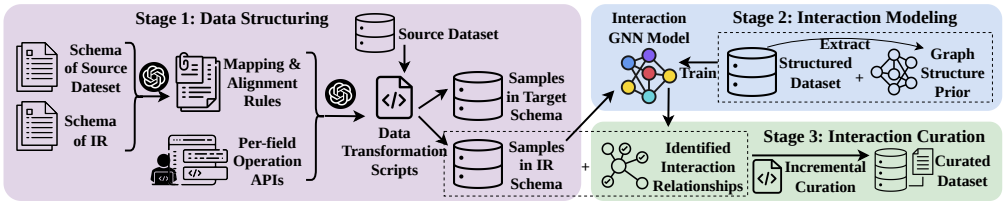


Figure 2: The CuraScene pipeline. Heterogeneous sources are first normalized into one intermediate representation (left; Appendix A). An encoder–decoder then infers a sample-specific directed interaction graph under a heuristic admissibility prior (center), and the resulting ego-centric structures drive budgeted selection (right).

4.1 Inferring the Directed Interaction Graph

A structural-equation view. Writing $\mathbf{z}^{(k)}$ for the joint state sequence of all entities in scenario k and $z_j(t)$ for the state of entity j at time t, we posit that each entity evolves as a function of its parents in a latent directed graph,

$$z_j(t) \;=\; f_j\big(\mathbf{z}_{\mathrm{pa}(j)}(t{-}L{:}t{-}1)\big) \;+\; \epsilon_j(t), \tag{2}$$

with $\mathrm{pa}(j)$ the parent set, f_j a learnable non-linear predictor, L the lag, and ϵ_j exogenous noise. Under this parameterization an inferred edge $i \to j$ carries the non-linear Granger reading we want—including the lagged states of i improves the prediction of j (Tank et al., 2021)—which is what makes the recovered structure directed rather than a proximity graph. We do not claim identifiability: Equation 2 is a modeling assumption, confounding is possible, and the recovered edges are predictive rather than causal in any interventional sense.

Admissibility prior. Estimating a directed graph over all entity pairs from a few seconds of trajectory is badly underdetermined, so we restrict the hypothesis space with a mask $M \in \{0,1\}^{n\times n}$ from physics and traffic context, where $M_{ij} = 0$ forbids the edge $i \to j$. Two rules generate it. For a pair of agents we admit *both* directed edges when the two come within a distance threshold $d_{\max}$ of each other at some point in the observation window and at least one of them exhibits non-trivial motion over it; proximity alone is not enough, since two parked vehicles are never candidates, and motion alone is not enough, since agents that never approach each other cannot interact. Both edges are admitted because the direction cannot be settled a priori. For a traffic signal in a red or yellow state we admit edges from the signal to every agent on a lane it controls. The first rule needs only agent kinematics and so applies to every source; the second is the signal enrichment of Section 3 and fires on Waymo alone among the corpora we use, with Argoverse 2 masked by the first rule and nothing else in the pipeline altered. The prior is deliberately permissive—its job is to remove pairs that cannot interact, not to decide who influences whom—and Section 5 shows the prior alone is a poor interaction detector.

Encoder–decoder. An encoder maps the scenario to a sample-specific weighted adjacency, and a decoder must reproduce the scenario dynamics using only that adjacency:

$$\Phi_{\mathrm{enc}} : \mathbf{z}^{(k)} \mapsto A^{(k)}, \qquad A^{(k)} \leftarrow A^{(k)} \odot M^{(k)}, \qquad \Phi_{\mathrm{dec}} : \big(\mathbf{z}^{(k)}, A^{(k)}\big) \mapsto \tilde{\mathbf{z}}^{(k)}_{\mathrm{next}}. \tag{3}$$

The coupling supplies the training signal: the encoder is rewarded only for edges the decoder can actually use to explain the observed evolution. Both networks are graph networks over the entity set, trained as a variational autoencoder whose latent variables are the edges (Kingma et al., 2013): the encoder emits a logit per admissible ordered entity pair from the two entities' trajectory encodings, and edges are sampled through a Gumbel–softmax relaxation so that a discrete adjacency remains differentiable (Kipf et al., 2018). The decoder is then asked to roll the scenario forward twenty frames given only the sampled adjacency; layer widths, the lag L, and the relaxation temperature are in Appendix D.

Objective. The encoder defines a variational posterior that factorizes over the admissible pairs, $q_\phi(A \mid \mathbf{z}) = \prod_{(i,j):M_{ij}=1} q_\phi(a_{ij} \mid \mathbf{z})$, with each factor a Bernoulli over the presence of the edge and

the masked pairs held deterministically absent. We place on it a factorized prior $p(A)$ with a fixed edge probability ρ, and fit both networks over the pool by minimizing the negative evidence lower bound together with an explicit sparsity penalty,

$$\mathcal{L}_{\text{total}} = \frac{1}{N}\sum_{k=1}^{N}\Big(\mathbb{E}_{q_\phi}\big[\mathcal{L}_{\text{fit}}\big(\Phi_{\text{dec}}(\mathbf{z}^{(k)}, A^{(k)}),\, \mathbf{z}^{(k)}_{\text{next}}\big)\big] + \beta\,\mathrm{KL}\big(q_\phi(A^{(k)} \mid \mathbf{z}^{(k)}) \,\|\, p(A)\big) + \lambda\big\|A^{(k)}\big\|_1\Big), \tag{4}$$

where $\mathcal{L}_{\text{fit}}$ is the negative log-likelihood of the next window under a fixed-variance Gaussian decoder, so the first two terms are the negative ELBO up to an additive constant. Because both q_ϕ and p factorize, the divergence is a sum of per-pair terms over the admissible set, and the expectation is estimated with a single reparameterized Gumbel–softmax sample. We retain the ℓ_1 term because λ is what we tune to reach a given sparsity *level*, and that level matters more here than it usually does: when the graph is dense the decoder reconstructs well while carrying no selection signal, every ego-centric subgraph collapses to the same topology, and the distance in Equation 1 stops discriminating. No interaction labels enter Equation 4; Waymo's labels are used only for evaluation in Section 5.

Recovering the graph. After training we threshold the encoder output at τ_e to obtain $\hat{G}^{(k)}$, keeping the bidirectional edge when both directions of a pair survive—the correct reading for mutual negotiation such as a merge—and only the stronger edge when one direction dominates. The ego-centric subgraph g_k is the r-hop neighborhood of the ego node, the recording vehicle each corpus identifies explicitly. The benchmarks do not score that node: error is measured on the agents they designate for prediction, at most eight `tracks_to_predict` per Waymo scenario and a single focal track in Argoverse 2 that is a separate entity from the recording vehicle. The ego-centric subgraph is therefore a summary of the interaction context a scenario offers, not of the immediate neighborhood of the scored agent; we treat that gap as a limitation rather than a property we exploit (Section 6). Values for β, ρ, λ, τ_e, and r, and the optimizer and schedule, are in Appendix D.

4.2 Structure-Guided Budgeted Selection

The rule runs in a single pass: each scenario is encoded once, its masked adjacency thresholded at τ_e, its ego-centric subgraph extracted, and the scenario admitted when that subgraph's novelty exceeds τ_c; Algorithm 1 in Appendix D states it in full. Scenarios whose ego node has no surviving in-edges—the ego vehicle is influenced by nobody, the common case in free-flow driving—produce the same trivial subgraph and are admitted at most once, which is where most of the reduction comes from.

Cost. Exact graph edit distance is NP-hard and the rule appears to need $|S|$ distance evaluations per candidate, but after sparsification the ego-centric subgraphs hold a handful of nodes rather than the sixty-plus agents of the raw scene, so each comparison is cheap and the space of distinct topologies saturates quickly. We nonetheless apply two standard reductions, an approximate bipartite assignment in place of the exact distance (Riesen & Bunke, 2009) and a relabeling-based signature that rejects structurally identical candidates before any distance is computed (Shervashidze et al., 2011); Appendix D names the solver and the signature scheme. Because the rule is incremental, the same code path would serve online filtering of rollouts as they are generated, the regime that matters for post-training; we evaluate only the offline setting.

5 Experiments

We ask three questions: does selecting on interaction structure beat spending the same budget at random (Section 5.2), is the gain attributable to the inferred structure or to the surrounding pipeline (Section 5.3), and what does the method cost (Section 5.4)?

5.1 Setup

Datasets. From Waymo Open Motion (Ettinger et al., 2021), which provides dense urban traffic with more than sixty tracked agents per scenario on average, we use 1,000 training and 150 validation scenarios; these are individual scenarios and not `tfrecord` shards. Selection runs over

Table 1: Forecasting accuracy at a matched 50% data budget. Both arms select the same number of scenarios from the same pool and are trained with identical architectures, optimizers, and schedules; they differ only in which scenarios are selected. Lower is better for minADE_6 and minFDE_6. Relative improvement is over the budget-matched random arm.

		Random		**CuraScene**		**Rel. improv.**	
Dataset	**Model**	minADE_6	minFDE_6	minADE_6	minFDE_6	minADE_6	minFDE_6
Waymo Open Motion	SeNeVa	1.22	2.63	**1.09**	**2.36**	10.7%	10.3%
	DeMo	1.53	3.94	**1.33**	**3.47**	13.1%	11.9%
Argoverse 2	SeNeVa	1.16	2.41	**1.03**	**2.16**	11.2%	10.4%
	DeMo	1.51	3.83	**1.29**	**3.39**	14.6%	11.5%

the training split, so each arm trains on 500 Waymo scenarios and both are evaluated on the same 150 held-out ones; Waymo's sparse interaction annotation is used exclusively to evaluate the graph in Section 5.3, never for training. Argoverse 2 (Wilson et al., 2021) contributes $224{,}896$ scenarios with high-frequency trajectories, local HD maps, and a heavier tail of unusual behavior, so the two pools differ by roughly two orders of magnitude—a limitation rather than a matched replication (Section 6). Cross-source coverage on nuScenes and simulation logs is in Appendix B.

Protocol and baseline. Every comparison is at matched budget: both arms select exactly $|S| = 0.5N$ scenarios from the same pool, the CuraScene arm by Algorithm 1 with τ_c calibrated to hit that count, the baseline arm by uniform sampling without replacement. The two sets then train the same architecture with the same learning rate, batch size, optimizer, and schedule, and are evaluated on the same held-out split; nothing but the identity of the selected scenarios differs. Uniform random subsampling is our only comparison point—a weaker set of baselines than a coreset paper would normally carry, which Section 6 treats as a limitation rather than explaining away.

Models. We evaluate two forecasting models with different inductive biases, both used without architectural modification so that any difference is attributable to the data: SeNeVa (Lu et al., 2024), a generative model over future trajectories that represents predictive uncertainty explicitly, and DeMo (Zhang et al., 2024), which decouples prediction into mode and state queries and represents the query-based family that currently dominates the benchmarks (Zhou et al., 2023). The minADE_6 values in Table 1 are higher than either model reports originally, because both arms train on half the data under a reduced schedule on a single RTX 3090; we claim nothing about absolute accuracy, only the relative difference between two arms identical except for scenario selection.

5.2 Structure-Guided Selection at Matched Budget

Across both datasets and both models, training on CuraScene-selected scenarios reduces minADE_6 by 10.7 to 14.6% and minFDE_6 by 10.3 to 11.9% relative to spending the same budget at random (Table 1). The effect is consistent in sign and similar in magnitude across four combinations of dataset and architecture, which is the main reason we read it as a property of the data rather than an interaction with a particular model.

Two patterns are worth naming. DeMo gains more than SeNeVa on both datasets while having the higher absolute error; one reading is that SeNeVa's explicit uncertainty modeling already down-weights uninformative samples internally, duplicating part of what structural selection provides, but we offer that only as a hypothesis. The improvement is also slightly larger on Argoverse 2, which would fit a larger pool holding more redundancy to remove, but the two pools differ by a factor of more than two hundred and in every other respect as well, so we do not attribute the gap to pool size. The Waymo arms train on 500 scenarios, and that result should be read as a check that the effect reproduces on a second corpus rather than as independent evidence of its size.

Table 2: Ablation on interacting-agent identification. The reference set for a scenario is the union of its `objects_of_interest` and `tracks_to_predict` fields—a proxy for interaction relevance, not interaction ground truth—and the prediction is the vertex set of the recovered ego-centric subgraph, excluding the ego node. *w/o prior* removes the admissibility mask M; *w/o learned graph* keeps only the heuristic prior. Entries are per-scenario set metrics averaged over the 150 held-out validation scenarios (Appendix D); higher is better.

	Precision	**Recall**	**F1**	**Jaccard**
CuraScene (full)	**0.9733**	**0.7067**	**0.8045**	**0.7067**
w/o prior M	0.3459	0.7026	0.4191	0.2802
w/o learned graph	0.5460	0.7044	0.5991	0.4502

5.3 Is It the Graph?

A selection pipeline can improve on random for reasons unrelated to its stated mechanism. To check that the recovered structure is doing the work, we evaluate it against Waymo's interaction annotations and ablate the two components that produce it (Table 2). Waymo does not label interactions as edges: `objects_of_interest` and `tracks_to_predict` are *sets of agents*, the first flagged as engaging in notable interactions and the second designated for prediction on grounds that include difficulty and agent-type balance (Ettinger et al., 2021). Neither field is an interaction graph, and `tracks_to_predict` is not chosen for interaction alone, so we treat their union as a *proxy for interaction relevance* rather than as interaction ground truth, and read this table as agreement with that proxy rather than as accuracy against a true structure. Scoring is accordingly at the level of vertices: the reference set for a scenario is the union of its two annotated fields, the predicted set is the agents appearing as vertices of the recovered ego-centric subgraph excluding the ego node itself, and precision, recall, F1, and the Jaccard index are ordinary set metrics between the two, computed per scenario and averaged. What the table measures is therefore whether the method selects the agents the benchmark treats as relevant, not whether it orients the influence between them; no public corpus labels direction, a gap we return to in Section 6. The structure model and its ablated variants are fit on Waymo's 1,000 training scenarios and scored on its 150 validation scenarios.

Both components are necessary and they fail in different ways. Removing the admissibility prior and letting the encoder range over all entity pairs collapses precision from 0.97 to 0.35 while leaving recall essentially unchanged at 0.70: the unconstrained model still reaches the relevant agents but surrounds them with agents that cannot plausibly interact with the ego vehicle at all, which is the failure that would undermine the selection criterion, since an indiscriminate neighborhood makes every ego-centric subgraph look alike. Keeping only the heuristic prior and removing the learned graph recovers some precision, to 0.55, but no more: the prior admits a pair whenever the two agents come close enough while at least one is in motion, so a large share of the agents it retains merely pass near the ego vehicle without ever influencing it, and pruning those is exactly what the prior cannot do. The full method reaches 0.97 precision at the same recall, and F1 rises from 0.42 and 0.60 to 0.80.

The near-constant recall of roughly 0.70 across all three variants is the most telling entry in the table: the ceiling is set by the admissibility prior's coverage rather than by the learned component, so the 30% of referenced agents the method misses are ones the prior never admitted as candidates. Widening the prior would trade precision for recall; we did not explore that trade-off.

5.4 Cost

Curation is cheap relative to the training run it is meant to shorten. On a single machine with a 24-core CPU and one RTX 3090, and counting graph inference and selection over the full pool, Argoverse 2's 224,896 scenarios are processed in 47,520 seconds, or 0.21 seconds per scenario; the 1,150 Waymo scenarios take 7,092 seconds, a higher 6.17 seconds each that is dominated by decoding overhead in the source SDK rather than by inference or selection. Curating Argoverse 2 therefore costs roughly thirteen machine-hours once, against the training cost saved on every subsequent run at half the data. Appendix E inspects the recovered structures at three levels of scene

density, including the dense case in which the reduction from eight surrounding agents to three is what keeps Equation 1 discriminative.

6 Discussion and Limitations

The evidence rests on one budget and two incommensurable pools. Every number in Table 1 is measured at 50%, which is the most important gap in the paper: one budget cannot separate the claim we want to make—that structural coverage is a better way to spend any budget—from the weaker fact that one particular half of each corpus trains better than a random half. A sweep would test two sharp predictions, that the advantage is largest at small budgets and vanishes as the budget approaches the full pool; we did not run it and do not claim its outcome. The two pools are also of very different size, Argoverse 2 contributing $224{,}896$ scenarios against Waymo's $1{,}000$, so the Waymo arms train on 500 scenarios and are scored on 150; that result shows the effect appearing in a second and very different data regime rather than replicating the first.

Random is the only selection baseline. Random is a defensible reference point rather than a straw man (Guo et al., 2022), but it does not answer whether *interaction* structure specifically beats generic diversity. The decisive comparison is k-center selection over trajectory or scene embeddings (Sener & Savarese, 2018): were a generic embedding-space criterion to match our numbers, the appeal to interaction structure would be unnecessary and our central argument would not survive. That is the experiment most likely to falsify the paper, and its absence is a real weakness rather than a formality.

What the graph evaluation establishes, and about which agent. Table 2 scores agents against a proxy, not edges against a label: Waymo publishes no interaction graph, and the union of `objects_of_interest` and `tracks_to_predict` indicates which agents the benchmark treats as relevant rather than who influences whom. That proxy is narrow, so a recovered agent it omits counts as a false positive even when the agent genuinely influences the ego vehicle, which makes the reported precision conservative; and direction, the property we argue matters most, is validated by no table here, its evidence being qualitative (Figure 5b) and indirect, through the downstream gain. A second mismatch compounds the first: we select on structure centered on the recording vehicle, whereas the benchmarks score the agents they designate for prediction, and in Argoverse 2 the focal track is not the recording vehicle at all. The two coincide only when the scored agents fall inside the ego-centric neighborhood, which we expect to hold often but did not measure, so Table 1 is evidence that structure around the recording vehicle is a useful selection signal, not that it is the right structure to center on.

Scope. We prove nothing about identifiability for Equation 2: unobserved common causes can induce edges between agents that do not influence each other, and the recall ceiling of Section 5.3 means interactions the prior never proposes are unrecoverable at any setting of the model. The normalization in Equation 1 also makes the novelty score asymmetric, so the guarantees that hold for greedy dispersion under a metric do not transfer and the rule is an order-dependent heuristic; Appendix C records the remaining limits on what we evaluated.

7 Conclusion

We argued that redundancy in driving corpora is structural, and that a scenario's value to a forecasting model is largely determined by the pattern of predictive influence among its agents. CuraScene makes that pattern explicit by inferring a directed interaction graph per scenario without supervision, then selects by the topological novelty of the ego-centric structure under a fixed budget, improving minADE_6 by 10.7–14.6% over budget-matched random sampling for two architectures on two corpora. The contribution is the formulation and the evidence rather than a new structure learning algorithm; whether structural coverage is the right surrogate in general, rather than at one budget on two datasets, is what the budget sweep of Section 6 would answer.

AI USE STATEMENT

Under the ICLR AI policy we declare use of generative AI for research ideation or execution, with details described in the paper: the schema-unification layer documented in Appendix A places a large language model inside the method itself, where it proposes per-field correspondences between source schemas and the intermediate representation. Those proposals are constrained to schema-declared fields, checked by deterministic validators for type, unit, and required-field consistency, and reviewed by the authors before any mapping is used; no model output enters the data path unvalidated. We additionally used generative AI tools to assist with drafting and editing the text of this paper. All AI-assisted text and all AI-proposed schema mappings were reviewed by the authors, who take responsibility for the final content of this work, including its claims, text, and artifacts.

ETHICS STATEMENT

This work uses three public driving datasets under their published licenses and simulation logs generated locally; no new data involving human subjects was collected, and no personally identifying information is used beyond what the source datasets already release in anonymized form. The intended effect of the method is to reduce the data and compute required to train forecasting models, which lowers the energy cost of a training run but also lowers the barrier to deploying such models. One risk deserves naming directly. A curation method that discards scenarios is a method that decides what a safety-relevant model never sees, and a systematic failure of the interaction-inference stage would remove a class of scenarios silently rather than visibly. The recall ceiling documented in Section 6 is exactly this kind of failure: interactions the admissibility prior never proposes are invisible to the selection rule, and the scenarios containing them can be pruned as structurally redundant. We therefore regard CuraScene as appropriate for allocating a training budget and not as a filter for safety validation suites, where coverage rather than diversity is the requirement.

REPRODUCIBILITY STATEMENT

The selection rule is stated in full as Algorithm 1, the structure-inference objective as Equation 4, and the novelty score as Equation 1. Appendix A specifies the intermediate representation in Backus–Naur form together with the three per-field operations and the validation checks applied to every mapping, which is what is needed to reproduce the data-preparation path from raw Waymo, Argoverse 2, or nuScenes releases. Appendix D lists architectures, hyperparameters, hardware, and the training configuration used for both downstream models; both are trained from their official implementations with unmodified default configurations, changed only in the identity of the training scenarios. Appendix B reports exactly which fields each source can populate, so that a reader can determine in advance whether a new corpus is usable. The raw Waymo and Argoverse 2 data cannot be redistributed under their licenses and must be obtained from the official sources.

REFERENCES

David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 859–868, 2018.

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. *arXiv preprint arXiv:1903.11027*, 2019.

Mingfei Cheng, Yuan Zhou, and Xiaofei Xie. Behavexplor: Behavior diversity guided testing for autonomous driving systems. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 488–500, 2023.

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 *International Conference on Learning Representations (ICLR)*, 2020.

Yao Deng, Xi Zheng, Mengshi Zhang, Guannan Lou, and Tianyi Zhang. Scenario-based test reduction and prioritization for multi-module autonomous driving systems. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 82–93, 2022.

Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):6971–6988, 2023.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, Zoey Yang, Aurelien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. *arXiv*, 2021.

Shiwei Feng, Yapeng Ye, Qingkai Shi, Zhiyuan Cheng, Xiangzhe Xu, Siyuan Cheng, Hongjun Choi, and Xiangyu Zhang. Rocas: Root cause analysis of autonomous driving accidents via cyber-physical co-mutation. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pp. 1620–1632, 2024.

Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. VectorNet: Encoding HD maps and agent dynamics from vectorized representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. DeepCore: A comprehensive library for coreset selection in deep learning. In *Database and Expert Systems Applications (DEXA)*, volume 13108 of *Lecture Notes in Computer Science*, pp. 181–195. Springer, 2022.

Yuqi Huai, Yuntianyi Chen, Sumaya Almanee, Tuan Ngo, Xiang Liao, Ziwen Wan, Qi Alfred Chen, and Joshua Garcia. Doppelgänger test generation for revealing bugs in autonomous driving software. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 2591–2603. IEEE, 2023.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Krishnateja Killamsetty, S Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning (ICML)*, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *AAAI Conference on Artificial Intelligence*, 2021b.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning (ICML)*, 2018.

Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *Advances in neural information processing systems*, 33:19783–19794, 2020.

Minghua Liu, Yin Zhou, Charles R Qi, Boqing Gong, Hao Su, and Dragomir Anguelov. Less: Label-efficient semantic segmentation for lidar point clouds. In *European conference on computer vision*, pp. 70–89. Springer, 2022.

Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C Knoll. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. *IEEE Transactions on Intelligent Vehicles*, 2024.

Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Conference on Causal Learning and Reasoning*, pp. 509–525. PMLR, 2022.

Juanwu Lu, Can Cui, Yunsheng Ma, Aniket Bera, and Ziran Wang. Quantifying uncertainty in motion prediction with variational bayesian mixture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15428–15437, 2024.

Yadan Luo, Zhuoxiao Chen, Zijian Wang, Xin Yu, Zi Huang, and Mahsa Baktashmotlagh. Exploring active 3D object detection from a generalization perspective. In *International Conference on Learning Representations (ICLR)*, 2023.

Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33016120. URL https://doi.org/10.1609/aaai.v33i01.33016120.

Osama Makansi, Özgün Çiçek, Yassine Marrakchi, and Thomas Brox. On exposing the challenging long tail in future prediction of traffic actors. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13147–13157, 2021.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning (ICML)*, 2020.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Davis Rempe, Jonah Philion, Leonidas J. Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 27(7):950–959, 2009.

Luke Rowe, Martin Ethier, Eli-Henry Dykhne, and Krzysztof Czarnecki. Fjmp: Factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13745–13755, 2023.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision (ECCV)*, 2020.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Huijia Sun, Christopher M. Poskitt, Yang Sun, Jun Sun, and Yuqi Chen. Acav: A framework for automatic causality analysis in autonomous vehicle accident recordings. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400702174. doi: 10.1145/3597503.3639175. URL https://doi.org/10.1145/3597503.3639175.

Qiao Sun, Xin Huang, Junru Gu, Brian C Williams, and Hang Zhao. M2i: From factored marginal trajectory prediction to interactive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6543–6552, 2022.

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B. Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2021.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning (ICML)*, 2015.

Max Welling. Herding dynamical weights to learn. In *International Conference on Machine Learning (ICML)*, 2009.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

Bozhou Zhang, Nan Song, and Li Zhang. Decoupling motion forecasting into directional intentions and dynamic states. *Advances in Neural Information Processing Systems*, 37:106582–106606, 2024.

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8823–8833, 2022.

Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

A SCHEMA UNIFICATION

This appendix documents the schema-unification layer named in Section 4. It is infrastructure: it makes heterogeneous corpora usable by the rest of the method and contributes nothing to the selection criterion. A reader interested only in the selection result can skip it.

A.1 WHY A UNIFIED REPRESENTATION IS NEEDED

The corpora we draw on describe the same underlying entities—agents with kinematic state, a road network, traffic control—but agree on almost nothing about how to write them down. nuScenes (Caesar et al., 2019) uses a relational, database-style organization; Waymo (Ettinger et al., 2021) and Argoverse 2 (Wilson et al., 2021) use different hierarchical layouts. Representation drift reaches individual attributes: nuScenes encodes orientation as a quaternion while the others use a yaw angle in radians, and lane speed limits appear in miles per hour in one source and metres per second in another. Table 3 summarizes the differences. Without a shared representation, adding a source means writing an adapter against every consumer, and the engineering cost grows quadratically in the number of sources rather than linearly. The geographic spread of the corpora makes this worth paying for: Waymo is concentrated in West Coast US cities and Argoverse 2 in Midwestern and Eastern ones, so combining them is how one obtains variation in road layout and driving convention rather than merely more samples from one distribution.

Table 3: Structural features of representative forecasting corpora. No public corpus supplies directed interaction labels, and only Waymo supplies any interaction annotation at all—and then as sets of interacting agents rather than as edges—which is why the structure in Section 4.1 has to be inferred rather than read off.

			Agent states & traj.				**Static env.**		**Dynamic env.**		
Dataset	**Yr**	**Format**	Type	Size	Kine.	Traj.	HD map	Lane conn.	TL states	Cplx. int.	Int. labels
ApolloScape (Ma et al., 2019)	2019	txt/csv	✓	✓	✓	✓	✗	✗	✗	✗	✗
nuScenes (Caesar et al., 2019)	2020	json (rel.)	✓	✓	✓	✓	✓	partial	✗	✗	✗
Argoverse 2 (Wilson et al., 2021)	2021	parquet	✓	✗	✓	✓	✓	✓	✗	✓	✗
Waymo (Ettinger et al., 2021)	2021	tfrecord	✓	✓	✓	✓	✓	✓	✓	✓	partial
CARLA / LGSVL	–	json/log/rosbag	✓	✓	✓	✓	✓	custom	custom	custom	✗

A.2 THE INTERMEDIATE REPRESENTATION

```
       <scenario> ::= <agents> <map> <traffic_signals>
         <agents> ::= [<agent>]
          <agent> ::= <attributes> <states>
     <attributes> ::= <id> <agent_type> <size>
         <states> ::= [<state>]
          <state> ::= <position> <heading> <velocity>
            <map> ::= <roadlines> <lanes> <stopsigns> <crosswalks>
      <roadlines> ::= [<roadline>]
       <roadline> ::= <id> <type> <polyline>
          <lanes> ::= [<lane>]
           <lane> ::= <id> <type> <centerline> <predecessors>
                      <successors> <neighbors> <roadline_refs>
      <stopsigns> ::= [<stopsign>]
       <stopsign> ::= <id> <position> <controlled_lanes>
     <crosswalks> ::= [<crosswalk>]
      <crosswalk> ::= <id> <polygon>
<traffic_signals> ::= [<signal_state>]
   <signal_state> ::= <controlled_lanes> <signal_type>
```

Listing 1: The CuraScene intermediate representation, in Backus–Naur form.

We standardize every source into the semantic tuple

$$\mathcal{U} = \langle \mathcal{A}, \mathcal{K}, \mathcal{M}, \mathcal{T} \rangle, \tag{5}$$

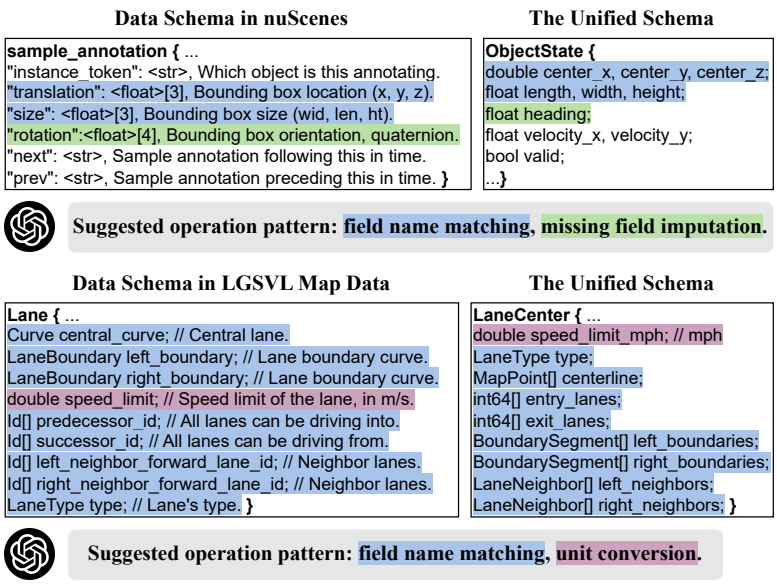


Figure 3: The three per-field operations in practice: name matching, affine unit conversion, and imputation of a field that is absent but derivable.

with $\mathcal{A}$ agent attributes, $\mathcal{K}$ kinematic states, $\mathcal{M}$ static map elements, and $\mathcal{T}$ dynamic traffic signals. Listing 1 gives the representation in Backus–Naur form. We specify it as a grammar rather than by example deliberately: the mapping step below is driven by a language model, and an explicit machine-readable grammar measurably reduces misinterpretation of deeply nested structures compared with a worked example.

Agents carry an identifier, a semantic type, physical dimensions, and a time-indexed sequence of position, heading, and velocity expressed in the scenario's map frame. Map elements carry geometry and semantics together—lane type, centerline, predecessor and successor relations, neighbouring lanes, and the road lines between them—because the interaction prior of Section 4.1 needs lane membership to decide which agents a traffic signal controls. Signal states are indexed by timestep and reference the lanes they govern.

A.3 Two-Hop Instance Mapping

Transformation proceeds in two hops, from the source schema $\mathcal{S}_s$ to the intermediate representation $\mathcal{S}_{ir}$ and then to whatever target schema a consumer expects:

$$x_t = f_{ir\to t}\big(f_{s\to ir}(x_s)\big). \tag{6}$$

Each hop is composed from a fixed operation set $\mathcal{O} = \{\texttt{match}, \texttt{convert}, \texttt{impute}\}$. For each target field y_j we predict a source field index $\pi(j)$ and an operation $o_j \in \mathcal{O}$, and apply $y_j = o_j(x_{\pi(j)})$. Figure 3 shows an example of each.

Name matching. Semantically equivalent fields are aligned across naming conventions, for instance the nuScenes `translation` list to the triple `(centerX, centerY, centerZ)`.

Unit conversion. Numeric fields are normalized by an affine map $x' = \alpha x + \beta$ with (α, β) determined by the source and target unit pair, which preserves physical meaning and prevents scale mismatch downstream.

Imputation. Missing values are filled temporally or semantically. Asynchronous timestamps are resolved by linear interpolation,

$$\hat{\mathbf{s}}(t) = \mathbf{s}(t_0) + \frac{t - t_0}{t_1 - t_0}\big(\mathbf{s}(t_1) - \mathbf{s}(t_0)\big), \quad t \in [t_0, t_1], \tag{7}$$

and fields that are required but absent are derived where the information exists in another form—nuScenes has no explicit heading, but one follows from the quaternion by a standard transformation.

A.4 Language-Model Assistance and Its Controls

Authoring the per-field correspondences by hand for each new source is the step we automate. A language model receives the source, intermediate, and target schema definitions, together with any field descriptions or examples the source documents, and proposes correspondences and operation types for both hops. The prompt is organized into four blocks: schema context; the constraint that only `match`, `convert`, and `impute` may be used; an output contract requiring JSON field mappings with confidence scores; and self-check instructions covering unit consistency, required-field coverage, and type compatibility.

The controls around it matter more than the prompt. Generation is constrained to schema-declared fields, mappings with unresolved references are rejected outright, and deterministic post-checks verify dimensional validity and the presence of every mandatory key. Developers review only the mappings that are low-confidence or that violate a rule, and their corrections are fed back as in-context exemplars for subsequent sources. The model's output is a transformation template that is checked before use, never data that flows through unvalidated. As noted in Section 6, this component is non-deterministic across runs and model versions, and the validators rather than the model are what make the pipeline reproducible.

B Cross-Source Coverage

Section 5 runs on Waymo Open Motion and Argoverse 2. This appendix reports whether the same pipeline runs unchanged on two further sources, nuScenes (Caesar et al., 2019) and simulation logs generated with Apollo under an AV testing engine (Huai et al., 2023), and which fields of the intermediate representation each source can populate. The question matters for the paper's argument because a selection criterion that only works on the corpus it was developed on is of little use, and because the absence of interaction labels in three of the four sources is the reason the structure has to be inferred at all.

Sources and scale. All counts below are scenarios. We process $1{,}150$ Waymo scenarios sampled from the official training and validation splits, the full $224{,}896$ scenarios of the Argoverse 2 training and validation splits, and the 850 scenes of the nuScenes `trainval` release, together with 908 simulation test cases, for a total of $227{,}804$ scenarios. Only the Waymo figure is a sample rather than a complete split, and it is the same $1{,}150$ scenarios used in Section 5.1. Every scenario in all four sources was structured into the intermediate representation without failure for the fields each source supports.

What the gaps mean. Table 4 shows full coverage for Waymo and specific gaps elsewhere, and in every case the gap is a property of the source rather than of the representation. Argoverse 2 and nuScenes publish no stop-sign annotations, so stop-sign control semantics cannot be populated from them. Argoverse 2 publishes no traffic-signal states and nuScenes only incomplete lane–signal associations, which limits the dynamic control-lane fields. The simulation logs do not type road lines finely enough to distinguish, say, a solid white line from a broken yellow one. None of these gaps prevents interaction inference, which needs only agent kinematics and those are complete in all four sources; they determine only whether the resulting graph can be enriched with signal nodes, which among these sources Waymo alone permits. This is the practical consequence of separating source parsing from the downstream stages: adapting to a new corpus requires a source adapter and a set of field mappings, and leaves the inference and selection code untouched.

Why simulation data is excluded from the downstream experiment. We structure the simulation logs but do not use them to train the models in Table 1. The reason is a sampling bias we could

Table 4: Field-level coverage of the intermediate representation across four sources, where ✓ is full support, ◯ limited support, and ✗ no support. All rows above the final block are fields read from the source, and their gaps reflect what the source annotates rather than what the representation can express. The final row is not a source field but a pipeline output: an interaction graph is produced for every source, because agent-only inference depends only on the agent kinematics that all four supply, and the signal rows determine only whether that graph can additionally be enriched with signal nodes.

		Waymo	**Argoverse 2**	**nuScenes**	**Simulation**
Agent states & trajectories	Type	✓	✓	✓	✓
	Size	✓	✗	✓	✓
	Kinematics	✓	✓	✓	✓
	Trajectory	✓	✓	✓	✓
Static env. elements	Lane	✓	✓	✓	✓
	Road line	✓	✓	✓	◯
	Stop sign	✓	✗	✗	✓
	Crosswalk	✓	✓	✓	✓
Static env. semantics	Geometry	✓	✓	✓	✓
	Controlled lane	✓	✗	✗	✓
	Topology	✓	✓	✓	✓
Dynamic env.	Signal states	✓	✗	◯	✓
	Controlled lane	✓	✗	✗	✓
Output	Interaction graph	✓	✓	✓	✓

not correct: generating diverse safety-relevant scenarios in simulation is itself an open problem, and the pool we collected is dominated by nominal free-flow driving with little interaction, with rare events underrepresented. Training on it would inflate performance on easy cases and say little about the long tail, so including it would make the matched-budget comparison easier to win and harder to interpret.

C Scope of the Evaluation

Three further restrictions bound what the experiments in Section 5 show, beyond the limits recorded in Section 6. We evaluate two forecasting models on two corpora under single-agent displacement metrics, and not planning, closed-loop driving, or the VLM- and VLA-based systems (Wang et al., 2025; Hwang et al., 2024) whose appetite for data is the strongest motivation for the problem; those were beyond the compute available to us. The simulation logs are structured but excluded from the downstream comparison for the sampling reason given above, so the cross-source claim of this appendix concerns coverage of the representation and not training value. Finally, the schema layer places a language model in the data path, and its outputs are non-deterministic across runs and model versions, which is why every mapping is validated by deterministic post-checks rather than trusted (Appendix A).

D Implementation Details

Hardware and software. All stages are implemented in Python 3 against the official SDKs of the source corpora, including the Waymo Open Dataset API and nuScenes-devkit. Structuring, graph inference, and selection run on a single machine with Ubuntu 20.04, a 24-core Intel i9-12900K, one NVIDIA RTX 3090 with 24 GB of memory, and 64 GB of system RAM. The same machine trains the downstream models, which is the binding constraint behind the reduced training schedule discussed in Section 5.1 and the absence of VLM- and VLA-scale experiments.

Downstream models. SeNeVa (Lu et al., 2024) and DeMo (Zhang et al., 2024) are trained from their official PyTorch implementations with their published configurations and without architectural modification. Learning rate, batch size, optimizer, and schedule are held fixed across the CuraScene

and random arms so that no part of the measured difference can be attributed to tuning. We did not tune hyperparameters for either arm, which means both arms are equally disadvantaged relative to the published full-data settings.

Structure inference. The encoder and the decoder are graph networks with three layers and 128 hidden units. The input lag is $L = 10$, and the decoder predicts the next 20 frames. The admissibility prior admits an agent pair when the two come within $d_{\max} = 30\,\mathrm{m}$. Training uses KL weight $\beta = 0.5$, prior edge probability $\rho = 0.8$, and sparsity weight $\lambda = 10^{-2}$. Recovered edges are those whose weight exceeds $\tau_e = 0.2$, and the ego-centric subgraph keeps nodes within hop radius $r = 2$. The structure model is trained with Adam at learning rate 10^{-3}, batch size 64, for 100 epochs.

Selection. Algorithm 1 states the selection rule of Section 4.2 in the form we run it. Each scenario is converted to an interaction graph and canonicalized, and exact or canonical duplicates are removed before any distance is computed. Novelty is the normalized graph edit distance of Equation 1, measured against graphs already retained. A corpus-specific τ_c is calibrated so that about half the scenarios are retained, then applied to the corpus; each newly accepted graph is added to the retained set before the next comparison.

Algorithm 1: Structure-guided scenario selection

Input: pool $\mathcal{D}$
Output: selected subset S
$\mathcal{G} \leftarrow \emptyset$
foreach $x_i \in \mathcal{D}$ **do**
  $g_i \leftarrow \textsc{InteractionGraph}(x_i)$ `// convert to an interaction graph`
  $g_i \leftarrow \textsc{Canonicalize}(g_i)$
  $\mathcal{G} \leftarrow \mathcal{G} \cup \{g_i\}$
end
$\mathcal{G} \leftarrow \textsc{DropDuplicates}(\mathcal{G})$ `// exact and canonical duplicates`
$\tau_c \leftarrow \textsc{Calibrate}(\mathcal{G})$ `// corpus-specific, about 50% retention`
$S \leftarrow \emptyset$
foreach $g_i \in \mathcal{G}$ **do**
  $\Delta \leftarrow \min_{g_j \in S} \mathrm{GED}(g_i, g_j)/(|V_i| + |E_i|)$, taking $\Delta = \infty$ if $S = \emptyset$ `// Eq. 1`
  if $\Delta > \tau_c$ **then**
    $S \leftarrow S \cup \{g_i\}$ `// incremental curation`
  end
end
return S

Ablation metrics. Precision, recall, F1, and the Jaccard index in Table 2 are computed independently for each of the 150 Waymo validation scenarios and then averaged over scenarios. The tabulated F1 is therefore a mean of per-scenario F1 scores, which is why it does not equal the harmonic mean of the tabulated precision and recall: for the full method $2PR/(P{+}R) = 0.819$ against the 0.8045 reported, and for *w/o prior* 0.464 against 0.4191. Scenario-level averaging weights every scenario equally regardless of how many agents it contains, which is the appropriate choice here because selection operates per scenario. The precision equals $146/150$: the mean is the average of 146 scenarios at precision 1 and 4 at precision 0.

Evaluation. Both arms are scored with minADE_6 and minFDE_6 under the standard top-6 protocol. Each cell of Table 1 is already an aggregate over repeated runs.

E Qualitative Cases in Detail

Figure 5 shows recovered ego-centric structures at three levels of scene density, and this appendix discusses each of them together with the annotated structure in Figure 4, which illustrates the partition of agents assumed by the selection criterion of Section 3.

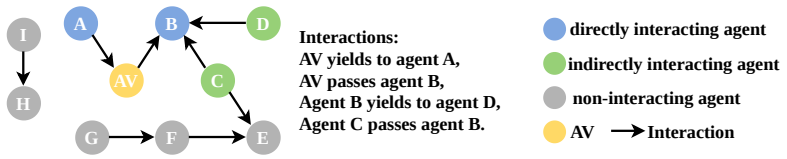
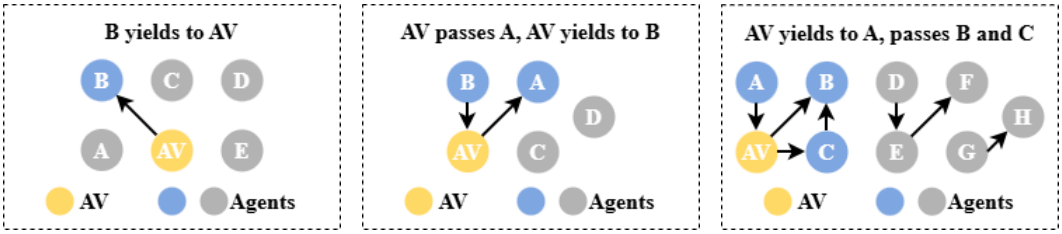


Figure 4: An ego-centric interaction structure. Agents are partitioned into direct influencers of the ego vehicle, indirect influencers that act through an intermediary, and non-interacting bystanders that the selection criterion ignores. Edge direction distinguishes yielding from passing.



(a) single yield (b) competing constraints (c) dense scene

Figure 5: Recovered ego-centric structures at increasing scene density. Yellow is the ego vehicle, blue marks agents retained as influencing it, grey marks agents the graph discards. In (c) the ego vehicle is surrounded by eight agents and three survive, which is the reduction that makes the structural distance in Equation 1 discriminative.

(a) Single yield. Five agents are present and one of them, agent B, yields to the ego vehicle. The recovered graph contains a single directed edge, and agents C, D, and E are discarded because neither their motion nor their position renders them capable of influencing the ego trajectory over the horizon. This is the atomic case, and it is also the most common non-trivial structure in both corpora. It illustrates why the ego-centric restriction in Section 3 matters: the full scenario graph contains edges among the discarded agents as well, but those edges do not change what the forecasting model must learn about the ego vehicle.

(b) Competing constraints. The ego vehicle passes agent A while simultaneously yielding to agent B. Two edges survive with opposite orientation relative to the ego node, and this is the case that makes direction load-bearing. An undirected representation would assign this scenario the same structure as one in which the ego vehicle yields to A and passes B, even though the two demand opposite behavior from the model. Because the novelty score in Equation 1 is computed over directed subgraphs, the two are distinguished and both can enter the selected set.

(c) Dense scene. Eight agents surround the ego vehicle and three survive: the ego vehicle yields to A and passes B and C, while D through H are classified as bystanders. Case (c) is the one that determines whether selection works at all. If the graph retained all eight agents, the ego-centric subgraph of nearly every dense urban scenario would be large and near-complete; all such subgraphs would sit at comparable graph edit distances from one another, the minimum in Equation 1 would vary little across candidates, and the threshold rule would either admit everything or nothing. Sparsification is therefore not a cosmetic step but the reason the distance is informative, which is the same argument made for the ℓ_1 penalty in Section 4.1.

Direct and indirect influence. Figure 4 shows a case with a two-hop chain: agent B yields to agent D, and B in turn influences the ego vehicle, so D affects the ego trajectory only through B. Whether such indirect influencers should be included in g_i is controlled by the hop radius r. Setting $r = 1$ keeps only direct influencers and produces the smallest, most repetitive structures; larger r captures chains of the kind that make dense intersections difficult but increases the fraction of scenarios that appear structurally distinct. Sweeping r is the cheapest remaining ablation, since it changes only the subgraph extraction and requires no retraining of the structure model; we did not run it.