Reading Vehicle Size and Powertrain from Motion: A Trajectory–CAN Benchmark with Held-Out Car Models, Pre-Registered Methods and Label Ceilings

Yuhang Wang, Yuechen Guo, Jingxin Yang, Chuheng Wei, Suyang Chen,
Abdulaziz Alhuraish, Yiyi Wang, Jingran Sun and Hao Zhou

***Abstract*— Trajectory data for traffic analytics, simulation calibration and infrastructure studies rarely record the vehicle type. We ask whether it can be read from motion alone: can 60-s windows of trajectory and CAN signals reveal the size class (three footprint classes) and the powertrain (battery-electric vs combustion, BEV vs ICE) of car models never seen in training? We release a pseudonymous dataset of 9,062 drives (about 3,950 h) from 1,431 vehicles and 174 car models, with held-out-car-model and held-out-device splits, window-, drive- and vehicle-level protocols and cluster-bootstrap intervals over car models. Four sequence architectures (ResNet, bi-GRU, Transformer, Mamba-2), hand-crafted statistics and physical features proposed by large language models (LLMs) serve as reference points; a pre-registered method study on model-disjoint internal folds adds car-model-balanced sampling, auxiliary physical targets, physics-safe augmentation, per-car-model loss re-weighting and, for BEV vs ICE, down-weighting of driving automation system (DAS) periods. On held-out car models, BEV vs ICE reaches 92.6–93.0 drive-level macro-F1 (95% interval [87.1, 96.2] for the final bi-GRU); size improves from 62.9 to 67.1 at drive level but stays far from 90. Zero-training oracle analyses attribute the gap to label resolution and training-fleet composition rather than to evidence per drive; the architecture benchmark shows model capacity is not the bottleneck.**

I. Introduction

Vehicle trajectories are collected at scale by roadside sensors, connected-vehicle platforms and aftermarket loggers, usually without a reliable record of the vehicle that produced them. Many uses need it: traffic analytics by vehicle class, calibration of microscopic simulation models whose car-following parameters differ between a compact car and a three-row SUV, and infrastructure studies of how often heavy or electric vehicles use a facility. Registration records, licence plates or camera images can fill the gap but raise privacy and availability problems; a trajectory that reveals its vehicle type by itself would be more privacy-friendly.

Motion should carry this information: wheelbase, steering ratio and mass distribution set the yaw and curvature response to steering, and the powertrain shapes launches and lift-off deceleration, so a battery-electric vehicle (BEV) braking largely through its motor should leave a different speed trace from an internal-combustion-engine (ICE) vehicle. We pose the question as it arises in deployment, for car models never seen in training, since otherwise every new model would need retraining. From 60-s windows of trajectory and controller area network (CAN) signals (among them speed, acceleration, steering, yaw rate, curvature, lateral acceleration, pedal and automation flags, and lead-vehicle distance, relative speed, headway and time-to-collision; Section III lists all channels) we predict a size class (SMALL, MID or LARGE, cut from the vehicle footprint) and the powertrain (BEV vs ICE).

Three things make this hard. First, the size label is a hard cut through a continuous quantity (footprints of 4.43 and 4.88 m^2 separate the classes), so a car model near a cut is hard to place even when its physics is read well. Second, a driving automation system (DAS) is engaged in 47.9% of the samples, so part of the measured behaviour is a controller signature that may not transfer between car models. Third, the fleet is uneven (174 car models from 38 brands with widely differing hours), so a held-out model in a footprint gap of the training fleet has no neighbour to be read against.

We therefore release the data with fixed labels and splits, benchmark sequence models and feature pipelines under one protocol, run a pre-registered method study on model-disjoint folds, and ask, without training anything, where the size ceiling lies and why. The contributions are:

- A public, privacy-preserving trajectory and CAN dataset (9,062 drives, about 3,950 h, 1,431 pseudonymous vehicles) with fixed footprint-size and powertrain labels, held-out-car-model (headline) and held-out-device splits, and window-, drive- and vehicle-level protocols with cluster-bootstrap intervals over car models (Section III).
- A benchmark of four sequence architectures, hand-crafted statistics and physical features proposed by a large language model (LLM) under those protocols, showing that architecture is not the bottleneck (all deep models within about 2 points) and drive-level aggregation adds 7–12 points (Table II).
- A pre-registered method study on model-disjoint internal folds (Sections IV and V): car-model-balanced sampling, auxiliary physical targets, physics-safe augmentation, per-car-model loss re-weighting (size) and DAS down-weighting (BEV vs ICE) raise the drive-level result by 4–6 points (size) and 2–4 points (BEV vs ICE); an episode-centric multiple-instance model over driving transients matches but does not beat the window recipes on the test models.
- Zero-training ceiling analyses (per-car-model majority, footprint-noise and curb-mass oracles, evidence sufficiency, separability of held-out models) that quantify why size cannot reach 90 on unseen car models under a footprint cut, and what would make it reachable (Figs. 1 and 2).

II. Related Work

Electric and combustion vehicles on the road. Battery-electric vehicles accelerate and slow down differently from combustion vehicles: torque is available from standstill, much of the deceleration comes from regenerative rather than friction braking, and one-pedal driving lets the car decelerate strongly when the accelerator is released. Instrumented and naturalistic studies report such differences: drivers new to electric vehicles accelerate and brake differently at first and adapt over months [1], electric vehicles under adaptive cruise control follow with smoother speed profiles and shorter spacing than combustion vehicles [2], and car-following models for electric vehicles add phases for strong acceleration and regenerative braking [3]. Most of this work compares populations whose type is known; we ask the inverse question, whether the type can be recovered from the motion of a car model never observed in training.

Vehicle and driver identification from kinematics and CAN. Vehicle classification from GPS trajectories separates passenger cars from trucks with hand-crafted features [4] or convolutional networks [5], and driver identification from CAN signals has shown that steering, pedal and speed traces carry strong identity information, from whole drives [6] down to a single turn [7]. When such pipelines feed summary statistics to a tree ensemble [8], [9] or a small network and split windows or trips at random, the same vehicles or car models appear on both sides, so identity rather than type may be learnt. Our splits hold out car models or devices by construction.

Sequence models for time-series classification. One-dimensional residual networks remain strong baselines for time-series classification [10], [11]; gated recurrent units [12] and Transformer encoders [13] are the usual recurrent and attention-based alternatives, and selective state-space models [14] with the structured state-space duality of Mamba-2 [15] offer linear-time sequence mixing. We benchmark one representative of each family under one protocol.

Shortcut learning and group-disjoint evaluation. Deep networks readily pick up cues that correlate with the label in training but not in deployment [16]. For vehicle-type inference the obvious shortcuts are the identities of the car model and of the logging device, so we evaluate on held-out car models and report cluster-bootstrap intervals over models. Group distributionally robust optimisation [17] and domain-adversarial training [18] are standard remedies for group shift; we test light-weight variants of both on model-disjoint internal folds.

Multiple-instance learning and ordinal targets. A drive is a bag of windows or episodes with a single label, which is the setting of attention-based multiple-instance learning [19]. The size classes are ordered cuts of a continuous footprint, so rank-consistent ordinal regression [20] and direct regression of the footprint are natural auxiliary targets.

Language models for time series and feature discovery. Using large language models directly as time-series forecasters has been questioned [21], while letting them propose programs that are then checked by execution has been fruitful [22]. We follow the second route: agents propose physically motivated features as code, which verifiers and an auditor screen before they enter a standard classifier.

Source logs. The drives come from the open-source driving automation system (DAS) ecosystem behind comma2k19 [23] and a large-scale naturalistic take-over dataset [24]; we expose only vehicle-state and lead-vehicle signals under pseudonyms, without positions, absolute times or video (Section III).

III. Dataset and Protocols

A. Source and Extraction

The drives are open logs of vehicles running an open-source driving automation system (DAS), the ecosystem behind comma2k19 [23] and the take-over dataset of [24]. Drive tables are indexed by the 10 Hz vehicle-state sample (row k at nominal time $k/10$ s). Lower-rate radar, pose and controller topics are zero-order held causally onto these sample times, with stale samples set to missing, and a flag marks rows preceded by a gap longer than 0.3 s. DAS-engaged periods are kept and flagged, so the same vehicles appear under human and automated driving.

B. Channels

Of the 44 columns of a drive table, 20 are the official model inputs: speed and longitudinal acceleration; steering-wheel angle, rate and normalised torque; yaw rate, curvature and lateral acceleration; the flags accelerator pressed, brake pressed, steering pressed, standstill and DAS engaged; and lead-vehicle validity, distance, relative speed, speed, acceleration (a causal slope of its speed), time headway and time to collision. Steering torque is normalised per drive by the 95th percentile of its magnitude while moving and steering. Pedal magnitudes are missing on 63.9% (accelerator) and 49.4% (brake) of the drives, so only the always-valid pressed flags are inputs; on 56.2% of the drives the lead comes from a camera model rather than a radar.

C. Labels

Both labels are attributes of the platform identified from the bus, not of a drive. Size is the footprint, wheelbase $\times$ mean track: SMALL below 4.43 m^2, MID from 4.43 up to 4.88 m^2, LARGE from 4.88 m^2. The thresholds were locked on the label inventory before any classifier was trained: every size $\times$ $\{$BEV, ICE$\}$ cell had to hold at least 100 h, 3 car models and 5 identity groups (same device, same vehicle identity or shared drive names); the admissible pairs maximising the smallest cell (309 h) were kept, each cut placed in a gap between platform footprints. Examples are the Civic and Bolt (SMALL), Camry and Model 3 (MID), Palisade and Model X (LARGE); in that inventory the BEV hours of SMALL, MID and LARGE came from only 15, 10 and 8 car models, against 55, 42 and 28 for ICE. Powertrain is ICE, hybrid (HEV), plug-in hybrid (PHEV) or BEV, with BEV vs ICE the main task; platforms sold with several powertrains that the logs cannot separate (MIXED) or of unknown powertrain carry a null label (11.1% of the drives). Table I summarises the release.

TABLE I

RELEASE SUMMARY. VEHICLES ARE DEVICE × PLATFORM PAIRS; CLASS SUPPORT COUNTS ELIGIBLE 60-S WINDOWS; THE TEST-SPLIT BLOCK COUNTS ELIGIBLE WINDOWS AND THE UNITS THEY COVER.

Drives (one ignition cycle each)	9,062
Hours, nominal	3,949.8
Hours moving	3,078.1
Pseudonymous vehicles	1,431
Identity groups (unit of the splits)	1,157
Platforms / brands / car models	360 / 38 / 174
Samples with DAS engaged	47.9%
Samples with any automation activity	62.6%
Windows listed	241,476
Windows eligible	201,390
Windows human-only (strict)	37,942
Eligible windows, size	201,169
Eligible windows, BEV vs ICE	144,534
Eligible windows, four-class powertrain	178,827

Class	Windows	Hours	Vehicles	Platforms
SMALL	91,369	1,522.8	673	169
MID	56,059	934.3	392	104
LARGE	53,741	895.7	295	76
ICE	79,566	1,326.1	598	200
HEV	23,708	395.1	197	57
PHEV	10,585	176.4	68	21
BEV	64,968	1,082.8	366	71
Powertrain null	22,563	376.1	156	34

Test split, held-out car models	Size	BEV vs ICE
Eligible windows	26,144	18,295
Hours	435.7	304.9
Drives	1,141	836
Vehicles	329	244
Car models	41	39

D. Splits

Both schemes split by identity group, since drives of one device are correlated. The headline scheme, `split_platform`, holds out whole car models, about 15% of the hours of each size × powertrain cell, as the test split (last block of Table I); drives of a group that also drove a held-out model are excluded from every split. The validation split is carved group-disjoint from the remaining pool and shares 53 of its 56 car models with train, so only the test split is model-disjoint and a validation score rewards recognising a known car model (Section IV). The secondary scheme, `split_device`, assigns groups at random inside each cell (70 / 15 / 15% of the hours) and shares car models across splits, so its test measures generalisation to unseen devices of known models; lab-owned devices are always in train.

E. Protocols

A 60-s window is eligible when it has 600 consecutive samples, no gap and at least 20% of its samples moving. Window-level protocols score each eligible test window on its own: `test_full` takes all of them (headline), `test_balanced` a published class-balanced list, and `test_human_only` the windows with no automation activity inside or in the 10 s before them.

Unit-level protocols pool the same per-window probabilities: `drive_full` averages the probability vectors of a drive's `test_full` windows and takes the arg-max, `drive_human_only` does so over human-only windows, and `vehicle_full` pools every window of all test drives of a vehicle; units without a selected window are excluded. The metric is macro-F1 over the classes present. Held-out car models are the replication unit, so 95% intervals come from a cluster bootstrap over car models (1,000 resamples, seed 0): each resample draws car models with replacement and keeps every unit of a drawn model once.

F. Shortcut Audits

Two group-disjoint audits with simple classifiers look for shortcuts [16]: nuisance metadata columns predict powertrain at 0.32 macro-F1 (chance 0.25) and size at 0.40 (chance 0.33), so they stay out of the inputs; window statistics of the official inputs predict brand at 0.02 (chance 0.03 over 38 brands) and platform at 0.00 but powertrain at 0.31 and size at 0.34: no detectable brand fingerprint and little class information under this audit.

G. Privacy and Availability

Vehicles, drives and groups carry random, independent pseudonyms drawn for this release only. The release holds no positions, absolute times or dates, video, images or driver-monitoring signals; region is a continent-level bucket and time is relative to the start of each drive. Access is gated to research on vehicle dynamics and driving behaviour and prohibits re-identification of vehicles or drivers and linkage with other datasets. The data, indexes, split lists, schema and checksums are public under CC BY 4.0 and the build, evaluation and experiment code under the MIT License: `https://huggingface.co/datasets/HenryYHW/Vehicle_Power_Type_Trajectory_Dataset` and `https://github.com/WangYuHang-cmd/-Vehicle_Power_Type_Trajectory_Dataset`.

IV. METHODS

All methods read only the 20 official channels of Section III and quantities derived from them. Parts A–B and F are the first round, Parts C–E and G the pre-registered campaign.

A. Window Models

A window is a 600×20 array (60 s at 10 Hz), standardised per channel with training-split statistics. We compare four sequence classifiers [10], [11]: a 1-D ResNet [10] on the 10 Hz signal and three token models on a shared two-layer convolutional stem with stride 4 (150 tokens of width 128): two bidirectional GRU layers [12], a four-layer Transformer encoder [13] and four bidirectional Mamba-2 layers [14], [15], a pure-PyTorch implementation of the published algorithm. All four share one pooling head whose hidden vector is the window embedding reused in Part E. Parameters: ResNet 2.31 M, bi-GRU 0.73 M, Transformer 0.70 M, Mamba-2 1.07 M.

B. Reference Training Recipe

Training uses class-balanced sampling weights $(N/(Kn_k))^{0.5}$ capped at 10 (N windows, K classes, n_k per class), AdamW [25] with warm-up over the first 3% of the steps and cosine decay, batches of 256, label smoothing 0.05 and early stopping on validation window macro-F1. The feature baselines feed 180 window statistics (nine per channel) and the physical features proposed by large language models (LLMs) in Part F to a class-balanced logistic regression and a histogram gradient-boosting classifier [26], [9].

C. Model-Disjoint Internal Folds and Pre-Registration

Because the released validation split shares car models with training (Section III), every campaign choice was made on internal car-model-disjoint folds drawn per task from the train + validation pool: three folds per task hold out 19 of 57 candidate car models (size) and 16, 15 and 20 (BEV vs ICE). A fold's training set drops every drive whose identity group also drove a held-out model; the one shared device pool spanning 19 car models is not an owner and loses only its drives of held-out models (fleet rule).

For the reference recipe, early stopping on the shared-model validation split selects epochs 16–22, on the folds 1–8 (median about 4): that validation rewards memorising car models. Final models therefore train for the median fold best epoch (GRU S9 11, ResNet S9 16, GRU B5 2, ResNet B5 7, episode model 10).

Six gates were pre-registered; G0 and G5 are procedural (stored numbers reproduced; one frozen test read per final model). G1: the reference recipe's fold drive-level score lies within 5 points of its test value, else the folds would have been redesigned. G2: the best recipe reaches a fold floor of 70 (size) or, for BEV vs ICE, 90 and at least 1 point over the reference recipe (or 92 on human-only drives); on failure, size reports ceilings instead of a 90 target and BEV vs ICE keeps the reference recipe. G3: the episode model gains 2 points over the best window recipe (size) or reaches 91 (BEV vs ICE), else it is only an ablation. G4: an ensemble or stacker needs 1.0 points over the best single model out of fold (outcomes in Section V and Table IV).

D. Recipe Changes S1–S9 and B1–B5

Each ladder adds one change at a time to the reference S0 / B0 (S1–S6 cumulative, S7–S9 each on S5; the BEV-vs-ICE ladder skips S3 and S6). S1 / B1: car-model-balanced sampling inside each class (shares $\propto$ hours$^{0.5}$). S2 / B2: regression head on the log footprint. S3: CORAL ordinal head [20]. S4 / B3: log curb-mass and four-class powertrain heads (B3 also trains the auxiliary heads on HEV / PHEV windows). S5 / B4: physics-safe augmentation (joint sign flip of the six lateral channels, one masked 3–15 s span with the lead validity flag cleared, dropout of one channel group, Gaussian noise 0.02). S6: same-class cutmix across car models. S7 / B5: down-weighting of windows with driving automation system (DAS) activity by $(1-\beta)+\beta(1-s)$, s the DAS share, $\beta = 0.5$. S8: conditional adversarial car-model head [18], [27] with gradient reversal ($\lambda = 0.1$). S9: group-DRO-lite [17] (per-car-model loss re-weighting). Decisions stay softmax arg-max; the final recipes are S9 (size) and B5 (BEV vs ICE).

E. Episode-Centric Multiple-Instance Model

The episode MIL (multiple-instance learning) model treats each drive as a bag of the transients in which mass and powertrain should show, trained for both tasks jointly. Eight episode types are detected on Savitzky-Golay-smoothed speed [28]: ACC (acceleration), DEC_LIFT and DEC_BRK (deceleration with the brake mostly released or pressed), STOP, LAUNCH, LEAD_ACC and LEAD_DEC (lead acceleration or deceleration onsets in car-following range) and TURN (sustained low-speed curvature); 735,571 episodes come from 8,662 drives (545 have none). An episode is an 18-s onset-anchored crop of 22 channels (the 20 inputs, smoothed acceleration, validity) with a 9-dimensional context vector; acceleration and deceleration episodes are also re-indexed on a speed grid (0–36 m/s, 0.5 m/s bins) as an acceleration-against-speed profile $a(v)$.

A shared encoder ($d = 128$; convolutional stem, residual blocks, type embedding, context MLP) embeds the crops, a grid encoder the $a(v)$ grid, and an adapter the frozen 60-s window embeddings of the window models (Part A) as a ninth instance type; on the folds the adapter uses only the fold's own window checkpoints. Gated attention [19] pools each bag per task; heads predict size, BEV vs ICE, four-class powertrain, log footprint, log curb mass and two ordinal thresholds. The loss is the bag cross-entropies + 0.3 instance cross-entropies + 0.5 Huber(log footprint) + 0.25 Huber(log mass) + 0.25 cross-entropy (powertrain-4) + 0.25 binary cross-entropy (ordinal); about 0.99 M parameters train for a fixed 10 epochs (AdamW, warm-up, cosine decay). The bag prediction is the drive-level output; window-mean aggregation of the same model gives the row comparable with the window recipes.

F. LLM-Assisted Feature Discovery

Following Section II, LLMs only propose programs that a verifier scores [22]; they never classify. Six proposer agents, one physics lens each, saw the channel definitions and a training-only sample of 2,727 windows and proposed 74 candidate features; six verifier agents ran the code, measured class separation and dropped 21, and an auditor removed 19 more, leaving 34 registered features. The coverage floor (the minimum share of windows on which a feature is defined) was lowered from 40% to 10% because gradient boosting treats a missing regime as information; everything was fixed before any evaluation split was touched.

G. Aggregation to Drive and Vehicle Level

Drive and vehicle protocols are those of Section III (mean of the per-window probability vectors, arg-max). Out of fold, weighted (DAS, episode, steering) arithmetic and log-odds means, robust and gated means and a two-stage vehicle rule were compared with the plain mean on the reference recipe;

none gained the required 1.0 drive-level points on both tasks (Section V). A supplementary rule fixed before the final test run, the log-odds mean with window weights $(1-s+0.1)^{0.5} \times (1+n_{\text{acc}}+n_{\text{dec}})$ (s DAS share, n acceleration / deceleration episode counts), is reported beside every final model.

V. Experiments

A. Set-Up

All runs used one RTX 5080 GPU (16 GB). Seeds are stated in the table captions; the gradient-boosting baselines (GBM) use scikit-learn [26]. Macro-F1 is in percent under the protocols of Section III; the headline is the drive level on held-out car models, scored once per final model after all choices had been made on the internal folds. Unit-level intervals are 95% cluster bootstraps over car models (with several seeds, the mean of the seeds' bounds); multi-seed entries are mean $\pm$ std.

B. Main Results on Held-Out Car Models

Table II gives the headline results. Architecture is not the bottleneck: the four reference sequence models lie within about 2 points at window level and close together at drive level. Averaging the windows of a drive adds 7–12 points; pooling further to vehicle level does not. For the sequence models, windows without automation activity (human-only columns) are harder for size and easier for BEV vs ICE, which Section VI relates to the controller of the driving automation system (DAS); the human-only drive columns cover fewer drives and are not directly comparable with the full columns.

The recipes selected on the folds raise the drive level without a matching gain at window level: S9 adds 4–6 points on size and B5 2–4 points on BEV vs ICE over the reference GRU and ResNet. The gains are stable over seeds, but the intervals over car models overlap (GRU S9 67.1 [54.4, 74.2] against the reference GRU 62.9 [48.9, 72.4]; GRU B5 92.6 [87.1, 96.2] against 89.0 [81.8, 93.7]). The episode MIL model matches but does not beat the window recipes under either aggregation; its fold advantage on BEV vs ICE (95.8, Table IV) did not carry to the test models.

C. Held-Out Devices

Table III repeats the main rows with devices held out and car models shared: all drive-level scores are far higher and the LLM + statistics GBM becomes the best size model. The recipes appear to trade within-model recognition for cross-model generalisation, since GRU S9 is below the reference GRU at window level and slightly above it at drive level (79.5 [70.8, 84.4] against 78.1).

D. Internal Folds and Recipe Sweep

Table IV summarises the campaign on the internal folds of Section IV. Gate G1 passed: the three-fold means of the reference recipe lie within about 5 points of its test values in Table II. The fold-0 size sweep spans 4 points, no more than the spread between folds, and no recipe approaches 70, so gate G2 for size fails; on BEV vs ICE, augmentation (B4) brings the gain and B5 reaches 93.1 over three folds, passing G2. The episode MIL model with gated attention [19] is level with S9 on size (G3 fails) and +2.7 over B5 on BEV vs ICE (G3 passes); its window-mean aggregation is weaker (61.1 and 88.8), and gated attention beats a learned instance mean on fold 0 (67.8 against 66.7; 96.0 against 94.8).

The final models train for fixed epoch counts taken from the folds; their shared-model validation scores (GRU / ResNet S9 60.7 / 62.8 and B5 75.6 / 79.3; MIL 79.1 on size and 97.5 on BEV vs ICE) are no guide to the test values.

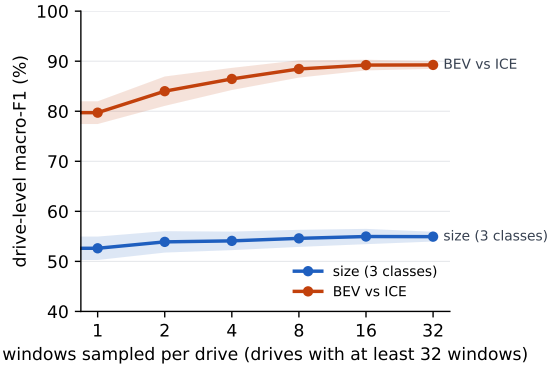


Fig. 1. Evidence sufficiency on held-out car models (reference predictions, drive-level macro-F1 in percent): on test drives with at least 32 eligible windows, m windows are drawn per drive and their probability vectors averaged (mean over 50 draws; band: one std). Size moves from 52.6 ($m = 1$) to 54.9 ($m = 32$; all windows 56.0), BEV vs ICE from 79.7 to 89.3 (all windows 89.1), saturating by 8–16 windows.

E. Ceilings, Separability and Evidence Sufficiency

Three zero-training oracles bound the size task on the held-out models: a per-car-model majority oracle (every drive of a held-out model gets that model's majority prediction) lifts the ensemble of the 12 reference deep runs from 63.2 to 72.0, so perfect within-model consistency would still leave size far from 90 (BEV vs ICE: 89.8 to 93.5). The footprint-noise oracle of Fig. 2 scores 92.1 at $\sigma = 0.10$ m^2 and 86.6 at 0.15, so reaching 90 needs a footprint error of about ± 0.12 m^2, about 2.5% of a 4.6 m^2 footprint. A curb-mass oracle (two mass cuts fitted on the training drives) scores 63.3 on the 1,139 test drives with a known mass, so mass is no better proxy.

The errors sit where the labels are closest: 21 of the 41 held-out car models lie within 0.15 m^2 of a differently labelled training model, and in one reference GRU run window accuracy is 53.0% within 0.1 m^2 of a threshold against 89.9% beyond 0.8 m^2. With the reference predictions, LARGE unibody vehicles are read as MID or SMALL while pickups and body-on-frame SUVs are recognised (per-model shares in Section VI), and MID models just above the 4.43 m^2 cut are read as SMALL (drives correct: GWM Haval H6, Audi Q5, Lexus RX and Nissan Altima 0%, Kia Stinger 25%).

Evidence per drive is not the limit for the reference models (Fig. 1): drive accuracy stays between 63.8 and 73.8% (size) and 85.7 and 93.2% (BEV vs ICE) from 1–2 to more than 64

TABLE II

HELD-OUT CAR MODELS (SPLIT_PLATFORM TEST SPLIT; SIZE / BEV VS ICE: 41 / 39 CAR MODELS, 1,141 / 836 DRIVES, 732 / 564 WITH HUMAN-ONLY WINDOWS, 329 / 244 VEHICLES). MACRO-F1 IN PERCENT (PROTOCOLS OF SECTION III); DEEP MODELS MEAN OVER THREE SEEDS (STD WHERE GIVEN), GBM SINGLE-SEED; CHANCE ABOUT 33 (SIZE) AND 50 (BEV VS ICE). S9 / B5: RECIPES SELECTED ON THE INTERNAL FOLDS (SECTION IV); EPISODE MIL WITH WINDOW-MEAN AGGREGATION AND ITS OWN BAG PREDICTION; BOOTSTRAP INTERVALS IN THE TEXT.

	Size (SMALL / MID / LARGE)					BEV vs ICE				
	window		drive		vehicle	window		drive		vehicle
Model	full	human	full	human	full	full	human	full	human	full
Statistics + GBM	49.7	46.3	54.7	50.5	51.5	69.5	73.5	73.2	74.3	76.0
LLM features + GBM	40.5	47.3	63.2	58.9	58.6	69.6	84.2	89.4	89.1	90.1
LLM + statistics + GBM	54.1	49.8	64.9	64.2	61.8	73.2	80.2	80.9	82.6	83.4
1-D ResNet	53.6 ± 1.0	48.0	61.1 ± 1.3	60.3	56.4	78.4 ± 0.4	83.2	91.1 ± 0.8	89.5	90.3
bi-GRU	54.7 ± 0.3	50.4	62.9 ± 0.8	63.3	57.2	79.2 ± 0.8	85.2	89.0 ± 0.1	90.8	88.6
Transformer	52.5 ± 0.8	47.7	61.8 ± 0.9	61.3	55.3	77.0 ± 0.4	83.5	88.4 ± 0.6	90.1	89.0
Mamba-2 (bidirectional)	53.1 ± 0.4	49.8	61.2 ± 0.3	61.1	55.2	77.3 ± 0.7	84.0	89.4 ± 0.3	90.9	90.6
bi-GRU, S9	53.2 ± 0.3	47.8	67.1 ± 0.0	61.7	60.0 ± 0.6	–	–	–	–	–
1-D ResNet, S9	53.2 ± 0.8	49.5	66.8 ± 1.7	62.9	62.6 ± 1.4	–	–	–	–	–
bi-GRU, B5	–	–	–	–	–	77.0 ± 0.3	83.7	92.6 ± 0.1	89.1	93.4 ± 0.4
1-D ResNet, B5	–	–	–	–	–	79.9 ± 0.7	83.2	93.0 ± 0.8	88.7	92.8 ± 0.4
Episode MIL, window mean	54.2 ± 0.3	50.0	64.6 ± 0.5	62.5	62.3 ± 0.9	78.8 ± 0.1	82.2	90.2 ± 0.2	89.9	90.4 ± 0.5
Episode MIL, bag prediction	–	–	64.3	63.2	62.4	–	–	93.0	91.5	92.8

TABLE III

HELD-OUT DEVICES, CAR MODELS SHARED (SPLIT_DEVICE TEST SPLIT; SIZE 1,218 DRIVES / 219 VEHICLES / 81 CAR MODELS, BEV VS ICE 922 / 143 / 63). MACRO-F1 IN PERCENT PER WINDOW AND PER DRIVE (PROTOCOL FULL); MEAN ± STD OVER THREE SEEDS, ONE SEED FOR GBM AND MIL.

	Size		BEV vs ICE	
Model	window	drive	window	drive
LLM + stat. + GBM	72.4	80.6	84.4	94.1
1-D ResNet	68.4 ± 3.0	77.4 ± 2.2	82.4 ± 0.7	96.2 ± 0.5
bi-GRU	70.2 ± 0.9	78.1 ± 1.4	87.4 ± 0.4	96.5 ± 0.1
ResNet S9 / B5	64.8 ± 0.3	80.6 ± 0.1	83.1 ± 0.2	97.2 ± 0.3
GRU S9 / B5	62.9 ± 1.0	79.5 ± 0.7	78.8 ± 0.4	95.1 ± 0.3
MIL (1 seed)	68.4	78.8	86.5	96.1

windows per drive, and it does not grow with the number of acceleration, deceleration or lead-acceleration episodes.

F. LLM-Proposed Physical Features

Table V compares gradient boosting on hand-crafted window statistics, on the 34 registered LLM-proposed features and on both: alone, the physical features are weak per window but strong per drive, on a par with the reference sequence models on BEV vs ICE, and adding the statistics helps size but lowers BEV vs ICE at drive level. By permutation importance of the LLM-features-only gradient boosting on the validation split, BEV vs ICE rests on lift-off deceleration (while moving and shortly after pedal release), the share of deceleration under a pressed accelerator or without the brake, the lead-to-ego deceleration delay and the 10th percentile of brake-free deceleration; size on the curvature-per-steering-angle gain at 6–18 m/s and at low speed, the Ackermann gain at 6–12 m/s and the same delay.

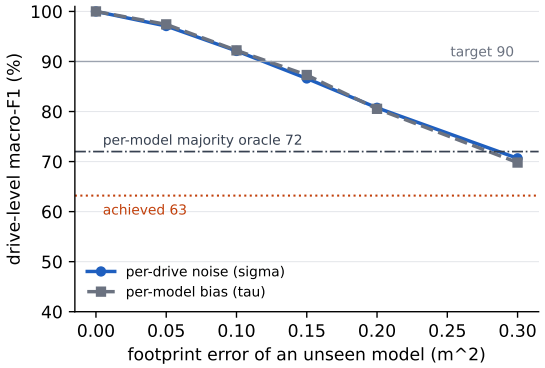


Fig. 2. Footprint-noise oracle for the size task on held-out car models (drive-level macro-F1 in percent, 200 repeats): each drive is classified from the true footprint of its car model plus an error, cut at the fixed thresholds 4.43 and 4.88 m^2; per-drive noise of std σ (circles) or a per-car-model bias τ (squares), both in m^2. Lines: the reference ensemble of the twelve reference sequence-model runs (63.2), its per-car-model majority oracle (72.0) and the 90 target.

G. Aggregation and Ensembles

Out of fold on the reference recipe (GRU and ResNet pooled), the best aggregation variant moves size from 67.5 to 67.8 (+0.3) and down-weighting DAS windows in the mean moves BEV vs ICE from 93.0 to 94.8 (+1.8), short of the pre-registered +1.0 on both tasks, so the plain mean remains the headline. Ensembles stayed below the G4 gate: a GRU + ResNet mean gains under one point over the best member out of fold (BEV vs ICE 93.9 against 93.1–93.5; size 68.8 against 68.2) and stacking is worse (92.3 and 66.5). The supplementary rule fixed before the final test run, the DAS- and episode-weighted log-odds mean of Section IV, moves size from 66.9 to 68.7 on the six final S9 runs (+1.7, 6 of 6 runs improved) and BEV vs ICE from 92.8 to 93.5 on the six

TABLE IV

MODEL-DISJOINT INTERNAL FOLDS OF SECTION IV: DRIVE-LEVEL MACRO-F1 (PROTOCOL FULL) ON EACH FOLD'S HELD-OUT CAR MODELS AND THE THREE-FOLD MEAN, ONE SEED PER ENTRY; MIL ROWS ARE THE MODEL'S OWN BAG PREDICTION; S0 / B0 REFERENCE, S9 / B5 FINAL RECIPES. (A) FOLD-0 SWEEP WITH THE BI-GRU; (B) CANDIDATES ON ALL THREE FOLDS.

(a) Fold 0, bi-GRU

Size	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
	65.3	62.8	66.8	64.8	65.2	66.2	64.8	65.6	66.5	66.9
BEV vs ICE	B0	B1	B2	B3	B4	B5				
	89.4	88.0	87.4	88.7	92.8	92.8				

(b) Three folds

Recipe	fold 0	fold 1	fold 2	mean
Size				
GRU S0 (reference)	65.3	69.5	60.4	65.1
GRU S2 (footprint head)	66.8	–	–	65.5
GRU S5 (augmentation)	66.2	–	–	66.5
GRU S9 (final)	66.9	70.5	64.8	67.4
ResNet S0 (reference)	–	–	–	66.6
ResNet S9 (final)	67.0	70.4	61.2	66.2
Episode MIL, bag	67.8	65.6	67.5	67.0
BEV vs ICE				
GRU B0 (reference)	89.4	90.8	95.4	91.9
GRU B4 (augmentation)	92.8	–	–	92.8
GRU B5 (final)	92.8	91.1	95.4	93.1
ResNet B0 (reference)	–	–	–	93.1
ResNet B5 (final)	95.7	93.9	96.1	95.2
Episode MIL, bag	96.0	98.2	93.2	95.8

TABLE V

GRADIENT BOOSTING ON WINDOW STATISTICS, ON THE 34 REGISTERED LLM-PROPOSED FEATURES AND ON BOTH (HELD-OUT CAR MODELS, ONE SEED); MACRO-F1 IN PERCENT PER WINDOW (FULL AND HUMAN-ONLY) AND PER DRIVE (PROTOCOL FULL).

	Size			BEV vs ICE		
Features	full	human	drive	full	human	drive
Statistics	49.7	46.3	54.7	69.5	73.5	73.2
LLM physical	40.5	47.3	63.2	69.6	84.2	89.4
LLM + statistics	54.1	49.8	64.9	73.2	80.2	80.9

B5 runs (+0.7, 6 of 6).

VI. DISCUSSION AND LIMITATIONS

A. Label Resolution, Not Sensing

Under the fixed footprint cut, 90 drive-level macro-F1 needs each held-out model's footprint to within about ± 0.12 m^2 (Fig. 2). Two curb-mass cuts fitted on training models score only 63.3, and 21 of the 41 held-out car models lie within 0.15 m^2 of a differently labelled training model. The gap is set by label resolution and fleet composition, not by model capacity or evidence per drive: architectures stay within about 2 points (Table II) and more windows per drive add little for size (Fig. 1). Per-model errors (Section V) split LARGE by body type: pickups and body-on-frame SUVs are recognised (GMC Yukon, Ford Ranger and Ford F-150 Lightning: 95.5–100% of drives), while unibody three-row SUVs and full-size sedans are read as MID or SMALL (Genesis G80 and G90, Volkswagen Atlas, Tesla Model S and Kia EV9: 0–5.6%), so the networks appear to learn LARGE as a truck's mass-and-driveline signature rather than as footprint.

B. What 90 Needs

Pooling a vehicle's drives does not rescue size (60.0–62.6 against 66.8–67.1 at drive level): about 45% of the test vehicles have a single drive and the errors are per car model, so even the per-car-model majority oracle lifts the reference ensemble only from 63.2 to 72.0. Three routes remain: seen car models, as under held-out devices, where the same recipes reach 79.5–80.6 (Table III); the vehicle level for BEV vs ICE, where the GRU B5 rises from 92.6 to 93.4 (95.5 on vehicles with at least two drives) and the ResNet B5 stays at 92.8–93.0; and labels whose margins exceed the resolution of about 0.12 m^2, such as cuts in wide gaps of the footprint distribution.

C. Two Protocol Lessons

First, while a driving automation system (DAS) is engaged, a controller tuned per car model supplies part of the control action. Windows with automation activity are easier for size than windows without (56.4 against 50.2, one bi-GRU run) and harder for BEV vs ICE (79.8 against 84.0, one ResNet B5 run): plausibly the controller carries platform information and masks the pedal behaviour that separates the powertrains. Hence the headline drive-level results (Table II) are paired with the human-only protocol, and the final BEV vs ICE recipe (B5) down-weights DAS windows (the size variant S7 did not win in the fold-0 sweep, Table IV). Second, early stopping on the released validation split, which shares car models with training, selects epochs 16–22, against 1–8 (median about 4) for the reference recipe on the model-disjoint folds (Section IV), and the final models' validation scores do not predict their test scores (Section V). Such a split rewards memorising car models, a shortcut [16]; we therefore fixed the training length from the folds (Table IV), and a future release would carve the validation split model-disjoint.

D. Privacy and Limitations

Section III lists the privacy measures. Labels describe car models, not people: a rare model may be recognisable as a model, never as a person. The same property that makes the data useful, that motion reveals the vehicle, restricts access to research on vehicle dynamics and driving behaviour and prohibits re-identification and linkage with other datasets. Several limitations remain. All logs come from one logger ecosystem and its software eras [23], and the fleet is skewed by region and brand (83.8% of the hours from North America, 4.8% from Europe, 4.2% from Asia); region, software era and device type are metadata, not inputs, but they correlate with powertrain. Labels are platform attributes: a misidentified platform mislabels every drive of that vehicle, and platforms sold with several powertrains the logs cannot separate, or of

unknown powertrain, carry no powertrain label (11.1% of the drives). On 56.2% of the drives the lead vehicle comes from a camera model rather than a radar, and pedal magnitudes are missing on 63.9% (accelerator) and 49.4% (brake) of the drives. The internal folds never hold out the LARGE $\times$ BEV cell, which one car model dominates, so their numbers for that cell reflect training models only; the test split does hold out models of that cell. With 41 (size) and 39 (BEV vs ICE) test car models, the GRU recipes' cluster-bootstrap intervals are wide (67.1 [54.4, 74.2] and 92.6 [87.1, 96.2]); differences of a few points between methods lie inside them.

VII. Conclusion

We asked whether a vehicle's size class and powertrain can be read from 60-s windows of trajectory and CAN signals for car models never seen in training. The released dataset (9,062 drives, about 3,950 h, 1,431 pseudonymous vehicles, 174 car models) fixes the labels, the model-disjoint and device-disjoint splits, the window-, drive- and vehicle-level protocols and the cluster-bootstrap intervals over car models. Powertrain is readable: BEV vs ICE reaches 92.6–93.0 drive-level macro-F1 on held-out car models. Size under a footprint cut is only partly readable: a pre-registered campaign raised the bi-GRU drive-level score from 62.9 to 67.1 (point estimates whose bootstrap intervals overlap), four sequence architectures stay within about 2 points, and zero-training oracles show that 90 would need a footprint resolution of about ± 0.12 m^2, while half of the held-out car models lie within 0.15 m^2 of a differently labelled training model. These results are reference points, not a leaderboard. The next steps are labels with margins above that resolution, more fleets and car models, and vehicle-level evaluation where vehicles have several drives.

Acknowledgments

References

[1] M. Helmbrecht, C. Olaverri-Monreal, K. Bengler, R. Vilimek, and A. Keinath, "How electric vehicles affect driving behavioral patterns," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 3, pp. 22–32, 2014.

[2] G. Geffen, J. Zhao, M. Shang, S. Wang, and Y.-J. Wu, "Safety, mobility, and environmental impacts of driver-assistance-enabled electric vehicles: An empirical study," *arXiv preprint arXiv:2601.17256*, 2026.

[3] Y. Liu, S. Wang, A. Panicker, K. Embry, A. Asanova, and T. Li, "A phase-aware AI car-following model for electric vehicles with adaptive cruise control: Development and validation using real-world data," *arXiv preprint arXiv:2510.21735*, 2025.

[4] Z. Sun and X. J. Ban, "Vehicle classification using GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 102–117, 2013.

[5] S. Dabiri, N. Marković, K. Heaslip, and C. K. Reddy, "A deep convolutional neural network based approach for vehicle classification using large-scale GPS trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 116, p. 102644, 2020.

[6] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 1, pp. 34–51, 2016.

[7] D. Hallac, A. Sharang, R. Stahlmann, A. Lamprecht, M. Huber, M. Roehder, R. Sosič, and J. Leskovec, "Driver identification using automobile sensor data from a single turn," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 953–958.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30*, 2017.

[10] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1578–1585.

[11] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[12] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017.

[14] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[15] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[16] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[17] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," in *International Conference on Learning Representations (ICLR)*, 2020.

[18] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[19] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 2127–2136.

[20] W. Cao, V. Mirjalili, and S. Raschka, "Rank consistent ordinal regression for neural networks with application to age estimation," *Pattern Recognition Letters*, vol. 140, pp. 325–331, 2020.

[21] M. Tan, M. A. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen, "Are language models actually useful for time series forecasting?" in *Advances in Neural Information Processing Systems 37*, 2024.

[22] B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F. J. R. Ruiz, J. S. Ellenberg, P. Wang, O. Fawzi, P. Kohli, and A. Fawzi, "Mathematical discoveries from program search with large language models," *Nature*, vol. 625, pp. 468–475, 2024.

[23] H. Schafer, E. Santana, A. Haden, and R. Biasini, "A commute in data: The comma2k19 dataset," *arXiv preprint arXiv:1812.05752*, 2018.

[24] Y. Wang, Y. Xu, J. Sun, and H. Zhou, "ADAS-TO: A large-scale multimodal naturalistic dataset and empirical characterization of human takeovers during ADAS engagement," *arXiv preprint arXiv:2603.06986*, 2026.

[25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Advances in Neural Information Processing Systems 31*, 2018.

[28] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.