BEYOND PERMUTATION INVARIANCE: IDEMPOTENCE IN SET DATA REPRESENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Permutation-invariant architectures are routinely described as learning on sets, but permutation invariance removes order without removing multiplicity. This distinction is often hidden because datasets typically provide only one finite registration of each underlying datum. If repeated registration of an already-present element does not change that datum, its representation must also be consistent across such alternative registrations. We formalize this requirement as idempotent set consistency and prove that a variable-cardinality family descends to a function on finite subsets exactly when it is permutation invariant and compatible with duplicate insertion and deletion. This yields registration fibers and quantitative fiber defects that distinguish set semantics from multiset and empirical-measure semantics. More generally, we use directed systems of registrations as a semantic checklist: the chosen cross-cardinality maps specify which changes of registration are intended to preserve the datum, and their compatibility reveals which invariances a representation should satisfy. For set semantics, this perspective leads naturally to Hausdorff geometry and near-idempotence under arbitrarily close observations. We study these ideas through pooling and Set Transformer audits, geometry-controlled interventions on pretrained ModelNet40 classifiers, and duplicate and near-duplicate corruptions in human trajectory prediction. Rather than treating idempotence as a universal desideratum, our framework uses registration consistency to distinguish competing data semantics and tests whether a model respects the one intended by the application.

1 INTRODUCTION

An unordered array can encode several different objects. The arrays (a, b), (a, a, b), and (a, b, b) have the same support, different multiplicities, and different empirical distributions. Permutation invariance identifies reorderings within each array length. It does not identify these three registrations. Whether a predictor should identify them depends on what the array represents.

For a geometric object, repeated observations need not add occupied geometry. For a tracked configuration, registering the same pedestrian twice need not introduce another physical pedestrian. Conversely, repeated observations can carry statistical evidence, and distinct entities may share the same observed state. Removing multiplicity in those settings can remove relevant information. A useful invariance therefore starts with a semantic decision: *which changes of registration preserve the datum?*

We formalize this decision through a quotient of finite registrations. A set-consistent predictor factors through support; a distribution-consistent predictor factors through empirical measure; a multiset predictor retains counts. These choices induce different cross-cardinality identifications. The distinction is consequential even when models fit the original task accurately: good predictions on one registration per datum do not determine behavior on its alternative registrations.

Our contribution is a representation-level framework and three complementary audits. First, we characterize when a variable-cardinality implementation factors through finite support and use its graded registration fibers to quantify departures from this requirement. Second, we separate exact registration compatibility from geometric stability and show why agreement at bounded cardinality does not determine the topology of an unbounded-cardinality limit. Third, we assess learned and implemented consistency under exact repetition of support elements, controlled geometric resampling,

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.

and the insertion of duplicate and near-duplicate object tracks. The results support a qualified conclusion: Set Transformer can learn a nearly idempotent maximum regressor, while clean point-cloud accuracy and low trajectory error do not by themselves imply registration consistency. The proposed criterion concerns the entire implemented representation, including preprocessing and inference controls.

2 Related work

Permutation invariance and multiplicity. Permutation invariance removes order, not multiplicity. Deep Sets (Zaheer et al., 2017) established a sum-decomposition framework for permutation-invariant learning, but summation over registered entries does not enforce invariance to changing their multiplicities on a fixed support. Permutation invariance and variable-length evaluation therefore do not by themselves establish that an implementation represents a function of support. Portilheiro (2019) explicitly distinguishes idempotent max pooling from multiplicity-sensitive summation, while developing representations whose learning targets retain multiset relations. Classical aggregation theory studies diagonal idempotence, $A_n(t, \ldots, t) = t$ (Grabisch et al., 2009), but this identity alone does not imply invariance to duplicating an element in an input containing distinct values. We require agreement across all registrations of the same support, characterize the corresponding factorization of a variable-cardinality family, and quantify its violations through graded registration fibers.

Geometric modeling and pooling stability. PointNet (Qi et al., 2017a) combines pointwise features with max pooling and establishes approximation results for Hausdorff-continuous finite-set functions. Bueno & Hylton (2021) develop cardinality-agnostic approximation results that distinguish max- and average-pooling networks through Hausdorff and Wasserstein geometry. Nikolentzos & Skianis (2026) analyze pooling and network stability under earth mover's, Hausdorff, and matching distances, including unequal-cardinality inputs. These geometries encode different notions of proximity and cannot be substituted without regard to the intended datum. In particular, stability in an empirical-measure metric can still permit disagreement between registrations with identical support. Greater approximation power does not resolve whether such multiplicity dependence is meaningful information or unwanted registration sensitivity. We separate exact support factorization from geometric stability and assess trained inference pipelines, rather than inferring consistency from approximation capability or a final pooling layer.

Cross-cardinality identifications and limit spaces. Levin et al. (2026) formulate size transfer through consistent sequences, and Yao et al. (2026) develop universality on the associated limit spaces. Their examples include zero-padding and uniform replication, which encode particular cross-cardinality identifications. These identifications are not interchangeable with finite-support semantics. In the topology of finite or compact subsets, collisions are intrinsic: distinct points may coalesce continuously, as $\{a, b_j\} \to \{a\}$ when $b_j \to a$. Zero-padding instead changes grade by adjoining a distinguished null coordinate and does not identify arbitrary repeated coordinates with their lower-cardinality support. Uniform replication preserves empirical frequencies and likewise does not impose invariance to arbitrary multiplicity changes. For support semantics, we therefore use the classical finite-subset quotient (Tuffley, 2002; Kallel & Sjerve, 2009), in which all registrations with the same support are identified, and separately specify the Hausdorff geometry that makes collisions continuous. We further exhibit cases in which the topological direct limit is strictly finer than the metric topology, despite agreement at every bounded stage. Thus finite-stage continuity and algebraic compatibility alone do not determine the geometry of an unbounded-cardinality limit; the geometry needed to interpret increasingly fine registrations must be specified separately.

Sampling density and redundant records. Density-aware point-cloud processing already separates geometry from its sampling law (Hermosilla et al., 2018; Wu et al., 2019). Our point-cloud audit combines a shared input bank, explicit coverage controls, two source populations, and native checkpoint validation. The trajectory audit addresses a different representation: each input trajectory stands for an entity. It studies downstream sensitivity to redundant records, rather than proposing a detector, tracker, or new forecasting architecture.

3 REGISTRATION CONSISTENCY

Let X be a nonempty set and Y an output space. A variable-cardinality implementation is a family $F_n : X^n \to Y$, $n \geq 1$. Its registration space $\mathcal{T}(X) = \coprod_{n\geq 1} X^n$ is *graded* by array length. Permutation orbits form $\mathcal{M}(X) = \coprod_{n\geq 1} X^n/S_n$, the space of finite nonempty multisets. Let

$$s_n(x) = \{x_1, \ldots, x_n\}, \qquad \mathrm{Fin}_{\leq n}(X) = \{K \subseteq X : 1 \leq |K| \leq n\},$$
$$\mathrm{Fin}(X) = \bigcup_n \mathrm{Fin}_{\leq n}(X). \tag{1}$$

The support quotient $X^n \to X^n/S_n \to \mathrm{Fin}_{\leq n}(X)$ first removes order and then multiplicity. Bounded support spaces are *filtered* by inclusion, whereas array length grades the registrations. These are different structures.

Definition 3.1 (Idempotent set consistency)**.** The family $F = (F_n)$ is set-consistent if there exists $f : \mathrm{Fin}(X) \to Y$ such that $F_n = f \circ s_n$ for every n.

For $1 \leq j \leq n$, write $d_j x = (x_1, \ldots, x_n, x_j)$ for duplicate insertion.

Theorem 3.2 (Exact factorization)**.** *The family F is set-consistent if and only if it is permutation invariant at every grade and $F_{n+1}(d_j x) = F_n(x)$ for every n, x, j. The factor f is unique. Insertion compatibility also gives deletion compatibility whenever another copy remains.*

Proof. Factorization gives both identities because neither operation changes support. Conversely, delete all repeated entries from a tuple and permute the survivors into any chosen enumeration of its support. Each operation preserves the output. Thus any two registrations with the same support have the same output, defining a unique f. □

The theorem characterizes a semantic contract, not an architectural approximation theorem. A shared parameterization across n does not establish its premise. For instance, $F_n(x) = n$ is permutation invariant, and even constant on every within-grade support fiber, but does not factor through $\mathrm{Fin}(X)$.

Fibers and quantitative defects. For $K = \{u_1, \ldots, u_r\}$, the grade-n ordered fiber is $s_n^{-1}(K)$; it has $r!\left\{{n \atop r}\right\}$ elements when $n \geq r$, where $\left\{{n \atop r}\right\}$ is a Stirling number. After permutation quotient, the fiber consists of positive counts $(c_1, \ldots, c_r)$ with $\sum c_i = n$, and has $\binom{n-1}{r-1}$ elements. If Y is metric, define

$$\Delta_F(K) = \sup_{n,m\geq r\,;\, s_n(x)=s_m(y)=K} d_Y(F_n(x), F_m(y)). \tag{2}$$

Then F is set-consistent exactly when every $\Delta_F(K)$ vanishes. Finite audits estimate restricted defects, not this unrestricted supremum. Experiment 1 compares each length-100 registration to the once-per-element baseline and also computes the diameter within that grade. These diagnose different failures.

Other semantics and directed systems. An empirical distribution uses $\mu_n(x) = n^{-1}\sum_i \delta_{x_i}$. Repeating *every* entry k times preserves μ_n, but repeating only one entry generally changes it. Thus uniform replication along $n \mid m$ describes a different directed system from the support inclusions along $n \leq m$. Their set-theoretic limits are rational finitely supported probabilities and finite supports, respectively. Counting measures $\sum_i \delta_{x_i}$ retain total mass. None of these is the universally correct representation. Appendix B spells out the systems and explains why a space with fibers is not automatically a topological fibration.

4 FROM EXACT CONSISTENCY TO GEOMETRIC STABILITY

For a metric space (X, d), support geometry is naturally measured by the Hausdorff distance

$$d_H(A, B) = \max\{\sup_{a\in A}\inf_{b\in B} d(a, b),\ \sup_{b\in B}\inf_{a\in A} d(a, b)\}. \tag{3}$$

On compact metric X, the quotient topology of X^n under support equivalence agrees with the Hausdorff topology on $\mathrm{Fin}_{\leq n}(X)$. Nevertheless, the topological direct limit over n can be strictly

finer than the Hausdorff topology on $\mathrm{Fin}(X)$. We give explicit counterexamples for supports, empirical distributions, and rational grids in Appendix C. Our construction is bottom-up: it begins with finite registrations and their identifications. An intended continuum object and its geometry supply additional modeling information; they do not follow automatically from permutation invariance and finite-dimensional continuity.

The metric union $(\mathrm{Fin}(X), d_H)$ instead has completion $\mathcal{K}(X)$, the nonempty compact subsets, when X is complete. This realization makes near-duplicate observations comparable across cardinalities. If a registration adds observations within distance ε of a retained support K, its support A satisfies $d_H(A, K) \le \varepsilon$.

Proposition 4.1 (Near-duplicate stability)**.** *Suppose $F_n = f \circ s_n$ and $d_Y(f(A), f(B)) \le \omega(d_H(A, B))$ for a nondecreasing modulus ω with $\omega(t) \to 0$ as $t \downarrow 0$. If A and B retain K and add only observations within ε of K, then $d_Y(f(A), f(B)) \le \omega(\varepsilon)$. The bound is independent of their cardinalities.*

Every point of either support is within ε of the retained K in the other. Exact consistency alone gives no such modulus: $f(K) = |K|$ is set-consistent but discontinuous at collisions. Conversely, mean and normalized attention can be smooth at each fixed length while retaining multiplicity dependence. Grouping identical keys in attention replaces their softmax weights by $c_j \exp(\ell_j)$; uniform replication cancels, nonuniform replication generally does not. This does not prevent learning a max-like function. Appendix D gives the exact formulas and the count-weighted computation used in Experiment 1.

5 Experiments

The experiments test three levels of the framework: exact support fibers, geometric resampling of a fixed source, and redundant entity records. Experiments 2 and 3 assess statistical reliability through agreement under repeated or controlled changes of registration, following the measurement-error perspective of Bartlett & Frost (2008). We report within-datum changes rather than an intraclass correlation. Each audit also includes a task metric: a constant predictor would be stable but uninformative.

5.1 Experiment 1: learned maximum versus registration consistency

We revisit maximum regression motivated by Lee et al. (2019). Inputs contain $n \sim \mathrm{Uniform}\{1, \dots, 10\}$ independent values in $[0, 100]$; the target is their maximum. Although the benchmark target is known analytically, we frame the learning problem as one in which its functional form need not be known to the practitioner. The semantic checklist is assumed to have established only that repeated registration of an already-present value does not change the represented datum or its target; equivalently, the target factors through finite support. We train mean-, sum-, max-, and log-mean-exp-pooling networks and a Set Transformer for 200,000 Adam updates, with ten seeds. Checkpoints are selected by independent online probes, and evaluated on a separate test bank of 100,000 examples per cardinality. This is an extended controlled reconstruction, not an exact rerun of the original training schedule.

For each of 100 shared two-element and 100 shared three-element supports, we exhaust all positive count vectors summing to 100: 99 and 4,851 registrations, respectively. Although these registrations have total length 100, their support cardinalities remain two and three. The audit therefore extrapolates in registration grade, not in the semantic size of the represented set: every registration represents the same support and has exactly the same target.

Let D_r denote mean absolute prediction change from the once-per-element support baseline. The within-grade diameter is the largest minus smallest registered prediction. All registrations preserve the maximum exactly, but length 100 lies outside the training range. Table 1 separates task fitting from registration behavior. Sum pooling attains lower clean MAE than mean pooling yet exhibits substantially larger repeated-registration changes. Max pooling has the structural duplicate identity, up to numerical computation. Set Transformer learns both accurate maximum predictions and small measured defects: clean MAE 0.0157, three-support mean defect 0.00385, and mean within-grade diameter 0.0340. These results support its ability to learn max-type behavior. They do not establish exact idempotence or a guarantee outside the audited supports. The main implication is

Model	Clean MAE	D_2	D_3	diam_3
Mean	0.769 ± 0.069	2.707 ± 0.132	3.447 ± 0.230	44.937 ± 1.427
Sum	0.278 ± 0.051	33.517 ± 8.906	27.716 ± 8.000	144.570 ± 36.149
Max	0.005 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
LME ($\tau = 0.1$)	0.034 ± 0.008	0.061 ± 0.018	0.072 ± 0.020	0.435 ± 0.119
LME ($\tau = 1$)	0.080 ± 0.007	0.159 ± 0.013	0.186 ± 0.016	1.123 ± 0.099
Set Transformer	0.016 ± 0.009	0.002 ± 0.003	0.004 ± 0.004	0.034 ± 0.037

Table 1: Maximum regression and exact-fiber audit. Mean $\pm$ sample SD across ten training seeds; fiber statistics are first averaged over 100 supports of each size. Defects are in target units. Max-pooling changes are below 10^{-5} on average and round to zero.

methodological: duplicate consistency should be measured explicitly when the task calls for it, even for a model that fits the task well. Appendix F gives all architectures, checkpoint rules, learning curves, and seed-level results.

5.2 Experiment 2: geometry-controlled point-cloud registration

Checkpoints, source lineages, and controlled sources. We evaluate seven frozen ModelNet40 classifiers: PointNet, DGCNN, PointMLP, PointNet++ MSG, PointConv, PointMamba, and PointGPT-S (Qi et al., 2017a; Wang et al., 2019; Ma et al., 2022; Qi et al., 2017b; Wu et al., 2019; Liang et al., 2024; Chen et al., 2023). The released checkpoints come from two distinct ModelNet40 input lineages. PointNet, DGCNN, and PointMLP are natively evaluated on the HDF5 release, whereas PointNet++ MSG, PointConv, PointMamba, and PointGPT-S use the released `modelnet40_normal_resampled` point/normal files. Each checkpoint first passes full-test-split evaluation under its native preprocessing contract; native accuracies range from 90.68% to 93.64%.

Evaluating each checkpoint only on its native lineage would confound checkpoint identity with source lineage. We therefore cross every checkpoint with two separately defined source populations and supply identical ordered float32 registrations to all checkpoints within each source. The released HDF5 objects contain only 2,048 points each. Although sufficient for native benchmark evaluation, this finite cloud is too small to support the repeated density and cardinality interventions used here: at the $n = 2048$ endpoint it would be exhausted, leaving no resampling variability. We therefore construct a *CAD-derived source* by sampling the positive-area faces of the original OFF meshes and mapping them into the HDF5 coordinate frame with a frozen object-specific calibration. The *pool-derived source* instead draws distinct rows from each released 10,000-point `normal_resampled` reservoir after one frozen normalization computed from the full pool.

The two sources are not assumed to be interchangeable registrations of one common surface. Matching object names do not certify identical geometry, orientation, normal attributes, or preprocessing lineage, so their results are reported separately and are never pooled. The full test split contains 2,468 objects, with three registration seeds and one spatial density field per object.

Geometric near-idempotence. Experiment 1 tests exact consistency along registration fibers, where duplicating an already-present element leaves the finite support unchanged. Literal duplicate insertion is less natural for a point cloud acquired from a continuous surface: independently sampled observations almost surely occur at different locations. Experiment 2 therefore tests a geometric relaxation of exact idempotence. If two finite registrations represent the same underlying geometric source and differ only through nearby observations, then their supports should remain close in Hausdorff distance, and a Hausdorff-continuous support-based representation should change correspondingly little. We refer to this controlled behavior as near-idempotence in the Hausdorff sense.

The density interventions act on the observation law, not on the material substance of the object. For a two-dimensional surface $S \subset \mathbb{R}^3$, its surface-area measure is determined geometrically by $\mathcal{H}^2|_S$. Changing the sampling density over that surface changes which regions are observed more frequently without changing either the support S or its geometric surface-area measure. This notion is distinct from a material mass density—for example, two geometrically identical objects made of different materials may have the same surface geometry and surface-area measure but different mass.

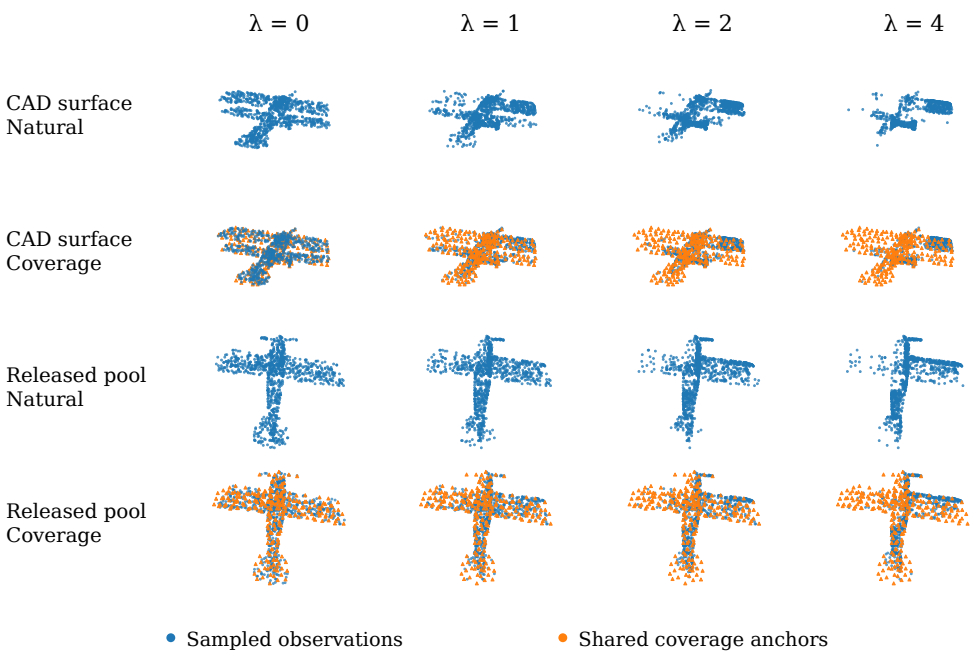


Figure 1: Controls for Experiment 2, illustrated with `airplane_0001`. Columns increase the spatial density strength λ at $n = 1024$. Natural sampling can leave parts of the source poorly covered. Coverage sampling retains 256 shared FPS anchors (orange) and adds 768 density-controlled observations (blue). The upper rows use the CAD-derived source; the lower rows use the released 10,000-point point/normal source. The two sources have distinct preprocessing and geometry lineages and need not coincide in orientation or surface realization. This object illustrates the protocol; quantitative results use the full test split.

For CAD objects, support semantics and surface-measure semantics are complementary rather than exclusive. A surface may be represented by its compact support $S \subset \mathbb{R}^3$, by its geometric area measure $\sigma_S = \mathcal{H}^2|_S$, or jointly by the pair (S, σ_S). The support records where the object is present, while σ_S records geometrically induced surface area. Neither should be confused with the acquisition law of sampled points or with material mass density. Experiment 2 primarily audits support-level near-idempotence through Hausdorff geometry, while the density interventions also expose behavior that is naturally interpreted through the induced surface measure and observation law.

Controlled interventions. A fixed spatial field changes the acquisition density at $\lambda \in \{0, 1, 2, 4\}$ without moving or renormalizing the source. *Natural* sampling uses the entire 1024-point budget under that law. *Coverage-controlled* sampling retains 256 FPS anchors and uses the remaining 768 points for density-controlled observations (Figure 1). Thus increasing λ can substantially alter empirical sampling density while limiting loss of geometric coverage.

A second registration provides a repeatability control at each fixed condition. Coverage-controlled pairs share their anchors, so their repeatability is conditional on that common scaffold. Two additional nested protocols use $512 \subset 1024 \subset 2048$ observations at $\lambda = 0$. Internal hierarchy counts remain fixed at their 1024-point reference values. These are explicit cross-cardinality stress tests of the frozen checkpoints, including PointMamba and PointMLP, rather than additional native benchmark scores.

Measurements. For 40-dimensional logits z, let $g(z) = z - \bar{z}\mathbf{1}$. The primary comparison is the symmetric relative centered-logit change

$$R(z, z') = \frac{\|g(z') - g(z)\|_2}{\max\{\sqrt{(\|g(z)\|_2^2 + \|g(z')\|_2^2)/2}, 10^{-12}\}}. \tag{4}$$

	CAD-derived source			Released 10k-pool source		
Model	R^0_{rep}	R^4_{den}	Acc.^{4}	R^0_{rep}	R^4_{den}	Acc.^{4}
PointNet	0.240	0.254	88.3	0.198	0.177	89.4
DGCNN	0.534	1.035	36.0	0.517	0.888	55.7
PointMLP	0.242	0.547	75.0	0.221	0.431	82.4
PointNet++	0.182	0.194	88.6	0.163	0.142	92.2
PointConv	0.116	0.285	81.8	0.095	0.187	90.1
PointMamba	0.129	0.453	78.8	0.110	0.291	86.7
PointGPT-S	0.167	0.452	75.9	0.140	0.306	85.3

Table 2: Point-cloud reliability under the two controlled source populations. R^0_{rep}: natural-sampling repeatability at $\lambda = 0$. R^4_{den}: coverage-controlled response from $\lambda = 0$ to 4. Acc.^{4}: accuracy (%) in that coverage condition. Lower R indicates greater agreement under the stated registration intervention, not higher classification accuracy. Full grids and confidence intervals appear in Appendix H.

Centering removes additive logit offsets, while the denominator controls for common scale away from the numerical floor. Repeatability compares independently generated registrations at a fixed condition; density response compares the observed registration at $\lambda > 0$ with its coupled $\lambda = 0$ counterpart; refinement compares a nested child registration with its parent.

We also report classification accuracy, class flips, finite-cloud Hausdorff distance, sliced W_2 (Bonneel et al., 2015), and coverage error. Hausdorff distance measures support-level geometric change, whereas sliced W_2 is sensitive to changes in the empirical distribution of observations. We average registration seeds within object and use 1,000 class-stratified object-cluster bootstrap replicates. Intervals are pointwise, not simultaneous. Checkpoint contrasts use paired object differences on the same stored registrations.

Results. The completed suite contains 1,451,184 result rows across all eight protocol/source combinations, with validated shared inputs. Under natural uniform sampling, DGCNN has $R^0_{\mathrm{rep}} = 0.534$ on the CAD-derived source and 0.517 on the pool-derived source, compared with 0.240 and 0.198 for PointNet (Table 2). The paired DGCNN–PointNet differences are 0.294 $[0.290, 0.297]$ and 0.319 $[0.316, 0.322]$, respectively. Because checkpoints receive the same stored registrations within each source, these are paired comparisons of the audited implementations under a common acquisition law.

Coverage control separates large changes in observation density from comparably large losses of geometric coverage, although it does not make the finite supports identical. On the CAD-derived source, the mean uniform repeat-pair Hausdorff distance is 0.0908, while the $\lambda = 4$ versus $\lambda = 0$ Hausdorff distance under coverage control is 0.1063. For the CAD-derived source, the density-pair sliced W_2 distance is 0.2592, compared with 0.0189 between uniform repeat registrations. Thus the empirical distribution of observations changes substantially while the support-level geometric perturbation remains much more limited.

The checkpoints respond very differently to this controlled change. For DGCNN, $R^4_{\mathrm{den}} = 1.035$ and accuracy falls from 90.8% to 36.0%. PointNet++ changes by only 0.194 and retains 88.6% accuracy. PointConv has relatively small repeatability changes but a nonzero response to the density intervention. Density compensation in one component of an architecture therefore does not by itself establish near-idempotent behavior of the complete checkpoint.

The pool-derived source exhibits a related but quantitatively different pattern: DGCNN has $R^4_{\mathrm{den}} = 0.888$ with 55.7% accuracy, whereas PointNet++ retains 92.2% accuracy with $R^4_{\mathrm{den}} = 0.142$. Equal values of λ do not impose equal geometric or distributional severity across the two source populations. Cross-source differences are therefore interpreted descriptively and reported together with their source-specific geometric measurements.

Increasing registration resolution does not automatically improve agreement with respect to the same underlying source. Every finer cloud retains its coarser predecessor, so the intervention adds observations without removing the previously registered support. PointNet++ and PointConv show consistently smaller mean representation changes than DGCNN, PointMLP, PointMamba, and

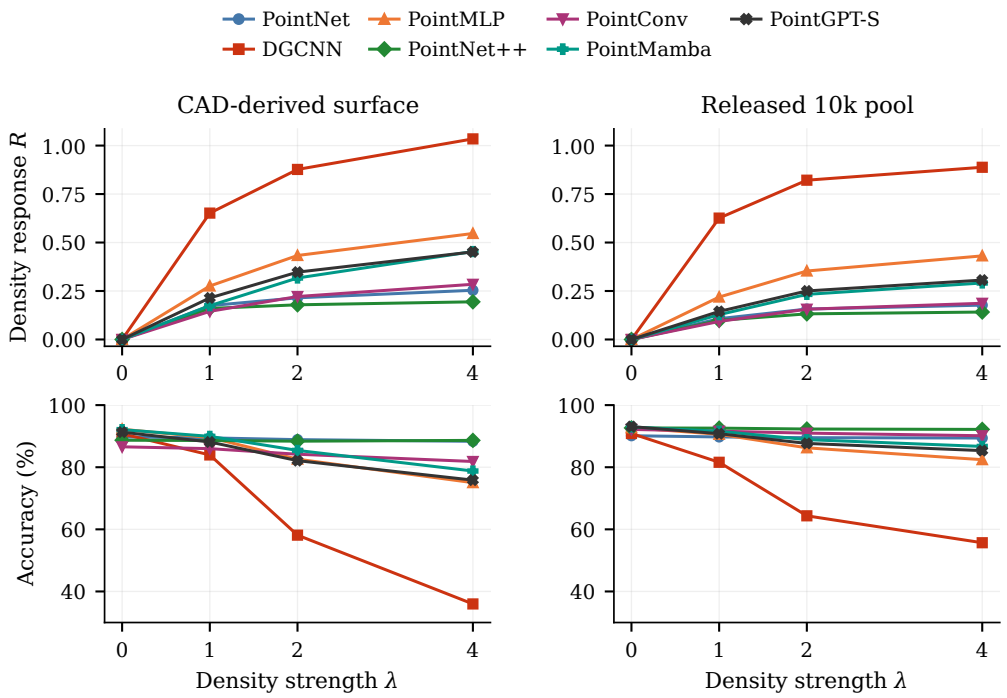


Figure 2: Coverage-controlled density response (top) and classification accuracy (bottom), averaged over objects and registration seeds. Within each panel, all checkpoints receive the same source population and the same ordered registrations, making checkpoint comparisons paired. Lines connect the four measured density strengths and do not assert behavior between them.

PointGPT-S across both sources, both nested designs, and both refinement transitions. PointNet joins this lower-change group for coverage-controlled $1024 \rightarrow 2048$ refinement, although its behavior is less uniform across the full nested suite. For PointMLP and PointGPT-S, the mean representation change increases at the second refinement despite retained coverage anchors.

These results separate task accuracy from geometric registration reliability. A classifier may retain its predicted label while its representation changes substantially under resampling, density shifts, or refinement of the same source object. Experiment 2 therefore tests whether agreement improves under controlled geometric approximations to a fixed datum, rather than whether additional points merely preserve classification accuracy. Detailed effect sizes and paired intervals appear in Appendix H.

5.3 Experiment 3: Duplicate Entity Records in Trajectory Prediction

A configuration record compresses an extended physical entity with mass and spatial extent into a position history. Duplicate tracks can register the same entity more than once. Classical tracking addressed detection- and trajectory-level exclusion (Milan et al., 2013), yet track duplication persists in modern trackers, as documented in UMOT's comparisons with MOTR and MOTRv2 (Xie et al., 2025). We test the downstream sensitivity of trajectory predictors to known redundant records. The occupancy perspective is motivated by set-driven multiagent dynamics (Capuani et al., 2024); here the datum is the set of complete observed histories $\{x^{(i)}_{1:8}\}$, with original-agent outputs required to transform equivariantly. The synthetic intervention preserves the represented physical entities, whose latent states need not be fully observed (Appendix E).

We examine Social GAN, Social-STGCNN, and AgentFormer (Gupta et al., 2018; Mohamed et al., 2020; Yuan et al., 2021) on ETH, HOTEL, UNIV, ZARA1, and ZARA2, with eight observed and twelve predicted frames. One source agent per valid scene window receives $k \in \{1, 2, 4\}$ extra records. Exact duplicates use $\sigma = 0$; near duplicates add independent Gaussian coordinate noise with $\sigma \in \{0.02, 0.05\}$ to each observed timestep. Predictions are scored only for the original agents.

Model	k	σ	$\langle\lvert\Delta\text{ADE}\rvert\rangle$ (relative)	$\langle\lvert\Delta\text{FDE}\rvert\rangle$ (relative)
Social GAN	2	0.00	0.0000 (0.0%)	0.0000 (0.0%)
Social GAN	2	0.05	0.0116 (2.2%)	0.0262 (2.5%)
Social GAN	4	0.00	0.0000 (0.0%)	0.0000 (0.0%)
Social-STGCNN	2	0.00	0.1162 (25.6%)	0.2531 (33.3%)
Social-STGCNN	2	0.05	0.1098 (24.2%)	0.2407 (31.7%)
Social-STGCNN	4	0.00	0.1415 (31.2%)	0.3046 (40.1%)
AgentFormer	2	0.00	0.0346 (14.8%)	0.0913 (22.7%)
AgentFormer	2	0.05	0.0356 (15.2%)	0.0951 (23.6%)
AgentFormer	4	0.00	0.0551 (23.5%)	0.1433 (35.6%)

Table 3: Trajectory error sensitivity: five-fold macro-mean absolute ADE/FDE changes and changes relative to each model's own baseline. k is the number of added records. Values shown as 0.0000 are at the numerical/reporting floor. These descriptive estimates measure changes in error, not trajectory displacement; the baseline errors are reported in the text.

Common random inputs and baseline-selected sample indices are held fixed for each comparison. AgentFormer additionally freezes the baseline DLow latent proposals (Yuan & Kitani, 2020), making its audit conditional on those proposals rather than an evaluation of the entire context-dependent sampling pipeline.

For each original agent, compute the signed change in its selected ADE or FDE and its absolute value before aggregation. Let M_0 be a five-fold macro-average baseline error and $A = \frac{1}{5}\sum_f \langle|\Delta M|\rangle_f$. We report A and $100A/M_0$. This ratio of macro-averages differs from both a signed percentage deterioration and an average of individual ratios. Baseline ADE/FDE are $0.533/1.044$ for Social GAN, $0.454/0.760$ for Social-STGCNN, and $0.234/0.403$ for AgentFormer. Two exact duplicates induce mean absolute ADE/FDE changes equal to $14.8\%/22.7\%$ of AgentFormer's baseline; four induce $23.5\%/35.6\%$. Its strong baseline accuracy therefore coexists with measurable duplicate sensitivity under the stated coupling. Social-STGCNN is also sensitive, while Social GAN's exact-duplicate error changes are at the numerical/reporting floor and grow for near duplicates. The audit therefore characterizes registration reliability at the level of the evaluated inference procedures under the stated stochastic couplings; unchanged ADE/FDE under exact duplication is an error-level stability result, not a test of trajectory-wise equality. Appendix I gives the full evaluation protocol, model-specific couplings, absolute and relative changes, all conditions and roles, and fold-level results.

6 Discussion and limitations

Registration consistency is an application-dependent requirement. The framework separates support, relative frequency, and count semantics before asking about architecture. In the exact-support audit, Set Transformer learns nearly idempotent behavior without acquiring a structural guarantee. In geometric resampling, controlled coverage and density comparisons reveal sensitivity beyond clean checkpoint accuracy. In forecasting, redundant records can materially affect errors despite a strong baseline predictor.

Architecture suggests hypotheses but does not replace a guarantee. Max aggregation with Lipschitz pointwise features admits Hausdorff stability bounds (Nikolentzos & Skianis, 2026). Hard FPS, radius cutoffs, and kNN can interrupt continuity even with fixed randomness. PointConv's density correction motivates a measure-based analysis, but a bound for a continuous integral operator does not automatically certify its discrete pipeline. Appendix D distinguishes structural guarantees from empirical stability and explains why the small duplicate defects observed for Set Transformer and LME in Experiment 1 do not establish exact idempotence.

The evidence concerns finite interventions and frozen implementations. Experiment 1 includes cardinality extrapolation; Experiment 2 changes finite supports under one field family; Experiment 3 reports descriptive error changes under specified couplings. Neither low defect nor high accuracy alone certifies the intended representation. Their joint evaluation exposes whether accuracy persists across the registrations that the application intends to treat as equivalent.

AI USE STATEMENT

In this work, we used generative AI tools for aiding and polishing manuscript writing, literature retrieval and discovery, and code implementation. We have not used generative AI tools for drafting sections of the paper, research ideation, or proving mathematical claims and generating synthetic datasets is not applicable to this work. Additionally, we used generative AI tools for structuring the code repository and occasional debugging assistance. We have reviewed all AI-assisted work. Specifically, LLM suggestions were used iteratively paragraph-by-paragraph and section-by-section to refine language and improve narrative coherence, while all underlying text and technical ideas remained human-authored. For discovery, AI tools assisted in searching for scattered cross-domain terminology (e.g., duplicate detection in multi-object tracking) and tracing the lineage of ModelNet40 dataset variants (`modelnet40_ply_hdf5_2048` and `modelnet40_normal_resampled`) for calibration; all retrieved references and dataset lineage facts were independently cross-checked and verified against primary sources. Furthermore, all AI-assisted code snippets were tested and validated for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The trajectory experiment evaluates sensitivity on existing benchmark records with synthetic redundant inputs. It does not establish safety of a deployed forecasting system or justify merging records of distinct people. In practical use, mistaken identity association can erase real entities; this risk is different from suppressing known duplicates. No new human-subject data were collected for these audits.

REPRODUCIBILITY STATEMENT

Mathematical assumptions, proofs, and supporting references are provided in Appendices A–E. Appendices F–I detail the experimental protocols, checkpoint contracts, stochastic couplings, statistical aggregation, and expanded results. The supplementary archive contains experimental code, configurations and execution instructions, the frozen point-cloud calibration, and numerical summaries for all three experiments. These summaries permit inspection of the reported results without repeating training or inference. Experiment 2 additionally includes CPU-only reporting tools for regenerating density and refinement figures and extracting tabular results. Full reruns generate the underlying training and audit artifacts needed for analyses beyond the supplied summaries. Public third-party datasets and pretrained checkpoints are obtained through the documented retrieval procedures. Appendix J summarizes the supplementary material and distinguishes inspection of the supplied results from full experimental reproduction.

REFERENCES

Jonathan W Bartlett and Chris Frost. Reliability, repeatability and reproducibility: analysis of measurement errors in continuous variables. *Ultrasound in obstetrics & gynecology*, 31(4):466, 2008.

François Bolley. Separability and completeness for the Wasserstein distance. In *Séminaire de probabilités XLI*, pp. 371–377. Springer, 2008.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Christian Bueno and Alan Hylton. On the representation power of set pooling networks. *Advances in Neural Information Processing Systems*, 34:17170–17182, 2021.

Rossana Capuani, Antonio Marigonda, and Marc Quincampoix. Set-driven evolution for multiagent system. *Journal of Optimization Theory and Applications*, 200(1):280–307, 2024.

Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. PointGPT: Auto-regressively generative pre-training from point clouds. *Advances in Neural Information Processing Systems*, 36:29667–29679, 2023.

Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 1999.

Michel Grabisch, Jean-Luc Marichal, Radko Mesiar, and Endre Pap. *Aggregation functions*, volume 127. Cambridge University Press Cambridge, 2009.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264. IEEE, 2018.

Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte Carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (tog)*, 37(6):1–12, 2018.

Sadok Kallel and Denis Sjerve. Remarks on finite subset spaces. *Homology, Homotopy and Applications*, 11(2):229–250, 2009.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pp. 655–664. Wiley Online Library, 2007.

Eitan Levin, Yuxin Ma, Mateo Díaz, and Soledad Villar. On transferring transferability: Towards a theory for size generalization. *Advances in Neural Information Processing Systems*, 38:67015–67083, 2026.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A simple state space model for point cloud analysis. *Advances in neural information processing systems*, 37:32653–32677, 2024.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations*, 2022.

Saunders Mac Lane. *Categories for the Working Mathematician*. Number V. 5. Springer, 1998.

Anton Milan, Konrad Schindler, and Stefan Roth. Detection-and trajectory-level exclusion in multiple object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3682–3689, 2013.

Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 14412–14420. IEEE, 2020.

Ilya Molchanov. *Theory of random sets*. Springer-Verlag London, 2017.

Giannis Nikolentzos and Konstantinos Skianis. On the lipschitz continuity of set aggregation functions and neural networks for sets. In *International Conference on Learning Representations*, volume 2026, pp. 134278–134321, 2026.

Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pp. 261–268. IEEE, 2009.

Vasco Portilheiro. Representation learning with multisets. In *NeurIPS 2019 Workshop on Sets and Partitions*, 2019.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Christopher Tuffley. Finite subset spaces of s1. *Algebraic & Geometric Topology*, 2(2):1119–1145, 2002.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12, 2019.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9613–9622. Ieee, 2019.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Yinghong Xie, Yongxing Ke, Xiaowei Han, Qiang Gao, and Chongli Wang. UMOT: A unified framework for long-and short-term association for multi-object tracking. *Plos one*, 20(9):e0332709, 2025.

Shengtai Yao, Eitan Levin, and Mateo Díaz. Any-dimensional invariant universality. *arXiv preprint arXiv:2605.23156*, 2026.

Ye Yuan and Kris Kitani. DLow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision*, pp. 346–364. Springer, 2020.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. AgentFormer: Agent-aware Transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9813–9823, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

APPENDIX GUIDE

The appendices separate the mathematical statements from the scope of their empirical tests. Appendix A proves the factorization and fiber results, including the corresponding statement for elementwise outputs. Appendix B distinguishes gradings, filtrations, and directed systems. Appendix C develops the finite-stage topology, the three counterexamples, and the metric completions. Appendix D analyzes aggregation, count-compressed inference, stability bounds for pointwise max aggregation, and the conditions needed for neighborhood and measure-based models. Appendix E discusses physical configurations and sampled geometry. The remaining appendices give the complete experimental methods, expanded numerical results, and reproducibility record.

A ALGEBRA OF REGISTRATIONS AND FIBERS

A.1 THREE MAPS OUT OF THE SAME ARRAY

Assume throughout that $X \neq \varnothing$ and registrations are finite and nonempty. Three useful maps from $\mathcal{T}(X) = \coprod_{n\geq 1} X^n$ are

$$s(x) = \{x_1, \ldots, x_n\}, \qquad \text{support}, \tag{5}$$

$$c(x) = \sum_{i=1}^{n} \delta_{x_i}, \qquad \text{counting measure}, \tag{6}$$

$$e(x) = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}, \qquad \text{empirical probability}. \tag{7}$$

Here Dirac notation can initially be read formally; no measurable structure is necessary for these finitely supported objects. Every map is invariant to coordinate permutation. They induce different equivalence relations. Counting-measure equality preserves both the distinct elements and their multiplicities. Empirical-probability equality preserves only their relative multiplicities. Support equality preserves only whether an element appears.

For example, $x = (a, b)$ and $y = (a, a, b, b)$ have equal support and empirical probability but unequal counting measures. The tuple $z = (a, a, b)$ has the same support and a different empirical probability when $a \neq b$. This example also distinguishes *uniform replication* from *arbitrary duplicate insertion*. A semantic contract must say which of these operations preserves the represented datum.

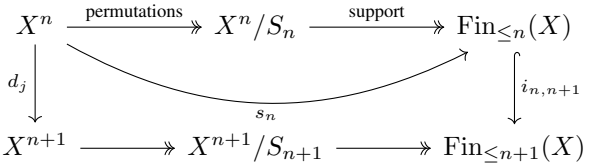


Figure 3: Order and multiplicity are removed in different quotient steps. For every duplicate insertion d_j, the outer support square commutes. There is no canonical choice of an element to duplicate on an unordered orbit; the support inclusion is canonical.

The empty set can be included by adding a grade $n = 0$ and specifying an output on its unique registration. No nonempty duplicate operation connects that grade to the others. Omitting it avoids conventions for empty means, maxima, and empirical probabilities without affecting any experiment.

A.2 A DETAILED FACTORIZATION ARGUMENT

Theorem A.1 (Equivalent formulations of set consistency)**.** *For maps $F_n : X^n \to Y$, the following are equivalent:*

(i) There is a function $f : \mathrm{Fin}(X) \to Y$ with $F_n = f \circ s_n$ for every n.

(ii) If $s_n(x) = s_m(y)$, then $F_n(x) = F_m(y)$.

(iii) Each F_n is permutation invariant and $F_{n+1}(d_j x) = F_n(x)$ for all x, j.

(iv) Each F_n factors through $\mathrm{Fin}_{\le n}(X)$ as $f_n \circ s_n$, and $f_m(K) = f_n(K)$ whenever $|K| \le m \le n$.

The factor f is unique.

Proof. (i)⇒(ii) follows by evaluating both sides at their common support. For (ii)⇒(iii), permutations and duplicate insertion leave support unchanged.

For (iii)⇒(ii), fix a tuple x. If a value occurs more than once, permute one occurrence to the last coordinate while leaving another occurrence among the first $n-1$. The resulting tuple is $d_j x'$ for some shorter tuple x'. Invariance and insertion compatibility give $F_n(x) = F_{n-1}(x')$. Repeating this reduction terminates at a tuple that lists every support element exactly once. Two tuples with the same support reduce to enumerations of the same finite set, and those enumerations differ by a permutation. Their outputs are therefore equal.

For (ii)⇒(i), define $f(K)$ by choosing any finite enumeration of K and evaluating the corresponding F. Statement (ii) makes the value independent of the enumeration and of repetitions. Since every finite nonempty support has an enumeration, this defines f everywhere. It also proves uniqueness. Finally, (i)⇒(iv) follows by restriction, and (iv)⇒(i) follows by gluing the compatible functions f_n on the increasing union. These definitions give the same f. □

Only an adjacent-grade insertion rule is needed; all larger changes are compositions. A deletion rule must preserve at least one occurrence of the deleted value. Deleting the sole occurrence changes support and is not an invariance requirement.

There is a useful separation inside statement (iv). At a single grade, permutation invariance alone does not imply support factorization. For $n = 3$, (a, a, b) and (a, b, b) have the same support but need not be in the same orbit. Even if factorization holds at every grade, compatibility of the resulting f_n is still necessary. The constant-at-each-grade family $F_n(x) = n$ fails precisely at that last step.

A.3 Counting ordered and unordered fibers

Proposition A.2 (Grade-wise fiber cardinalities)**.** *Let $K = \{u_1, \dots, u_r\}$ with distinct elements. For $n < r$ its grade-n fiber is empty. For $n \ge r$,*

$$|s_n^{-1}(K)| = r! \left\{ {n \atop r} \right\} = \sum_{j=0}^{r} (-1)^j \binom{r}{j} (r-j)^n, \tag{8}$$

$$|s_n^{-1}(K)/S_n| = \binom{n-1}{r-1}. \tag{9}$$

For a fixed positive count vector $c = (c_1, \dots, c_r)$, its permutation orbit has $n!/\prod_j c_j!$ elements.

Proof. An ordered registration with support K is a surjection from the n positions to the r named support elements. Partition the positions into r nonempty blocks and label those blocks by the elements of K, giving $r! \left\{ {n \atop r} \right\}$. Alternatively, inclusion–exclusion subtracts functions that omit one or more support elements, giving the second expression in equation 8.

A permutation orbit is uniquely determined by positive multiplicities summing to n. Place $r-1$ dividers in the $n-1$ gaps between n objects; this gives equation 9. For prescribed multiplicities, choose the positions of each support element successively, giving the multinomial coefficient. Summing these orbit sizes over the positive count vectors recovers the ordered count. □

At $n = 100$, an unordered two-element support has 99 registrations, and a three-element support has $\binom{99}{2} = 4,851$. Exhausting these count vectors is exhaustive *within that grade and sampled support*. It is not exhaustive over all supports or registration lengths. This is the precise sense of exhaustiveness in Experiment 1.

Proposition A.3 (Fibers of empirical probability). *Let $\mu = \sum_{j=1}^{r} p_j \delta_{u_j}$, where u_j are distinct, $p_j > 0$ are rational, and $\sum_j p_j = 1$. Let q be the least positive integer with $qp_j \in \mathbb{N}$ for every j. Then the grade-n empirical-measure fiber is nonempty exactly when $q \mid n$. In that case it contains one permutation orbit, of size*

$$\frac{n!}{\prod_{j=1}^{r}(np_j)!}. \tag{10}$$

Proof. Equality $e(x) = \mu$ forces the count at u_j to be np_j. Such counts exist exactly when every np_j is an integer, equivalently when n is divisible by the least common denominator q. They are then unique, so all ordered realizations form the indicated multinomial orbit. □

This fiber is markedly smaller than a support fiber. An empirical-measure model must equate the same relative frequencies across compatible denominators. A support model must additionally equate all positive frequency vectors on the support.

A.4 Restricted fiber defects and what they measure

For a permutation-invariant model, write $F(K; c)$ for its output on multiplicities $c_j > 0$. Let u_K be a once-per-element enumeration of K. For the count set $C_{n,r} = \{c \in \mathbb{N}^r : \sum_j c_j = n\}$, define

$$D_n(K) = \frac{1}{|C_{n,r}|} \sum_{c \in C_{n,r}} d_Y(F(K; c), F_r(u_K)), \tag{11}$$

$$D_n^{\max}(K) = \max_{c \in C_{n,r}} d_Y(F(K; c), F_r(u_K)), \tag{12}$$

$$V_n(K) = \max_{c, c' \in C_{n,r}} d_Y(F(K; c), F(K; c')). \tag{13}$$

The first quantity depends on the chosen distribution over registrations; here it is uniform over positive count vectors. It is not the same as drawing independent copies uniformly from the support and conditioning on observing each element. The latter weights count vectors by multinomial orbit sizes.

The triangle inequality gives $V_n(K) \le 2D_n^{\max}(K)$. But $V_n(K) = 0$ alone does not imply $D_n(K) = 0$: all grade-n outputs can agree with each other while disagreeing with the baseline grade. Nor does a small average D_n imply a small worst-case defect. In the scalar audit, V_n is exactly the registered prediction range, whereas reported maximum and 95th-percentile defects compare to the lower-cardinality baseline.

Approximate factorization has a simple interpretation. Suppose for each K all registration outputs have diameter at most $\delta(K)$. Choosing one representative output defines $f(K)$ with $d_Y(F_n(x), f(K)) \le \delta(K)$ for every registration of that support. Conversely, if some f satisfies a uniform error bound ε on all registrations, every fiber diameter is at most 2ε. These observations do not require a linear output space, but empirical restrictions on n and K restrict the scope of either bound.

A.5 Elementwise outputs and equivariant consistency

Trajectory predictors return one output per original entity. This requires an equivariant version of the contract, rather than equality of output arrays of different lengths. Let $G_n : X^n \to Y^n$. Permutations act by the same coordinate permutation on inputs and outputs; d_j appends the jth coordinate in either space.

Proposition A.4 (Descent of elementwise outputs). *The following are equivalent:*

- *(i) There is an assignment $g(u, K) \in Y$ for every $K \in \mathrm{Fin}(X)$ and $u \in K$ such that $G_n(x)_i = g(x_i, s_n(x))$.*
- *(ii) G_n is permutation equivariant for every n, and $G_{n+1}(d_j x) = d_j G_n(x)$ for every x, j.*

Proof. The assignment in (i) is unaffected by reordering; duplicating an element merely duplicates its assigned output, proving (ii). Conversely, if $x_i = x_j$, transposing i and j fixes the input. Equivariance

forces the corresponding outputs to be equal. Compatibility then allows removal of a duplicate coordinate without changing the outputs attached to the remaining elements. Reduce to a distinct-element enumeration of K as in Theorem A.1. Equivariance makes the resulting output associated with u independent of the enumeration. Define it to be $g(u, K)$ and expand duplicates to recover the original tuple. □

This proposition concerns duplicates of the *entire modeled element*. If agent identity, physical size, or another mark is part of X, equal positions alone do not make two inputs equal. For known redundant records, one may instead define a semantic quotient that identifies their artificial record identifiers while retaining the original entity. Experiment 3 measures the change after restricting the perturbed prediction to the original agents; it does not assume that every implementation satisfies the proposition.

For Experiment 3, take $X = H = (\mathbb{R}^2)^8$ and $Y = V = (\mathbb{R}^2)^{12}$. The input is a tuple of complete observed histories, and the quotient assignment can equivalently be written $g(x_{1:8}, \mathcal{K})$ for $x_{1:8} \in \mathcal{K} = \{x^{(i)}_{1:8}\}$. Comparing two pointed inputs means comparing $(x_{1:8}, \mathcal{K})$ and $(y_{1:8}, \mathcal{S})$ with $y_{1:8} \in \mathcal{S}$, not two sequences of instantaneous occupied sets. The desired identities are summarized by

$$\begin{array}{ccc} H^n & \xrightarrow{\quad G_n \quad} & V^n \\ {\scriptstyle d_j}\downarrow & & \downarrow{\scriptstyle d_j} \\ H^{n+1} & \xrightarrow[\quad G_{n+1} \quad]{} & V^{n+1}. \end{array} \tag{14}$$

The output duplication on the right preserves each original coordinate and assigns the added record its source's output. Appendix E.4 explains why retaining whole histories matters.

A.6 Randomized predictors require a stated coupling

Let $\mathcal{U}$ be an auxiliary probability space of inference random inputs, and let U denote a draw from it (for example, a scene latent code or a collection of Gaussian innovations). For a randomized implementation $F(x, U)$, two contracts are different:

(i) *Pathwise consistency*: $F(x, U) = F(y, U)$ under a specified common-randomness coupling whenever $s(x) = s(y)$.

(ii) *Distributional consistency*: the laws of $F(x, U)$ and $F(y, U)$ agree under their respective inference distributions.

The coupling must have the correct marginal random-input law at each registration; per-agent noise also needs an explicit rule when an agent is duplicated. Pathwise equality under such a coupling implies equality in law. The converse does not hold for an arbitrary coupling. For example, if U is uniform on $[0, 1]$, the outputs U and $1 - U$ have the same law but generally differ when paired by the same U. Thus a common-noise audit is reproducible and informative about that controlled inference path, but nonzero paired changes alone do not establish inequality of the full predictive distributions. This qualification is particularly relevant when proposals themselves depend on the scene, as in the AgentFormer audit.

B Graded registrations, filtered supports, and directed systems

B.1 A grading is not a filtration

The decomposition $\mathcal{T}(X) = \coprod_n X^n$ is a grading: each registration has exactly one length. So does $\mathcal{M}(X) = \coprod_n X^n/S_n$, since permutation quotient preserves length. An element in grade n is not literally an element in grade $n + 1$ until a connecting map is specified.

In contrast, the spaces $\mathrm{Fin}_{\le n}(X)$ form a filtration:

$$\mathrm{Fin}_{\le 1}(X) \subseteq \mathrm{Fin}_{\le 2}(X) \subseteq \cdots. \tag{15}$$

A singleton is present in every bounded stage, and those appearances refer to the same support. The disjoint decomposition of $\mathrm{Fin}(X)$ into exact-cardinality pieces can also be called a grading of the underlying set, but those pieces need not be closed under geometric limits. Two distinct points can collide and leave a singleton.

The support fiber $s^{-1}(K)$ inherits the registration grading:

$$s^{-1}(K) = \coprod_{n \geq |K|} s_n^{-1}(K). \tag{16}$$

This is the origin of the phrase *graded registration fiber*. It records an entire family of implementation-level descriptions above one semantic datum. Forgetting the grade identifies descriptions of different lengths; forgetting order within a grade leaves the positive multiplicity vectors counted in equation 9.

B.2 Support inclusions and their universal property

We use the standard universal-property formulation of a colimit (Mac Lane, 1998, Chapter III). The category and its morphisms are part of the specification. For $m \leq n$, define $i_{m,n} : \mathrm{Fin}_{\leq m}(X) \to \mathrm{Fin}_{\leq n}(X)$ by $i_{m,n}(K) = K$. These maps satisfy $i_{n,n} = \mathrm{id}$ and $i_{n,p} \circ i_{m,n} = i_{m,p}$. Their set-theoretic colimit is $\mathrm{Fin}(X)$ with the evident inclusions. Explicitly, a compatible family $f_n : \mathrm{Fin}_{\leq n}(X) \to Y$ defines a unique function on $\mathrm{Fin}(X)$ because any two definitions of $f(K)$ agree in a stage containing both.

One can realize an inclusion using tuple padding, for example by repeating the last entry of a tuple. Such a padding operation is not a canonical permutation-equivariant choice of a distinguished support element. The inclusion on supports is canonical because all possible choices have the same support. Equivalently, use all surjections $\alpha : [n] \twoheadrightarrow [m]$ to repeat entries:

$$\alpha^*(x)_j = x_{\alpha(j)}. \tag{17}$$

The support is unchanged, and composition of surjections composes repetitions. Requiring consistency for every such map, together with permutation symmetry, is exactly the duplicate contract. A single preferred padding convention should not be confused with that full contract.

B.3 The divisibility system for empirical probabilities

Let $E_n(X) = \{n^{-1} \sum_{j=1}^n \delta_{x_j} : x \in X^n\}$. Within grade n, the empirical map identifies precisely the permutation orbits. If $n \mid m$, repeating every coordinate m/n times induces an inclusion $E_n(X) \hookrightarrow E_m(X)$ that leaves the probability measure unchanged. The divisibility order is directed because $\mathrm{lcm}(m, n)$ is an upper bound.

Its union is

$$\mathcal{P}_{\mathrm{fin},\mathbb{Q}}(X) = \left\{ \sum_{j=1}^r p_j \delta_{u_j} : p_j \in \mathbb{Q}_{>0},\ \sum_j p_j = 1 \right\}. \tag{18}$$

If two empirical registrations give the same measure, lift both to a common multiple of their lengths. Their multiplicities there agree, so they differ by a permutation. Consequently, permutation invariance and uniform-replication compatibility characterize factorization through this empirical-probability quotient, by the same common-stage reasoning as for supports.

The spaces E_n are generally not nested under $n \leq m$. For two distinct elements, $(\delta_a + \delta_b)/2 \in E_2$ does not belong to E_3. A nested cofinal sequence is $E_{1!} \subseteq E_{2!} \subseteq E_{3!} \subseteq \cdots$, since every positive integer divides a sufficiently large factorial. Its union and final topology agree with those of the full divisibility system: continuity or closedness can be checked on the cofinal stages.

Uniform replication is an appropriate invariance when relative frequency is meaningful but array length is an acquisition choice. It is inappropriate if absolute count or confidence from sample size is part of the input. In that case, a representation such as $(n, e(x))$ or $c(x)$ can retain the missing information explicitly.

B.4 WHY "FIBER" DOES NOT ESTABLISH A FIBRATION

A fiber is simply a preimage $s^{-1}(K)$. A fiber bundle or a topological fibration requires additional local triviality or lifting properties. Neither follows from the existence of the support map.

A concrete obstruction to a bundle picture appears for $X = \mathbb{R}$ and s_2. The fiber above $\{a\}$ contains just (a, a), while the fiber above $\{a, b\}$ for $a \neq b$ contains (a, b) and (b, a). Two-point supports approach the singleton as $b \to a$. A neighborhood of that singleton therefore cannot trivialize s_2 with a fixed finite fiber. Passing to unordered registrations changes the count, but collision strata remain structurally different, and higher grades have different multiplicity fibers.

We use the language of a *space over a semantic base with graded fibers*. No local trivialization, connection, parallel transport, or homotopy lifting is invoked in the results. If a specialized application supplies such extra structure, a fibration formulation may be useful; it should be established rather than inferred from terminology.

B.5 A SEMANTIC CHECKLIST IN MATHEMATICAL FORM

A registration-based modeling decision can be organized into five explicit choices:

1. Specify the semantic object B and the registration map $q : \mathcal{T}(X) \to B$.
2. Identify the transformations that preserve q, including their cross-cardinality behavior.
3. Require the implemented predictor to factor through q, or quantify its failure to do so on selected fibers.
4. Choose a topology or metric on B for approximate changes of the datum.
5. Test task performance and stability separately, with a stated registration distribution and numerical control.

For supports, the first three steps yield Theorem A.1. For empirical probabilities, they yield the divisibility system. The fourth step is additional: algebraic equivalence alone does not impose Hausdorff continuity or Wasserstein continuity. The fifth step cannot certify all fibers from a finite experiment, but can reveal inconsistencies hidden by ordinary test accuracy.

C TOPOLOGY, COUNTEREXAMPLES, AND METRIC COMPLETION

This appendix gives self-contained versions of the three counterexamples concerning supports, empirical distributions, and rational grids. We work with ordinary topological colimits equipped with the final topology. We do not infer properties of other categories of spaces without specifying them.

C.1 FINITE-STAGE SUPPORT TOPOLOGY

Proposition C.1 (Support quotient at bounded cardinality)**.** *If X is compact metric, the quotient of X^n by equality of support is homeomorphic to $(\mathrm{Fin}_{\leq n}(X), d_H)$.*

Proof. Equip X^n with the maximum product metric. For two tuples x, y, each x_i is at distance at most $\max_j d(x_j, y_j)$ from $s_n(y)$, and the symmetric statement holds. Hence

$$d_H(s_n(x), s_n(y)) \leq \max_i d(x_i, y_i). \tag{19}$$

The support map is continuous and surjective. Its fibers are exactly the defining equivalence classes, so it induces a continuous bijection from the quotient to $\mathrm{Fin}_{\leq n}(X)$. The domain is compact as a quotient of compact X^n, and the codomain is Hausdorff as a metric space. A continuous bijection from a compact space to a Hausdorff space is a homeomorphism: images of closed sets are compact and therefore closed, giving continuity of the inverse. □

In particular, every bounded stage is compact and closed in the hyperspace of compact subsets. Collisions occur within these stages: $\{a, b_j\} \to \{a\}$ if $b_j \to a$.

C.2 THE FINAL TOPOLOGY ON THE UNION

Let L_{set} be the set $\mathrm{Fin}(X)$ with the final topology for the inclusions of its bounded stages. A subset U is open precisely when $U \cap \mathrm{Fin}_{\leq n}(X)$ is open in every stage. Equivalently, C is closed precisely when all stage intersections are closed. The identity

$$\iota : L_{\mathrm{set}} \longrightarrow (\mathrm{Fin}(X), d_H) \tag{20}$$

is continuous because each stage inclusion into the metric union is continuous. This says the final topology is at least as fine as the metric topology. It need not be equal.

Theorem C.2 (Counterexample: support direct limit)**.** *If X is an infinite compact metric space, there is a set closed in L_{set} whose image is not closed in $(\mathrm{Fin}(X), d_H)$. Thus ι is not a topological embedding, despite being an embedding on every bounded stage.*

Proof. Compactness of an infinite metric space gives distinct points x_j converging to a point x_0, with every $x_j \neq x_0$. For completeness, start with infinitely many distinct points and choose a convergent subsequence; omit any occurrence equal to its limit. Define

$$K_n = \{x_{n+1}, x_{n+2}, \ldots, x_{2n}\}, \qquad C = \{K_n : n \geq 1\}. \tag{21}$$

All x_j are distinct, so $|K_n| = n$. Therefore $C \cap \mathrm{Fin}_{\leq N}(X) = \{K_1, \ldots, K_N\}$ is finite and closed in the metric stage. The final-topology criterion makes C closed in L_{set}.

On the other hand,

$$d_H(K_n, \{x_0\}) = \max_{n<j\leq 2n} d(x_j, x_0) \longrightarrow 0. \tag{22}$$

The singleton $\{x_0\}$ is in $\mathrm{Fin}(X)$ but not in C. Thus C is not closed in the Hausdorff subspace topology. A homeomorphism onto its image would preserve closed sets, so ι cannot be an embedding. □

The unbounded cardinalities are essential in this construction. It does not contradict Proposition C.1; each fixed stage sees only finitely many members of C. If X is finite, the support filtration stabilizes, and this counterexample is unavailable.

C.3 FINITE-STAGE EMPIRICAL-MEASURE TOPOLOGY

For $p \in [1, \infty)$ and Borel probabilities on a compact metric X, define

$$W_p(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \left(\int d(x, y)^p \, \mathrm{d}\pi(x, y) \right)^{1/p}, \tag{23}$$

where $\Pi(\mu, \nu)$ is the set of couplings. Pairing the ith coordinates provides the elementary estimate

$$W_p(e_n(x), e_n(y)) \leq \left(\frac{1}{n} \sum_{i=1}^{n} d(x_i, y_i)^p \right)^{1/p}. \tag{24}$$

The empirical map is continuous. At a fixed grade, equality of the empirical measure is precisely equality of the multiset, so X^n/S_n is continuously bijective to $E_n(X)$. Compactness and the Hausdorff property again imply a homeomorphism. The divisibility inclusions preserve the probability measure and are isometries in W_p.

More precisely, for equal-size empirical registrations, optimal transport is a finite matching problem:

$$W_p(e_n(x), e_n(y)) = \min_{\pi \in S_n} \left(\frac{1}{n} \sum_{i=1}^{n} d(x_i, y_{\pi(i)})^p \right)^{1/p}. \tag{25}$$

Indeed, a coupling is a doubly stochastic matrix after multiplication by n, and a linear cost attains its minimum at a permutation matrix. This remains valid for repeated points by viewing their occurrences as labeled atoms before quotienting.

At a fixed finite dimension d, all ℓ_p norms for $1 \le p < \infty$ induce the same topology. For $1 \le p \le q < \infty$,

$$\|v\|_q \le \|v\|_p \le d^{1/p-1/q}\|v\|_q. \tag{26}$$

For the normalized matching distances in equation 25, this gives $W_p \le W_q \le n^{1/p-1/q}W_p$ at grade n. The finite-dimensional equivalence constants can depend on cardinality; they do not by themselves supply a uniform limit geometry. This distinction is pertinent to size generalization (Levin et al., 2026).

On a compact metric X, every W_p with $1 \le p < \infty$ metrizes weak convergence of probability measures (Villani, 2009, Theorem 6.9). Compactness makes the required moment control automatic. Thus these Wasserstein topologies agree on the entire probability space, not merely at a fixed grade. This statement excludes $p = \infty$: moving a vanishing amount of mass a fixed positive distance can converge weakly while its W_∞ distance stays positive. The final topology of the directed union below is a separate issue from either of these norm-equivalence statements.

Let L_{prob} be $\mathcal{P}_{\mathrm{fin},\mathbb{Q}}(X)$ with the final topology of these stages. The canonical injection into the same set equipped with W_p is continuous, but again not generally an embedding.

Theorem C.3 (Counterexample: empirical-probability direct limit)**.** *Suppose compact metric X contains distinct points a, b. The set of dyadically weighted probabilities on $\{a, b\}$ is closed and discrete in L_{prob}, but is not closed in the W_p subspace topology on $\mathcal{P}_{\mathrm{fin},\mathbb{Q}}(X)$.*

Proof. Write $\mu_t = t\delta_a + (1-t)\delta_b$ and

$$S = \{\mu_t : t \in [0,1] \cap \mathbb{Z}[1/2]\}. \tag{27}$$

The intersection of S with $E_n(X)$ is contained in the finite set $\{\mu_{j/n} : 0 \le j \le n\}$. More strongly, for any subset $A \subseteq S$, $A \cap E_n(X)$ is finite and therefore closed in $E_n(X)$. The final-topology closedness criterion shows that every subset A is closed in L_{prob}. In particular, S is closed and its subspace topology is discrete.

For two such probabilities, the unmatched mass $|t-u|$ must move between a and b, and transporting exactly that mass is feasible. Hence

$$W_p(\mu_t, \mu_u) = d(a,b)|t-u|^{1/p}. \tag{28}$$

Choose dyadic $t_j \to 1/3$. Then $\mu_{t_j} \to \mu_{1/3}$ in W_p, while $\mu_{1/3} \in E_3(X)$ does not belong to S. Thus S is not closed in the metric union. Its metric closure there is exactly $\{\mu_t : t \in [0,1] \cap \mathbb{Q}\}$: dyadics are dense in $[0,1]$, and a limit supported on $\{a, b\}$ remains supported there. □

The failure already occurs on a two-point base. Unlike the support construction, it does not require infinitely many distinct spatial points. Increasing denominators are sufficient.

C.4 Counterexample: discrete rational grids

Example C.4 (A simple direct-limit analogy)**.** Fix an integer $b > 1$. Give each $b^{-n}\mathbb{Z}$ its Euclidean subspace topology, which is discrete. The inclusions for $m \le n$ form a directed system with union $\mathbb{Z}[1/b]$. Every subset of the union has an open intersection with every discrete stage. Consequently the final topology on the union is discrete.

The natural injection of this union into $\mathbb{R}$ is continuous and restricts to an embedding on every stage. It is not an embedding on the union, because $b^{-n} \to 0$ in the Euclidean topology, whereas a sequence of distinct points cannot converge to zero in the discrete topology. For the divisibility-directed system $n^{-1}\mathbb{Z}$, the same argument gives $\mathbb{Q}$ with the discrete final topology, rather than the usual Euclidean subspace topology on the rationals.

The example isolates the logical issue: agreement at each stage does not identify the topology on an infinite union. The final topology is a valid categorical construction; the metric topology is a different valid structure suited to limits of increasingly fine registrations. It would be inaccurate to describe either merely as having "more structure" without stating which category or continuity requirement is intended.

C.5 THE METRIC UNION AND ITS UNIVERSAL PROPERTY

Consider the category of metric spaces with nonexpansive maps (maps with Lipschitz constant at most one). The bounded support stages with Hausdorff metrics embed isometrically in one another.

Proposition C.5 (Metric colimit)**.** *Their colimit in this category is* $(\mathrm{Fin}(X), d_H)$*. The empirical stages under divisibility similarly have metric colimit* $(\mathcal{P}_{\mathrm{fin},\mathbb{Q}}(X), W_p)$*.*

Proof. For supports, take a compatible family of nonexpansive maps f_n to a metric space Y. Compatibility defines a unique map f on the union. For any two finite supports A, B, choose a stage containing both. Nonexpansiveness in that stage gives $d_Y(f(A), f(B)) \le d_H(A, B)$. This proves the universal property. The empirical argument is identical, using a common multiple of the two denominators as the common stage. The metrics agree with the stage metrics by construction. □

The specified morphisms matter. The argument also glues a uniformly L-Lipschitz family into an L-Lipschitz map. Merely knowing that each f_n is continuous or Lipschitz with an unbounded constant does not supply a common modulus on the union.

$$\begin{array}{ccccc}
\mathrm{Fin}_{\le n}(X) & \hookrightarrow & \mathrm{colim}_{\mathrm{Top}}\, \mathrm{Fin}_{\le n}(X) & & \\
\| & & \downarrow{\scriptstyle \iota} & & \\
(\mathrm{Fin}_{\le n}(X), d_H) & \hookrightarrow & (\mathrm{Fin}(X), d_H) & \overset{\text{completion}}{\dashrightarrow} & (\mathcal{K}(X), d_H) \\
\\
E_n(X) & \hookrightarrow & \mathrm{colim}_{\mathrm{Top},|}\, E_n(X) & & \\
\| & & \downarrow{\scriptstyle \jmath} & & \\
(E_n(X), W_p) & \hookrightarrow & (\mathcal{P}_{\mathrm{fin},\mathbb{Q}}(X), W_p) & \overset{\text{completion}}{\dashrightarrow} & (\mathcal{P}(X), W_p)
\end{array}$$

Figure 4: Two bottom-up constructions on compact metric X. The diagrams commute and agree at each finite stage. The vertical maps on the unions are continuous bijections but need not be embeddings (Theorems C.2 and C.3). The bottom rows use a chosen metric and its completion. The empirical system uses divisibility, denoted $|$, and $1 \le p < \infty$.

C.6 COMPLETION BY COMPACT SUPPORTS

The standard hyperspace results (Molchanov, 2017, Theorems D.2 and D.9), together with density of finite subsets, give the following identification.

Theorem C.6 (Hausdorff completion)**.** *If X is complete,* $(\mathcal{K}(X), d_H)$ *is complete and* $\mathrm{Fin}(X)$ *is dense in it. Therefore it is the metric completion of* $(\mathrm{Fin}(X), d_H)$*. If X is compact, $\mathcal{K}(X)$ is compact.*

For noncomplete X, the corresponding completion is $\mathcal{K}(\widehat{X})$, where $\widehat{X}$ is the metric completion of X.

C.7 COMPLETION BY PROBABILITIES

For compact metric X and $1 \le p < \infty$, finitely supported probabilities with rational weights are dense in $(\mathcal{P}(X), W_p)$, and this Wasserstein space is complete. These are standard approximation and completeness results; see Villani (2009, Chapter 6) and Bolley (2008). Thus the metric completion of $(\mathcal{P}_{\mathrm{fin},\mathbb{Q}}(X), W_p)$ is $(\mathcal{P}(X), W_p)$. On compact X, it is also compact, and W_p metrizes weak convergence.

The two completions retain different information: $\mathcal{K}(X)$ describes compact supports, while $\mathcal{P}(X)$ describes probability laws. The latter normalizes total mass to one. Retaining variable mass requires a different registration map and a separately specified metric.

C.8 Uniform near-idempotence and its qualifications

Suppose $A, B \in \mathrm{Fin}(X)$ contain K and every added point lies within ε of K. Then each point of A is within ε of a point of $K \subseteq B$, and symmetrically for B. In this *retained-anchor* setting, $d_H(A, B) \le \varepsilon$, as in Proposition 4.1. More generally, the 2ε bound applies whenever $d_H(A, K) \le \varepsilon$ and $d_H(B, K) \le \varepsilon$ without requiring anchor retention. Consequently,

$$d_Y(f(A), f(B)) \le \omega(2\varepsilon), \quad \text{and with retained } K, \quad d_Y(f(A), f(B)) \le \omega(\varepsilon). \tag{29}$$

If f is L-Lipschitz, replace the right sides by $2L\varepsilon$ and $L\varepsilon$. The constants do not depend on the number of observations.

For compact X, any continuous map $\mathcal{K}(X) \to Y$ is uniformly continuous because its domain is compact. Such a map supplies a modulus on all finite supports. A continuous function on a noncompact finite-support union need not be uniformly continuous; continuity on each fixed stage is weaker still. The assumptions behind a cardinality-independent stability bound must therefore be stated.

Finally, a finite surface sample is not exactly the source surface, and independent samples generally have different supports. The relevant geometric statement is conditional: if $d_H(A, S) \le \varepsilon_A$ and $d_H(B, S) \le \varepsilon_B$, a Hausdorff-continuous representation of supports satisfies the corresponding modulus bound with $\varepsilon_A + \varepsilon_B$. An experiment that estimates coverage against a finite reference only approximates these premises. It does not prove them for every point on a continuum surface.

D Aggregation, attention, and count-compressed inference

D.1 Structural identities and learned behavior

Consider a shared pointwise feature map $h : X \to \mathbb{R}^d$ and a decoder $\rho : \mathbb{R}^d \to Y$. For a support $K = \{u_1, \dots, u_r\}$ and positive multiplicities c_j summing to n, the common pooling rules are

$$P_{\max}(K; c) = \max_j h(u_j), \tag{30}$$

$$P_{\mathrm{sum}}(K; c) = \sum_j c_j h(u_j), \tag{31}$$

$$P_{\mathrm{mean}}(K; c) = \sum_j (c_j/n) h(u_j), \tag{32}$$

$$P_{\mathrm{LME},\tau}(K; c) = \tau \log\left(\frac{1}{n}\sum_j c_j \exp(h(u_j)/\tau)\right), \qquad \tau > 0, \tag{33}$$

with max, exponential, and logarithm coordinatewise. Max removes multiplicity structurally. Sum records it directly. Mean and log-mean-exp remove common scaling of all counts but generally retain relative multiplicities. A decoder can sometimes cancel a pooling difference, so non-invariance of the intermediate pooling operator is not by itself a proof of non-invariance for every parameter choice of $\rho \circ P$.

Classical diagonal idempotence, such as $\mathrm{mean}(x, \dots, x) = x$, is insufficient for arbitrary duplicate insertion: $\mathrm{mean}(a, b)$ and $\mathrm{mean}(a, a, b)$ differ when $a \ne b$. This is why we use the explicit phrase *idempotent set consistency* and a cross-cardinality definition.

D.2 A Hausdorff bound for pointwise max

Proposition D.1. *Suppose each coordinate h_j is L_h-Lipschitz on X. Then $P(K) = \max_{x \in K} h(x)$ satisfies*

$$\|P(A) - P(B)\|_\infty \le L_h d_H(A, B). \tag{34}$$

If ρ is L_ρ-Lipschitz from the maximum norm to d_Y, then $\rho \circ P$ is $L_\rho L_h$-Lipschitz in Hausdorff distance. If ρ is instead specified in the Euclidean norm, a bound is $L_\rho \sqrt{d}\, L_h$.

Proof. For $a \in A$ choose $b \in B$ within $d_H(A, B) + \eta$, where $\eta > 0$. For each coordinate, $h_j(a) \le \max_{b' \in B} h_j(b') + L_h(d_H(A, B) + \eta)$. Take the maximum over a, let $\eta \downarrow 0$, and reverse A, B. This bounds every coordinate difference. Apply the decoder Lipschitz bound and, if needed, $\|v\|_2 \le \sqrt{d}\|v\|_\infty$. □

This proposition assumes a pointwise feature map independent of the other registered elements. A full point-cloud network may include input-dependent transformations, neighborhood selection, normal estimation, or hierarchy construction. Applying the proposition to its final max layer alone would not certify the complete pipeline.

D.3 Log-mean-exp and finite-temperature approximation

For scalar features $v_1, \dots, v_n$ and $M = \max_i v_i$,

$$M - \tau \log n \le \tau \log \left(n^{-1} \sum_i e^{v_i/\tau} \right) \le M. \tag{35}$$

The upper bound uses $e^{v_i/\tau} \le e^{M/\tau}$; the lower bound keeps one maximizing term. Thus LME approaches max as $\tau \downarrow 0$ at each bounded registration size, but this elementary estimate is not uniform over unbounded n. Multiplicity shifts the fraction of near-maximal features in the normalized exponential average.

The temperatures in Experiment 1 apply to *learned latent features*, not directly to the raw target values. Their effective sharpness depends on the encoder's feature scale. The observed ordering of defects at $\tau = 0.1$ and 1 is an empirical result for those trained networks, not a parameter-free ordering of all possible LME models.

D.4 Exact duplicate grouping in attention

Let distinct keys k_j have values v_j and multiplicities $c_j > 0$. For a query q, let ℓ_j be its scaled dot-product score. Expanding the repeated keys gives

$$\mathrm{Attn}(q; K, c) = \frac{\sum_{j=1}^r c_j e^{\ell_j} v_j}{\sum_{j=1}^r c_j e^{\ell_j}} = \sum_{j=1}^r \mathrm{softmax}_j(\ell + \log c)\, v_j. \tag{36}$$

Uniformly multiplying c by a positive constant cancels from the quotient. Nonuniform count changes act as additive log-count biases and need not cancel. This is a statement about the operator at fixed keys, values, and queries. The full network can learn a function for which the final change is small or vanishes on a domain.

A simple attention construction also illustrates why approximate maximum behavior is possible. With scalar input keys x_j, values x_j, and scores βx_j, the weighted average

$$A_\beta(x) = \frac{\sum_j e^{\beta x_j} x_j}{\sum_j e^{\beta x_j}} \tag{37}$$

approaches the maximum for every fixed finite tuple as $\beta \to \infty$. At finite β, duplicate weights still matter; the rate depends on multiplicities and value gaps. This distinction between approximation capability and exact semantic invariance is what the maximum-regression audit is designed to expose.

D.5 Equivalence of compressed and explicitly repeated computation

The audit need not allocate a length-100 array for each count vector. Pooling can be evaluated with the weighted formulas above. For the tested Set Transformer, equation 36 applies to every attention block.

Proposition D.2 (Count-compressed deterministic attention)**.** *Consider a deterministic network composed of shared pointwise maps, self-attention blocks, rowwise residual operations, and attention pooling with fixed seed queries. Assume no positional features, no independently sampled rowwise noise or dropout, and no operation that distinguishes identical copies. Replacing repeated rows by one row with its count, and adding* $\log c_j$ *to each key score, gives exactly the expanded output in real arithmetic.*

Proof. Initially identical rows have identical features. Suppose this remains true before a block. Identical query rows then have the same scores against each key class. Grouping the expanded softmax numerator and denominator by equal keys gives equation 36, so each query class has the same result as in the expanded computation. Shared pointwise maps and residual additions preserve equality within each class. Induction proves the assertion through all self-attention blocks. The same grouping applies to the final fixed seed queries in attention pooling. A deterministic decoder preserves the resulting equality. □

The implementation uses two self-attention blocks and one learned pooling seed, with width 64, four heads, and the public MAB score convention dividing by $\sqrt{64}$. Layer normalization and dropout are disabled. Count compression changes computational cost, not the intended multiplicity semantics: the count bias is retained precisely so that the audit represents repeated observations. Omitting it would incorrectly deduplicate the model under examination.

Floating-point arithmetic can break exact equality between batched, expanded, and compressed evaluation through rounding and kernel selection. Max's very small measured defects in Experiment 1 are consistent with this computational distinction, even though its pooling identity holds algebraically. The tables do not promote those residuals to a substantive failure of max idempotence.

D.6 Max aggregation and geometric stability

Nikolentzos & Skianis (2026) relate the continuity of set aggregation to the chosen input metric and derive Lipschitz bounds for networks built from these operators. Their Hausdorff analysis of max aggregation supplies the relevant insight for PointNet-style encoders (Qi et al., 2017a): Lipschitz pointwise features followed by coordinatewise max admit a stability bound independent of the number of registered observations. The elementary bound in Appendix D.2 records this fact in our notation. Exact duplicate consistency follows from the max identity, while near-duplicate stability additionally uses the feature map's continuity.

This perspective explains why support coverage matters. If finite registrations K and L both approximate a compact surface S, then $d_H(K, L) \le d_H(K, S) + d_H(L, S)$; a Hausdorff-continuous encoder therefore relates their representation discrepancy to geometric coverage. Cardinality alone does not control coverage under an arbitrary acquisition law. Experiment 2 measures both geometric discrepancies and model responses for this reason.

A bound for an aggregation block must be distinguished from a numerical certificate for a released checkpoint. Learned alignment transforms, preprocessing, and inference mode belong to the complete map, and logit continuity does not itself guarantee an unchanged argmax label. We use the cited analysis to motivate the audit and do not estimate checkpoint-specific Lipschitz constants.

D.7 Hard neighborhoods: local regularity versus a global bound

PointNet++ combines FPS centers, radius neighborhoods, and local max aggregation (Qi et al., 2017b). A fixed physical radius is a geometric inductive bias, but neither its use nor fixing the FPS seed proves global Hausdorff continuity. FPS and kNN can switch selections at ties; a hard radius test can change membership at the boundary. Randomness is one possible source of variation, not the only obstruction.

Example D.3 (A hard ball query can be discontinuous)**.** For finite $K \subset \mathbb{R}$ containing zero, define $Q(K) = \max(K \cap [-1, 1])$. For $0 < \varepsilon < 1$, put $K_\varepsilon = \{0, 1 - \varepsilon\}$ and $L_\varepsilon = \{0, 1 + \varepsilon\}$. Then $d_H(K_\varepsilon, L_\varepsilon) = 2\varepsilon$, while

$$|Q(K_\varepsilon) - Q(L_\varepsilon)| = 1 - \varepsilon. \tag{38}$$

Thus continuous pointwise features followed by max do not repair an arbitrary hard neighborhood boundary.

This is a counterexample for an operator, not a proof that a particular trained classifier realizes that jump: downstream weights might cancel it. Conversely, on a fixed-cardinality region with strict selection margins, fixed ordering, and fixed neighborhood identities, a composition of Lipschitz feature operations is locally Lipschitz. A uniform cross-cardinality theorem would additionally need control of neighborhood changes, repetitions, and constants across these regions.

PointMLP uses residual MLPs, FPS/kNN grouping, geometric affine normalization, and local aggregation (Ma et al., 2022). DGCNN recomputes kNN in learned feature space (Wang et al., 2019). PointMamba and PointGPT combine geometric grouping with sequence/state-space or attention processing (Liang et al., 2024; Chen et al., 2023). For each, geometry-dependent selection or serialization must be included in a hyperspace analysis. The absence of a guarantee here is not a theorem that these architectures cannot learn stable functions.

D.8 A CONDITIONAL MEASURE-STABILITY STATEMENT FOR DENSITY CORRECTION

PointConv and Monte Carlo convolution motivate separating surface geometry from the observation density (Wu et al., 2019; Hermosilla et al., 2018). Let σ be the unnormalized surface-area measure on a compact surface S, with $0 < \sigma(S) < \infty$. If samples follow $\nu = p\sigma$, where $p > 0$ and $\int p \, \mathrm{d}\sigma = 1$, then an idealized integral feature has the importance-sampling identity

$$T_\sigma(x) = \int_S k(x, y)\psi(y) \, \mathrm{d}\sigma(y) = \mathbb{E}_{Y\sim\nu}\left[\frac{k(x, Y)\psi(Y)}{p(Y)}\right]. \tag{39}$$

A change of p leaves this target integral unchanged. Finite-sample estimates need not agree, and neither their variance nor their numerical stability follows from the identity alone.

For finite nonnegative measures on a common compact metric domain D, define

$$d_{\mathrm{BL}}(\mu, \nu) = \sup_{\|f\|_\infty + \mathrm{Lip}(f) \le 1} \left| \int_D f \, \mathrm{d}\mu - \int_D f \, \mathrm{d}\nu \right|. \tag{40}$$

This formulation retains mass; it does not normalize every surface to area one. For fixed integrands $g_x : D \to \mathbb{R}^q$ whose coordinatewise bounded-Lipschitz norms are at most C, uniformly over evaluation locations x, define $T_\mu(x) = \int_D g_x(y) \, \mathrm{d}\mu(y)$. The standard dual-norm inequality applied to integration on bounded-Lipschitz functions gives

$$\sup_x \|T_\mu(x) - T_\nu(x)\|_\infty \le C d_{\mathrm{BL}}(\mu, \nu). \tag{41}$$

For the underlying operator/dual-norm estimate, see Folland (1999, Sections 5.1–5.2, especially Eq. (5.3)).

The fixed-integrand assumption matters for a density-aware network. If the integrand also depends on the input measure, that dependence requires its own control. A discrete PointConv pipeline additionally uses neighborhood selection, estimated densities, and learned weights; inverse-density factors require suitable lower bounds. Equation 41 thus explains the stability of a continuous integral surrogate under stated assumptions, without asserting a surface-measure Lipschitz guarantee for the audited checkpoint.

D.9 INTERPRETING LEARNED CONSISTENCY WITHOUT AN ARCHITECTURE RANKING

The max identity is structural. Set Transformer and finite-temperature LME instead admit accurate approximations to max over bounded domains and registration ranges. The bound $\tau \log n$ and the log-count term in attention explain how small defects can coexist with multiplicity-sensitive operators. Learned feature scales and value gaps affect the effective approximation; the aggregate defects do not identify a unique mechanism inside the trained network.

The same distinction applies to point-cloud accuracy. A small class-error change can coexist with a measurable representation change, and a large theoretical upper bound can coexist with good empirical agreement. In Experiment 2, paired effects and intervals measure the particular checkpoints under defined acquisition laws. They neither recover global Lipschitz constants nor establish a universal reliability order. The combination of architectural analysis, input geometry, and measured within-object changes is more informative than treating any one of them as a substitute for the others.

E GEOMETRY, OCCUPANCY, AND PHYSICAL CONFIGURATIONS

E.1 FINITE REGISTRATIONS AND APPLICATION-LEVEL CONTINUUM STRUCTURE

Our starting point is bottom-up: an implementation receives finite registrations, and the intended equivalence relation determines their semantic quotient. A continuum surface or a physical multi-

agent configuration is additional modeling information supplied by the application. The topology counterexamples in Appendix C show why permutation invariance, finite-dimensional continuity, and compatible algebraic maps do not by themselves recover the geometry of such a continuum.

The same finite array can therefore represent different data depending on the application. It may encode a finite support, an empirical distribution, a counting measure, or a collection of physical entities. These choices need not be mutually exclusive. In particular, for a compact two-dimensional CAD surface $S \subset \mathbb{R}^3$ of positive finite area, support semantics and surface-measure semantics are naturally complementary. One may retain the compact support S, its geometrically induced surface-area measure

$$\sigma_S = \mathcal{H}^2|_S,$$

or the joint object (S, σ_S). The support records where geometry is present, whereas σ_S records the surface area carried by subsets of that support.

The geometric surface-area measure should be distinguished from both an observation law and a material mass distribution. Let

$$\nu_\lambda = \frac{w_\lambda}{\int_S w_\lambda \, d\sigma_S} \, \sigma_S, \qquad w_\lambda > 0,$$

denote a point-acquisition law on the same surface. Changing w_λ changes where observations are concentrated without moving the support S or changing its geometric area measure σ_S. The normalization belongs to the acquisition law: σ_S itself is not normalized, and $\sigma_S(S)$ is the geometric surface area in the chosen coordinate frame.

Nor does σ_S describe the material substance of the object. Two geometrically identical objects made, for example, from different materials can have the same (S, σ_S) while having different mass distributions. A physical mass measure requires additional information, such as material density, and is not determined by the embedded surface geometry alone. Observation counts, sampling probabilities, quadrature weights, geometric surface area, and physical mass are therefore different quantities.

This distinction is important for Experiment 2. Exact repetition of an already-observed point is mathematically useful for defining set consistency but is artificial as a model of independent acquisition from a continuous surface: independently sampled observations generally occur at different locations. The relevant geometric relaxation is therefore near-idempotence. Finite registrations of the same source surface need not have identical supports, but when they approximate the same geometry their discrepancy can be measured in Hausdorff distance. A Hausdorff-continuous support representation should then change little under sufficiently small geometric perturbations.

The density interventions in Experiment 2 additionally vary the observation law while attempting to control loss of support coverage. They do not change the intended CAD object or its geometric surface-area measure. Accordingly, the experiment primarily audits support-level near-idempotence through Hausdorff geometry, while its density controls also probe sensitivity to changes in the empirical distribution of observations.

E.2 The indistinguishability assumption and latent physical state

Capuani et al. (2024, Section 5) distinguish knowledge of the distribution of carried mass from the weaker knowledge of the region in which some mass is present. Indistinguishability of carriers at the same position supports a measure description; dependence only on occupied locations further removes concentration. These are separate modeling assumptions. The set-driven model describes how a typical agent's admissible motion depends on its own position and the occupied set of the crowd.

For forecasting, an entity has an underlying physical state that can include body extent, mass, intent, velocity, and other unobserved variables. A position history is an observation of that state, not a complete specification of it. The audit perturbs those observed positions and asks whether redundant observation records change the represented configuration. This adapts the occupancy viewpoint to a finite-history predictor; it does not assume that the measured coordinates exhaust the pedestrian's physical state.

Under the intended redundant-record semantics, two records carrying the same observed history of the same entity provide no new occupancy or entity. Their artificial registration labels should not

create another interacting pedestrian. This assumption motivates idempotent context aggregation and permutation-equivariant individual outputs. It does not assert that two distinct physical pedestrians with equal observed histories must have identical latent states. If mass, count, physical identity, or an additional mark is relevant and available, it belongs in the modeled element or its association structure.

E.3 Configurations compress bodies; surface clouds sample geometry

A configuration deliberately replaces an extended body by one modeled point or trajectory. The represented body still has mass and volume, and its motion remains constrained by surrounding geometry. Adding a physical body changes that configuration; adding a second record of the same body changes only its registration. In a surface cloud, by contrast, the individual observations sample an extended object's geometry: an additional point is normally a new observation location rather than another body.

This distinction explains why density and duplication require different controls in Experiments 2 and 3. A point-cloud resample generally changes the finite support while approximating the same geometric source. An exact duplicate history preserves the finite history support itself. Near-duplicate histories model uncertain redundant records; their neutrality is approximate and depends on the association assumption. Neither interpretation licenses merging distinct people merely because they are close in space.

E.4 Sets of histories are not sequences of occupied sets

Let $H = (\mathbb{R}^2)^8$ and $V = (\mathbb{R}^2)^{12}$. A scene is a registration $\mathbf{x} = (x^{(1)}_{1:8}, \dots, x^{(n)}_{1:8}) \in H^n$, with history support

$$\mathcal{K} = \{x^{(i)}_{1:8} : 1 \le i \le n\} \subset H. \tag{42}$$

Each element retains the temporal association of one agent's observations. Its per-time occupied sets $E_t = \{x^{(i)}_t : i = 1, \dots, n\}$ lose that association. Thus $\mathcal{K}$ must not be replaced by a notation suggesting only $(E_1, \dots, E_8)$.

Already at two times, histories $\{(-1,-1),(1,1)\}$ and $\{(-1,1),(1,-1)\}$ on a line have identical occupied sets at both times and different histories. Embed the line in $\mathbb{R}^2$ to obtain the same example for positions. Between observation times, these histories admit continuous interpolations with different motion. In continuous time, Capuani et al. (2024, Proposition 3.1) establish, under regularity assumptions, a superposition description $E(t) = \{\gamma(t) : \gamma \in \mathcal{A}\}$ for a set-valued tube. This relates trajectories to occupancy; it does not uniquely recover which trajectory produced which sequence of positions.

We therefore use the finite history space H and the multiagent map $G_n : H^n \to V^n$. For any permutation π, the desired equivariance is $G_n(\pi\mathbf{x}) = \pi G_n(\mathbf{x})$. Duplicate consistency additionally requires $G_{n+1}(d_j\mathbf{x}) = d_j G_n(\mathbf{x})$, as in Proposition A.4. This includes preservation of predictions for the original agents. The theory does not require construction of a probability measure on an infinite-dimensional space of entire multiagent paths.

Physical trajectories nevertheless remain continuous underlying objects. The benchmark's nominal 0.4-second spacing discretizes them; eight observations span seven such intervals, while twelve future steps extend 4.8 seconds beyond the last observation. Sampled histories preserve more temporal information than instantaneous occupancy, but do not determine arbitrary behavior between frames.

E.5 Near histories and an equivariant modulus

Equip H with $d_{\mathrm{hist}}(x,y) = \max_{t \le 8} \|x_t - y_t\|_2$, and use the induced Hausdorff metric, denoted $d^{\mathrm{hist}}_{\mathrm{set}}$, on finite subsets of H. On pointed history sets define

$$\begin{aligned} \mathcal{D} &= \{(x_{1:8}, \mathcal{K}) : \mathcal{K} \in \mathrm{Fin}(H),\ x_{1:8} \in \mathcal{K}\}, \\ d_{\mathcal{D}}((x,\mathcal{K}),(y,\mathcal{S})) &= \max\{d_{\mathrm{hist}}(x,y), d^{\mathrm{hist}}_{\mathrm{set}}(\mathcal{K},\mathcal{S})\}. \end{aligned} \tag{43}$$

The distinguished history specifies the agent whose future is predicted. If the descended predictor $g(x,\mathcal{K})$ has modulus ω in this metric and redundant histories are added within ε of $\mathcal{K}$, with all original

histories retained, then

$$\|g(x, \widetilde{\mathcal{K}}) - g(x, \mathcal{K})\|_{V,\infty} \le \omega(\varepsilon), \qquad x \in \mathcal{K}, \tag{44}$$

where $\|v\|_{V,\infty} = \max_{t \le 12} \|v_t\|_2$. This is a conditional stability statement for the full equivariant predictor, not just its context pool.

The Gaussian corruption in Experiment 3 is independent across observed coordinates and times; it changes apparent velocities as well as positions. Its σ is a noise standard deviation, not a deterministic radius in d_{hist}. The reported audit statistics measure the resulting changes in ADE and FDE under the specified inference couplings.

E.6 Record identifiers, entity association, and physical weights

A tracking pipeline can assign distinct track identifiers to the same pedestrian. A registration identifier, an entity identity, and a latent physical state are different notions. Synthetic duplicates make the source-entity association known by construction. Identifying duplicate tracks in tracker outputs requires an association decision, which near-idempotent aggregation alone does not provide. Our audit isolates downstream sensitivity to known redundant records from the separate problem of identifying those records.

Similarly, duplicating a record of a massive body does not double its mass. Two real bodies sharing a modeled position may require separate mass or identity marks. In surface integration, quadrature weights compensate for observation density rather than representing material mass. Coverage anchors in Experiment 2 are a sampling design, not independent draws from the density law, so ordinary importance weights cannot be assigned to them by silently treating every point as an i.i.d. observation.

F EXPERIMENT 1: COMPLETE MAXIMUM-REGRESSION AUDIT

F.1 SEMANTIC CONTRACT AND AUDIT OBJECTIVE

The controlled task is maximum regression, motivated by Lee et al. (2019). The benchmark target is $f(K) = \max K$ for a finite nonempty support $K \subset [0, 100]$. For the purpose of the audit, however, we distinguish knowledge of the target's analytic form from knowledge of its representation semantics. We assume that the semantic checklist has established that repeated registration of an already-present value does not change the represented datum or its target. Equivalently, the target factors through finite support. The practitioner need not know that the unknown regression function is a maximum in order to require this support-level consistency.

The experiment therefore asks two different questions. First, can a trained model predict the target accurately on ordinary fresh examples? Second, does the learned variable-cardinality implementation respect the support identification that has already been specified by the problem? The second question is not inferred from task accuracy: two models can fit the same regression target while behaving differently on alternative registrations of the same support.

The training distribution uses registration lengths at most 10. The consistency audit instead considers total registration length 100 on supports containing only two or three distinct elements. This is an extrapolation in registration grade, but not in support cardinality or target semantics. A length-100 registration of a two-element support still represents the same two-element set, and every positive multiplicity profile on that support has exactly the same target. Accordingly, the audit stresses whether the learned implementation extends the prescribed support equivalence across registration grades, rather than asking it to solve a new 100-element set problem.

The task is an extended controlled reconstruction of the maximum-regression study in Lee et al. (2019), not an exact rerun of its original training schedule. We use a longer training budget, explicit checkpoint selection, common held-out test banks, and an exhaustive multiplicity audit.

F.2 DATA GENERATION AND TRAINING

Each training update chooses one cardinality uniformly from $\{1, \dots, 10\}$ for the entire minibatch of 128 examples. Each example contains independent uniform values in $[0, 100]$, and its target is their maximum. Because the sampling law is continuous, exact duplicates occur with probability zero. The training data therefore do not directly supervise agreement between alternative registrations of the same support, and we use no duplicate augmentation or explicit consistency regularizer. The support-level identification tested below comes from the stipulated semantic contract, not from an empirical pattern inserted into the training distribution.

All six architectures use Adam with a constant learning rate of 10^{-3} and an absolute-error objective for 200,000 updates. Ten runs use seed indices 0–9. The released configuration separates random streams with base seeds 20260923 (training), 430001 (initialization), 530001 (checkpoint probes), 630001 (final test), and 730001 (audit supports). Device selection is automatic in the training configuration; hardware timing is not an experimental outcome here.

Every 1,000 updates, a fresh probe evaluates 10,000 examples for each cardinality, 100,000 examples in total. The scalar probe MAE averages these ten equally sized cardinality groups. For each model/seed, the selected checkpoint minimizes this online probe MAE over the 200 checkpoints. It is then evaluated on a separate final bank of 100,000 examples per cardinality, shared across model comparisons. Final test errors are not used for checkpoint selection. A distinct comparison evaluates the checkpoint at update 20,000 on the same final bank.

The reporting script checks that every selected step agrees with the minimum recorded probe error, that all 60 runs are present, and that all 600 model/seed/cardinality final-test rows are available. The online probe errors should not be treated as an independent final test after checkpoint selection.

F.3 ARCHITECTURES

The five pooling models share the same pointwise encoder and scalar decoder. The encoder is $1 \to 64 \to 64 \to 64 \to 64$, with ReLU after the first three linear layers and a final linear feature layer.

The decoder is $64 \to 64 \to 1$, with an intermediate ReLU. Each has 16,833 parameters. Pooling is coordinatewise mean, sum, max, or LME at $\tau = 0.1$ or 1.

The Set Transformer first applies a $1 \to 64 \to 64$ pointwise encoder with ReLU after both layers. It uses two width-64 self-attention blocks, four heads per block, one learned pooling seed, and a $64 \to 64 \to 1$ decoder. This configuration has 58,497 parameters. The implementation follows the public MAB layout with scores divided by the square root of the full value width, not the per-head width, and uses projected-query residual connections. Layer normalization and dropout are disabled. These details matter for reproducing the experiment and the count-compression equivalence.

The architecture sizes are intentionally reported: the experiment is not an equal-parameter comparison. Its primary question concerns task accuracy and registration behavior under common data and training budgets, rather than attribution of differences to a single operator in isolation.

These architectures deliberately span different relationships to the stipulated support semantics. Coordinatewise max satisfies duplicate insertion structurally; mean, sum, and finite-temperature LME generally retain multiplicity information; Set Transformer has no corresponding structural identity but can learn a nearly support-consistent function. The experiment tests these distinctions at the level of trained implementations.

Model	Parameters	Selected step (10^3)	MAE at 20k	Selected MAE
Mean	16,833	190.5	2.594 ± 0.747	0.769 ± 0.069
Sum	16,833	178.3	1.616 ± 0.292	0.278 ± 0.051
Max	16,833	128.9	0.737 ± 0.597	0.005 ± 0.002
LME ($\tau = 0.1$)	16,833	156.1	0.477 ± 0.321	0.034 ± 0.008
LME ($\tau = 1$)	16,833	181.0	0.539 ± 0.365	0.080 ± 0.007
Set Transformer	58,497	149.0	1.219 ± 0.919	0.016 ± 0.009

Table 4: Training and checkpoint selection. Selected step is the mean across seeds, expressed in thousands of updates. Errors are mean $\pm$ sample SD over ten seeds. The 20k checkpoint is evaluated separately and is not the full training budget.

F.4 Registration-grade stress and exhaustive enumeration and computation

For each support size $r \in \{2, 3\}$, draw 100 distinct supports from the fixed audit stream and sort each support into $u_1 < \cdots < u_r$. These supports are shared across all model/seed combinations. Enumerate every positive integer count vector with total 100. The baseline prediction is the network's ordinary forward pass on the r distinct values; the repeated-registration prediction is evaluated using count-compressed pooling or attention.

For each support, record the mean, 95th percentile, and maximum of the absolute change from the baseline, the prediction range within the repeated-registration grade, the mean absolute task error relative to u_r, and the maximum task error. Count compression is algebraically equivalent to expanded deterministic evaluation under the assumptions in Appendix D. Its purpose is to evaluate the repeated registrations efficiently while retaining their multiplicities.

There are $60 \times 100 \times (99 + 4,851) = 29,700,000$ registered scalar predictions in the exhaustive count audit. The reporting tables contain 12,000 support-level rows and 60 run-level rows. Individual count-vector predictions are not included in the archived summaries; accordingly, the paper does not reconstruct a curve for a particular count vector that is absent from those files.

F.5 Aggregation and uncertainty

For a fixed model and seed, we first average each support statistic over its 100 supports. We then report the mean and sample standard deviation over the ten training seeds. The statistics at different count vectors on one support are deterministic evaluations of the same trained model and are not independent experimental replicates. Treating the 29.7 million registrations as independent observations would severely overstate precision.

The reported standard deviation describes variation over initialization/training seeds on the shared audit bank. It is not a confidence interval over all possible supports. The distribution plots below pool

the support-level measurements only to show heterogeneity, without inferential error bars. Complete run-level values are provided in Tables 7 and 8.

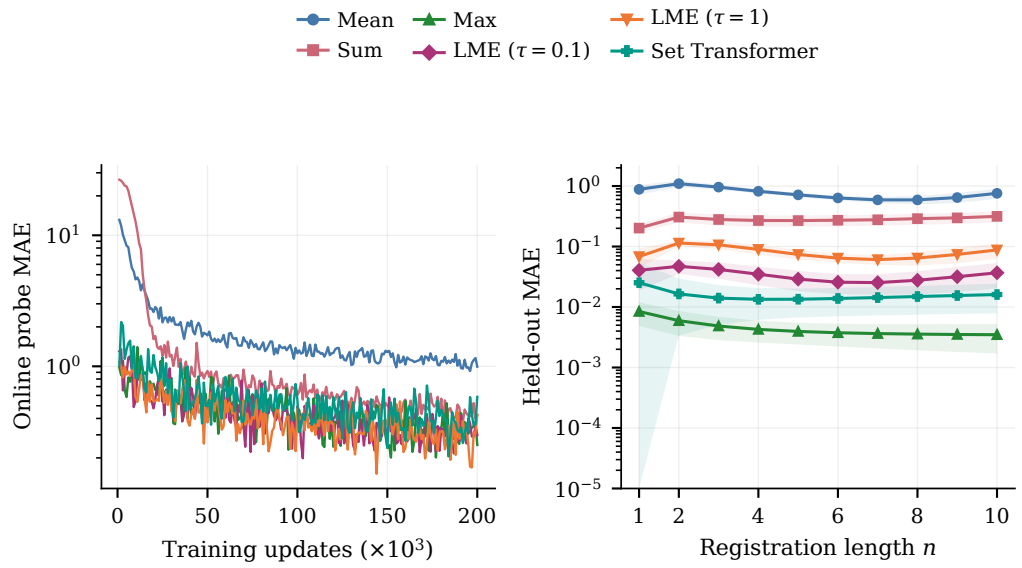


Figure 5: Left: mean online probe MAE across ten seeds at each update. Right: selected-checkpoint held-out MAE by cardinality, with a one-standard-deviation band over seeds. Both vertical axes are logarithmic. The right-hand data are independent of checkpoint selection.

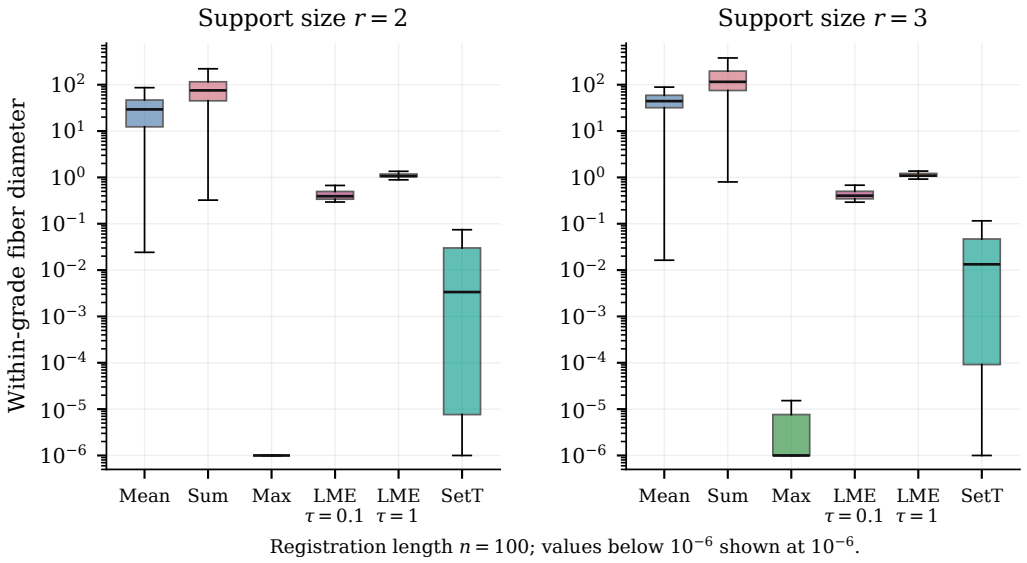


Figure 6: Distribution of within-grade fiber diameters across 100 supports and ten seeds per model. Boxes show quartiles, the line is the median, and whiskers use the standard 1.5-IQR rule; outlier markers are omitted. Values below 10^{-6} are placed at 10^{-6} for display only. The logarithmic axis reveals the scale separation without changing the tabulated values.

Model	Mean defect	95th pct.	Max defect	Diameter	Task MAE
Mean	2.707 ± 0.132	14.795 ± 1.274	29.764 ± 0.755	30.744 ± 0.691	2.916 ± 0.204
Sum	33.517 ± 8.906	64.564 ± 18.667	81.122 ± 20.356	91.399 ± 21.313	33.522 ± 8.900
Max	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.006 ± 0.003
LME ($\tau = 0.1$)	0.061 ± 0.018	0.201 ± 0.056	0.370 ± 0.101	0.424 ± 0.121	0.076 ± 0.020
LME ($\tau = 1$)	0.159 ± 0.013	0.520 ± 0.050	0.950 ± 0.093	1.096 ± 0.097	0.195 ± 0.016
Set Transformer	0.002 ± 0.003	0.009 ± 0.010	0.020 ± 0.024	0.022 ± 0.026	0.017 ± 0.013

Table 5: Two-element support audit, mean $\pm$ sample SD over ten seeds after support averaging. "95th pct." and "Max defect" are first computed within each support's 99 registrations.

Model	Mean defect	95th pct.	Max defect	Diameter	Task MAE
Mean	3.447 ± 0.230	18.281 ± 1.538	43.690 ± 1.493	44.937 ± 1.427	3.492 ± 0.224
Sum	27.716 ± 8.000	63.145 ± 18.907	115.933 ± 32.435	144.570 ± 36.149	27.743 ± 8.017
Max	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.005 ± 0.002
LME ($\tau = 0.1$)	0.072 ± 0.020	0.227 ± 0.061	0.345 ± 0.088	0.435 ± 0.119	0.080 ± 0.022
LME ($\tau = 1$)	0.186 ± 0.016	0.576 ± 0.055	0.880 ± 0.092	1.123 ± 0.099	0.208 ± 0.019
Set Transformer	0.004 ± 0.004	0.014 ± 0.016	0.030 ± 0.034	0.034 ± 0.037	0.017 ± 0.010

Table 6: Three-element support audit, with the same aggregation rule. Each support has 4,851 positive count vectors. Task MAE is relative to the true maximum, whereas the defect is relative to the model's once-per-element output.

F.6 Expanded results and interpretation

Mean and sum models can fit the maximum target to some extent while changing substantially along a repeated-registration fiber. Sum's large defect is consistent with its direct dependence on registered multiplicity. Mean removes common scaling of all counts but remains sensitive to their relative proportions.

The audit deliberately evaluates a registration grade beyond the training range. This is not a change in the semantic target: every length-100 input in these comparisons represents a support containing only two or three distinct elements, and its correct output is identical to that of the once-per-element registration. The stress therefore asks whether the learned implementation respects a support equivalence that is prescribed independently of registration length. It should be distinguished from ordinary extrapolation to sets containing 100 distinct elements or to an unknown target regime.

Set Transformer's small measured changes are a positive result for learned consistency on this task. Its empirical behavior should not be inflated into a structural identity: the count-weighted attention formula still permits multiplicity dependence, and a finite audit cannot rule it out elsewhere. Nor should the result be reversed into a claim that attention is unable to learn maximum functions. The meaningful distinction is between an operator-level guarantee and a learned approximation that performs well on the measured registrations.

For max pooling, average baseline-to-audit differences are approximately 4–5×10^{-6}, and within-grade diameters are zero or at a similar numerical scale. Algebraic idempotence applies to the mathematical pooling function. Floating-point kernels, batch shapes, and the evaluation route can induce small residuals in a whole program. Reporting this scale is more informative than declaring a literal nonzero value a semantic failure.

F.7 EVERY TRAINING RUN: TWO-ELEMENT SUPPORTS

Table 7: All 60 runs: two-element support audit. D_2 is the support-averaged mean defect and V_2 the support-averaged fiber diameter. Additional digits expose numerical residuals.

Model	Seed	Step	Clean MAE	D_2	V_2	Audit MAE
Mean	0	192000	0.7300	2.84625	31.71305	3.10482
Mean	1	185000	0.7806	2.69362	30.60075	2.83508
Mean	2	192000	0.8086	2.78537	31.59682	3.24725
Mean	3	191000	0.6806	2.74872	30.80985	2.91858
Mean	4	195000	0.7085	2.72719	31.19740	2.90416
Mean	5	181000	0.8762	2.73776	30.21671	2.89995
Mean	6	200000	0.7925	2.79443	30.51785	3.02398
Mean	7	188000	0.8524	2.79704	30.91221	3.02277
Mean	8	185000	0.7859	2.47971	29.32574	2.63521
Mean	9	196000	0.6731	2.45873	30.55000	2.57084
Sum	0	156000	0.2607	26.51312	99.39926	26.39652
Sum	1	174000	0.2530	34.12691	101.29907	34.06300
Sum	2	185000	0.3078	21.23193	61.48213	21.31857
Sum	3	194000	0.3727	34.09187	113.50085	34.10324
Sum	4	147000	0.2194	32.41637	80.70495	32.45594
Sum	5	191000	0.2522	22.84246	61.86299	22.87591
Sum	6	179000	0.2650	32.92674	83.02806	32.92184
Sum	7	188000	0.2482	39.69484	102.15788	39.71850
Sum	8	170000	0.2419	39.98118	83.08338	40.03927
Sum	9	199000	0.3571	51.34133	127.46914	51.32440
Max	0	186000	0.0031	0.00000	0.00000	0.00490
Max	1	90000	0.0046	0.00000	0.00000	0.00698
Max	2	137000	0.0052	0.00000	0.00000	0.00657
Max	3	67000	0.0036	0.00000	0.00000	0.00510
Max	4	116000	0.0071	0.00000	0.00000	0.01004
Max	5	165000	0.0041	0.00000	0.00000	0.00681
Max	6	125000	0.0042	0.00000	0.00000	0.00278
Max	7	80000	0.0077	0.00000	0.00000	0.00911
Max	8	196000	0.0026	0.00001	0.00000	0.00339
Max	9	127000	0.0032	0.00000	0.00000	0.00197
LME ($\tau = 0.1$)	0	83000	0.0346	0.06099	0.43154	0.07438
LME ($\tau = 0.1$)	1	154000	0.0230	0.04262	0.29788	0.05468
LME ($\tau = 0.1$)	2	193000	0.0423	0.07229	0.49647	0.08230
LME ($\tau = 0.1$)	3	99000	0.0305	0.04799	0.34736	0.06506
LME ($\tau = 0.1$)	4	194000	0.0312	0.06013	0.41544	0.07270
LME ($\tau = 0.1$)	5	190000	0.0414	0.08459	0.57849	0.09865
LME ($\tau = 0.1$)	6	130000	0.0501	0.09615	0.66337	0.12051
LME ($\tau = 0.1$)	7	150000	0.0306	0.04932	0.34267	0.06882
LME ($\tau = 0.1$)	8	178000	0.0265	0.04678	0.33169	0.05822
LME ($\tau = 0.1$)	9	190000	0.0302	0.04774	0.33343	0.06281
LME ($\tau = 1$)	0	193000	0.0708	0.14772	0.99732	0.17694
LME ($\tau = 1$)	1	141000	0.0756	0.15049	1.04219	0.18465
LME ($\tau = 1$)	2	128000	0.0869	0.17582	1.20930	0.21186
LME ($\tau = 1$)	3	199000	0.0838	0.15541	1.06059	0.18258
LME ($\tau = 1$)	4	195000	0.0800	0.15010	1.04765	0.18208
LME ($\tau = 1$)	5	195000	0.0868	0.16996	1.17678	0.21469
LME ($\tau = 1$)	6	195000	0.0856	0.15791	1.10421	0.20403
LME ($\tau = 1$)	7	197000	0.0903	0.18562	1.28124	0.21819
LME ($\tau = 1$)	8	199000	0.0693	0.14599	0.97825	0.18097
LME ($\tau = 1$)	9	168000	0.0747	0.15499	1.06361	0.18984
Set Transformer	0	149000	0.0139	0.00886	0.08555	0.01967
Set Transformer	1	187000	0.0164	0.00030	0.00347	0.01077
Set Transformer	2	58000	0.0177	0.00089	0.00723	0.02867
Set Transformer	3	195000	0.0143	0.00135	0.00894	0.00737
Set Transformer	4	117000	0.0034	0.00004	0.00072	0.00371
Set Transformer	5	137000	0.0064	0.00384	0.03672	0.00983
Set Transformer	6	134000	0.0294	0.00283	0.02490	0.03151
Set Transformer	7	169000	0.0176	0.00008	0.00135	0.00844
Set Transformer	8	167000	0.0081	0.00271	0.02346	0.00789
Set Transformer	9	177000	0.0298	0.00376	0.03233	0.04450

F.8 EVERY TRAINING RUN: THREE-ELEMENT SUPPORTS

Table 8: All 60 runs: three-element support audit, using the same selected checkpoints. D_3 and V_3 use the same definitions as in Table 7.

Model	Seed	Step	Clean MAE	D_3	V_3	Audit MAE
Mean	0	192000	0.7300	3.69104	46.53946	3.71224
Mean	1	185000	0.7806	3.55419	44.93650	3.51711
Mean	2	192000	0.8086	3.72893	46.45200	3.82764
Mean	3	191000	0.6806	3.53333	45.32870	3.59109
Mean	4	195000	0.7085	3.67862	46.62277	3.66015
Mean	5	181000	0.8762	3.36769	43.83819	3.41713
Mean	6	200000	0.7925	3.24696	43.81037	3.30961
Mean	7	188000	0.8524	3.43057	45.22048	3.54690
Mean	8	185000	0.7859	3.07390	42.16356	3.14899
Mean	9	196000	0.6731	3.16716	44.45846	3.18798
Sum	0	156000	0.2607	16.84450	130.52491	16.83726
Sum	1	174000	0.2530	28.10282	159.57474	28.08429
Sum	2	185000	0.3078	18.88753	118.88772	18.85019
Sum	3	194000	0.3727	25.39182	182.38505	25.40638
Sum	4	147000	0.2194	29.34625	109.71446	29.45063
Sum	5	191000	0.2522	19.69609	85.17870	19.71155
Sum	6	179000	0.2650	25.81824	139.15661	25.91837
Sum	7	188000	0.2482	36.49816	166.39714	36.50054
Sum	8	170000	0.2419	39.53659	146.07244	39.55693
Sum	9	199000	0.3571	37.03901	207.80451	37.11680
Max	0	186000	0.0031	0.00001	0.00000	0.00382
Max	1	90000	0.0046	0.00001	0.00000	0.00524
Max	2	137000	0.0052	0.00001	0.00000	0.00579
Max	3	67000	0.0036	0.00001	0.00000	0.00365
Max	4	116000	0.0071	0.00001	0.00000	0.00882
Max	5	165000	0.0041	0.00000	0.00000	0.00585
Max	6	125000	0.0042	0.00001	0.00000	0.00297
Max	7	80000	0.0077	0.00001	0.00000	0.00948
Max	8	196000	0.0026	0.00001	0.00000	0.00272
Max	9	127000	0.0032	0.00001	0.00000	0.00251
LME ($\tau = 0.1$)	0	83000	0.0346	0.07400	0.45046	0.08127
LME ($\tau = 0.1$)	1	154000	0.0230	0.05125	0.30953	0.05998
LME ($\tau = 0.1$)	2	193000	0.0423	0.08415	0.50091	0.08862
LME ($\tau = 0.1$)	3	99000	0.0305	0.06000	0.37320	0.06831
LME ($\tau = 0.1$)	4	194000	0.0312	0.07092	0.42375	0.08061
LME ($\tau = 0.1$)	5	190000	0.0414	0.09772	0.58359	0.10498
LME ($\tau = 0.1$)	6	130000	0.0501	0.11249	0.67295	0.12826
LME ($\tau = 0.1$)	7	150000	0.0306	0.05854	0.34982	0.06640
LME ($\tau = 0.1$)	8	178000	0.0265	0.05707	0.34832	0.06230
LME ($\tau = 0.1$)	9	190000	0.0302	0.05686	0.33995	0.06079
LME ($\tau = 1$)	0	193000	0.0708	0.16951	1.01570	0.18524
LME ($\tau = 1$)	1	141000	0.0756	0.17667	1.07062	0.19829
LME ($\tau = 1$)	2	128000	0.0869	0.20449	1.23437	0.22670
LME ($\tau = 1$)	3	199000	0.0838	0.17987	1.08192	0.19967
LME ($\tau = 1$)	4	195000	0.0800	0.17707	1.07558	0.19832
LME ($\tau = 1$)	5	195000	0.0868	0.19918	1.20791	0.23059
LME ($\tau = 1$)	6	195000	0.0856	0.18741	1.14442	0.21874
LME ($\tau = 1$)	7	197000	0.0903	0.21676	1.30986	0.23474
LME ($\tau = 1$)	8	199000	0.0693	0.16534	1.00596	0.18164
LME ($\tau = 1$)	9	168000	0.0747	0.17999	1.08346	0.20198
Set Transformer	0	149000	0.0139	0.01346	0.12577	0.02031
Set Transformer	1	187000	0.0164	0.00054	0.00523	0.01451
Set Transformer	2	58000	0.0177	0.00263	0.01903	0.02625
Set Transformer	3	195000	0.0143	0.00223	0.01578	0.01028
Set Transformer	4	117000	0.0034	0.00060	0.00632	0.00440
Set Transformer	5	137000	0.0064	0.00563	0.05244	0.01074
Set Transformer	6	134000	0.0294	0.00250	0.02373	0.02219
Set Transformer	7	169000	0.0176	0.00049	0.00544	0.01406
Set Transformer	8	167000	0.0081	0.00370	0.03204	0.00779
Set Transformer	9	177000	0.0298	0.00676	0.05432	0.03775

G EXPERIMENT 2: PROTOCOL AND ESTIMANDS

G.1 CHECKPOINT LINEAGES AND CONTROLLED SOURCE POPULATIONS

The ModelNet40 experiment (Wu et al., 2015) distinguishes two sources of variation that should not be conflated: the native data lineage of a released checkpoint and the source population used in the controlled audit.

The seven checkpoints originate from two ModelNet40 preprocessing lineages relevant here. PointNet, DGCNN, and PointMLP use the HDF5 release in their native evaluation contracts. PointNet++ MSG, PointConv, PointMamba, and PointGPT-S instead use the released `modelnet40_normal_resampled` files, which provide 10,000 XYZ/normal rows per object. These lineages differ in preprocessing history and are not assumed to be interchangeable realizations of one common geometric source.

Evaluating each checkpoint only on its native lineage would make checkpoint identity inseparable from source lineage. We therefore cross all seven checkpoints with two controlled source populations. The first is CAD-derived from the original OFF meshes; the second is derived from the released 10,000-row point/normal reservoirs. Within each source, every checkpoint receives the same stored registrations. Across sources, results are reported separately rather than treated as paired realizations of one certified common surface.

CAD-derived source. The released HDF5 representation contains only 2,048 points per object. This is sufficient for native benchmark evaluation, but it is too small to serve as the source population for the present resampling study. Under a without-replacement design, an $n = 2048$ registration would exhaust the released cloud and eliminate registration variability; sampling with replacement would instead introduce exact repetitions that are artificial as observations of a continuous surface. We therefore return to the original OFF geometry and use its positive-area triangle support as a continuous sampling source.

The source called `hdf5` in the result files samples the positive-area triangle support of an OFF mesh and applies a frozen OFF-to-HDF5 calibration. The transformed coordinate is

$$y = (Qx - c)/s, \qquad s > 0. \tag{45}$$

One object-specific map is shared by all checkpoints and conditions. The calibration places the CAD-derived surface in the coordinate convention of the HDF5 lineage; it does not assert equality between the continuous OFF surface and the released 2,048-point HDF5 cloud. The existing calibration table is used without modifying its entries or uniqueness labels. Isolated vertices and zero-area edges are not sampled by a surface-area law; outlying positive-area faces are not silently removed.

Pool-derived source. The source called `normal_resampled` uses the released 10,000 XYZ/normal rows for each object. Its centroid and maximum radius are computed once from the entire pool, and that transform is reused for every registration and checkpoint. Subsampling uses distinct row indices without replacement. Distinct indices need not imply distinct coordinates if the released pool already contains coincident observations. This protocol inserts no new duplicate rows and does not assume that the reservoir is an exact finite sample of the calibrated OFF surface.

The two controlled sources therefore differ in geometry, orientation/normalization lineage, normal attributes, and sampling law. CAD normals are face normals, whereas pool normals are released attributes. PointNet++ MSG and PointConv use the aligned normals supplied with the common input, while XYZ-only models receive no added feature channel. Cross-source differences can reflect any combination of these source differences and are therefore interpreted descriptively, not as a controlled causal contrast of density or architecture alone. Within-source checkpoint comparisons, by contrast, use identical stored registrations and are directly paired.

G.2 IDENTITY AND CALIBRATION VALIDATION

The test split contains 2,468 named objects. Input validation checks agreement of object identities and labels across the calibration, CAD, HDF5, and normal-resampled manifests, and fully reads the reservoir files. The recorded validation reports no missing or extra test identities and no label mismatches.

Of the 2,468 test objects, the OFF-to-HDF5 calibration is uniquely identified for 2,462 and non-uniquely identified for six. In the latter cases, the available geometric evidence admits more than one compatible transform. One representative transform is retained for evaluation, while the non-unique calibration status is preserved in the analysis.

The non-uniquely identified calibrations were manually inspected when establishing the OFF-to-HDF5 fit. There are 36 such cases in the full calibration table and six in the test split. As a sensitivity check, we recompute the aggregate summaries after excluding the six non-uniquely identified test calibrations. Across the summary grid, this exclusion changes accuracy by at most about 0.157 percentage points; the largest changes in mean relative-logit repeatability, density response, and refinement are approximately 0.000794, 0.001228, and 0.000592, respectively. The reported conclusions are therefore not driven by the six calibration-ambiguous objects.

G.3 Checkpoint contracts and native preflight

All seven checkpoints use native PyTorch inference. The verified environment reports Python 3.12.3, PyTorch 2.10.0 with CUDA 12.8, `torch-geometric` 2.8.0.post1, `torch-pointcloud` 0.0.1, and `mamba-ssm` 2.3.2.post1 on a single consumer NVIDIA GPU. The geometric extensions and a CUDA Mamba forward pass succeed.

Each checkpoint is first evaluated with its registered source evaluation transform at its native cardinality of 1024. This is distinct from the controlled input-bank evaluation. PointNet uses the HDF5 release's prefix contract; the other checkpoints use their registered transforms, including FPS, normalization, and normal features where prescribed. Native full-split preflight tests all 2,468 objects and passes the configured minimum accuracy gates. A separate fixed-hierarchy reference-logit gate tests eight examples at 1024 points with absolute and relative tolerances 10^{-5}; the largest reported absolute discrepancy among these gates is approximately 1.9×10^{-6}.

Checkpoint	Native lineage	OA (%)	Min. OA (%)
PointNet	HDF5	90.68	85
DGCNN	HDF5	92.91	85
PointMLP	HDF5	93.35	85
PointNet++ MSG	Normal pool	92.67	85
PointConv	Normal pool	92.30	85
PointMamba	Normal pool	93.64	80
PointGPT-S (1k)	Normal pool	93.31	80

Table 9: Native checkpoint preflight. Native-lineage accuracy is a validity check for each checkpoint, not a score under either controlled source population.

Checkpoint source identifiers are `pointnet2-msg.modelnet40.xu-yan`, `dgcnn.modelnet40-1024.an-tao`, `pointmlp-base.modelnet40.xu-ma`, `pointconv-density-base.modelnet40.wenxuan-wu`, `point-mamba-base.modelnet40.dingkang-liang`, and `pointgpt-s.modelnet40.guangyan-chen` in the registered `torch-pointcloud` collection. PointNet uses the benchmark-author checkpoint specified by the source manifest. These are individual released checkpoints, some converted into the registry's format, not a claim of evaluating every official training seed or the strongest result of each architecture.

G.4 Acquisition-density field

The primary field uses a $4 \times 4 \times 4$ grid of independent standard normal values in the object's axis-aligned bounding cube. The grid is smoothed with a Gaussian filter of standard deviation 0.75 grid units and reflective boundary handling. For CAD, the field is evaluated at each triangle centroid and standardized using normalized triangle-area weights. It is constant on each triangle, hence tessellation-dependent. For the finite pool, it is evaluated at each released row and standardized under the uniform empirical measure.

Let a_j denote normalized triangle area, or $1/M$ for a pool with $M = 10,000$ rows, and let h_j be the standardized field. For CAD, the weights a_j arise from geometric surface area: they discretize the normalized surface-area law and do not represent material mass density. The selection weights at

strength λ are

$$p_j^{(\lambda)} = (1-\eta)\frac{a_j e^{\lambda h_j}}{\sum_\ell a_\ell e^{\lambda h_\ell}} + \eta a_j, \qquad \eta = 0.02. \tag{46}$$

At $\lambda = 0$ this reduces to the base acquisition law. Changing λ redistributes observations over the same designated source; it does not by itself alter the intended CAD support or its geometric surface-area measure. The uniform mixture keeps all positive-base-mass regions eligible, but does not guarantee that a particular finite draw covers every region.

On CAD, triangles are sampled with replacement from these weights, then points are drawn uniformly inside each chosen triangle using barycentric sampling. On the reservoir, a weighted order without replacement is generated by independent exponential-race keys $-\log U_j / p_j^{(\lambda)}$ and sorted ascending. Its first n rows form the registration. Thus the pool law is a finite-population design rather than an i.i.d. point process, even at $\lambda = 0$.

There is one object-specific field in the completed primary suite. Registration seeds are 0, 1, and 2, while the field-seed index is 0. The base seed is 20260905; object identifiers determine stable object-specific codes. The effective field seed combines the base, object code, and field index; the observed registration seed additionally uses the registration index. The repeat seed adds 700000001 modulo $2^{32} - 1$. A model name or execution order never enters the registration recipe.

G.5 Four protocols crossed with two sources

Protocol	Cardinalities	Density strengths	Controlled comparison
Natural	1024	0, 1, 2, 4	Sampling and density
Coverage-controlled	1024	0, 1, 2, 4	Density with anchors
Nested IID	512, 1024, 2048	0	Refinement stress
Nested coverage	512, 1024, 2048	0	Anchored refinement stress

Table 10: Every protocol runs on both controlled sources and all seven checkpoints. The inherited "nested IID" name is literal for CAD surface draws; on the finite reservoir its implementation is nested sampling without replacement.

Natural sampling uses the entire 1024-point budget. Coverage-controlled sampling first constructs 256 anchors by FPS. On CAD these anchors are selected from an independently generated uniform pool of 2,048 surface points; on the reservoir they are selected from all released rows. At a fixed object/registration seed, the same anchors are used for every density and for the observed and repeat draws. The remaining 768 observations follow the density law. Reservoir anchors are excluded from the remaining eligible row indices. The coverage design is not i.i.d.; anchor density ratios are intentionally undefined for importance weighting.

The nested protocols generate a maximum-size draw and take exact ordered prefixes. Nested coverage retains the 256 anchors at every size and extends only the remaining observations. The prefix relation and corresponding normals are verified in the stored input bank. The grids are part of the bank recipe: separately generating a different maximum size is not assumed to reproduce the same smaller draw merely because the integer seed agrees.

For each density and size, the bank stores observed and repeat arrays. Natural pairs use separate random draws. Coverage pairs are independent only conditional on their shared scaffold. Different registration seeds generate different scaffolds. Across densities, the observed registration uses the same seed and field, creating a controlled coupling; it is not an independent resampling contrast. The $\lambda = 0$ density-change metric is therefore identically zero by construction.

G.6 Fixed hierarchy and input-bank integrity

Every model receives the same ordered float32 XYZ array for a given bank entry, and the same aligned normals when its contract uses them. There is no external model-specific FPS, per-condition recentering, renormalization, reordering, or duplicate insertion after materialization. Internal operations of the classifier still execute: identical external input does not imply identical internal neighborhoods.

The reference hierarchy is held fixed rather than scaled proportionally with n. PointMLP uses centroid counts $(512, 256, 128, 64)$; PointNet++ MSG uses $(512, 128)$; PointConv uses $(512, 128, 1)$. PointMamba and PointGPT-S use 64 groups of 32 neighbors. DGCNN retains its fixed $k = 20$ neighborhood rule, and PointNet has no sampled hierarchy. This policy isolates external cardinality changes from an additional change in internal hierarchy sizes. It is an explicit stress protocol, not a claim that this is the only valid cross-cardinality extension of a checkpoint.

In particular, a 512- or 2048-point PointMamba/PointGPT input is fed to the native network at that actual input size; it is not padded to 1024 or externally subsampled back to 1024. Internal grouping selects 64 FPS centers from the available n points and collects 32 neighbors per center. The local coordinates and groups therefore change with n, while the number of groups and neighborhood size remain fixed. Groups can overlap; 64×32 is not a count of distinct required input points. Subsequent model-specific serialization and token processing operate on these groups. The weights and feature widths are unchanged, and 1024 remains the checkpoint's native benchmark cardinality. These conditions support the tested $\{512, 1024, 2048\}$ grid, not an automatically validated extension to arbitrary larger sizes.

The stored input bank records the source and calibration, frozen transforms, sampling seeds, density settings, coordinates, aligned normals, and source-row or triangle indices. Entries remain unchanged during inference. Validation checks that checkpoints receive identical registrations and that each nested child retains its recorded parent as an ordered prefix. The completed suite passes these shared-input checks.

For inference, the model RNG is reset to zero on each forward call. TF32 is disabled for matrix multiplication and cuDNN, cuDNN benchmarking is disabled, and deterministic cuDNN behavior is requested. This is not a guarantee that all compiled GPU kernels are bitwise deterministic. Repeating the same uniform observed input for each object/seed/size measures numerical discrepancy, with tolerance $\mathrm{atol} = \mathrm{rtol} = 10^{-5}$. Actual raw-logit maximum absolute differences and centered-logit RMS values are recorded.

G.7 Definitions of reliability and geometric measurements

A *repeat registration* is a newly materialized point cloud of the same object under the same source, density strength, and cardinality. It changes acquisition randomness, not the designated source. For natural sampling, A and B use separate draws; for coverage-controlled sampling, their extra observations are redrawn conditional on shared anchors. In nested protocols, each draw is a complete sequence whose cardinalities are prefixes. This is measurement repeatability conditional on the design (Bartlett & Frost, 2008). A *repeat forward call*, by contrast, runs the model again on exactly the same stored array to measure numerical variation. The two controls should not be conflated.

Let $A_{\lambda,n}$ and $B_{\lambda,n}$ denote the observed and repeat registrations, and write $z(A)$ for the model logits. The three primary pairings are

$$\text{repeatability:} \quad z(A_{\lambda,n}) \text{ versus } z(B_{\lambda,n}), \tag{47}$$

$$\text{density response:} \quad z(A_{0,n}) \text{ versus } z(A_{\lambda,n}), \tag{48}$$

$$\text{refinement:} \quad z(A_{0,n/2}) \text{ versus } z(A_{0,n}), \quad n = 1024, 2048. \tag{49}$$

The centered-logit RMS is $\|g(z') - g(z)\|_2/\sqrt{40}$; its symmetric relative version is equation 4. The latter is bounded above by two and is invariant to common nonzero scaling away from its denominator floor. It does not remove changes of class-dependent scale or make independently trained logit geometries identical. Accuracy, class flips, and Jensen–Shannon divergence of softmax probabilities remain useful secondary measurements. Raw unnormalized logit changes should not be interpreted as directly comparable effect sizes across checkpoints.

The finite-cloud Hausdorff distance is computed exactly from nearest-neighbor distances between the two observed arrays. Sliced W_2 (Bonneel et al., 2015) uses 32 fixed random unit directions, exact one-dimensional empirical transport in each projection, including for unequal cardinalities, and the square root of the mean squared projected distance. This is a finite-direction approximation to sliced Wasserstein distance, not the full three-dimensional W_2.

These quantities diagnose different components of the registration change. Hausdorff distance measures finite-support discrepancy and is the direct geometric diagnostic for support-level near-

idempotence. Sliced W_2 measures change in the empirical distribution of sampled observations. For the CAD-derived source, a change in this empirical distribution does not by itself change the intended continuum support or its geometrically induced surface-area measure. Reporting both quantities therefore distinguishes loss of geometric coverage from redistribution of observations over the same designated source.

Coverage error is the maximum reference-to-registration nearest-neighbor distance. CAD uses a separate deterministic sample of 16,384 uniform surface points as reference. Its finite-reference maximum approximates, and can underestimate, continuum coverage error. For the pool-derived source, the reference is the entire released 10,000-row reservoir, so coverage is relative to that finite population. Points lie on their designated source by construction; that statement does not imply that pool points lie on the CAD surface.

G.8 Aggregation and confidence intervals

For a fixed checkpoint and condition, first average the three registration-seed measurements for each object. The reported object mean weights each of the 2,468 objects equally. To compute uncertainty, sample objects with replacement independently within each of the 40 classes, preserving the observed number of objects per class, and average the resampled object means. Percentiles 2.5 and 97.5 of 1,000 bootstrap replicates give the reported interval. The summary seed is 20260920.

For a model contrast, form a within-object difference after matching identical stored registrations and averaging corresponding registration seeds. Bootstrap these paired differences with the same class-stratified object scheme. This preserves dependence between models evaluated on the same source object. Individual registrations, result rows, or points are not independent bootstrap units. The intervals quantify sampling variation over the evaluated object population conditional on the fixed checkpoints, field design, and registration protocol; they do not include uncertainty from retraining models or selecting additional field families.

The intervals are pointwise. The large collection of conditions and model contrasts is not a familywise-significance procedure. We emphasize descriptive patterns and selected paired contrasts, and avoid declaring a global architecture ranking. The two controlled source populations are summarized separately and are not pooled in a single bootstrap estimate.

G.9 Completion and numerical checks

The summary metadata records all eight protocol/source combinations, all seven checkpoints, no missing shards, no unverified checkpoints, and exclusion of smoke runs. There are $588 \times 2,468 = 1,451,184$ result rows. The reporting inputs are checked for consistency with this completed design.

Across model/condition means computed over the full test split of 2,468 objects, the largest mean repeated-call raw-logit discrepancy is approximately 1.39×10^{-6} and the largest mean centered RMS discrepancy is about 4.45×10^{-7}. These are maxima across full-split aggregate means, not maxima over individual forward calls or calibration subgroups. Several checkpoint summaries report zeros at the stored precision. The resampling and density changes are far larger on the reported scale.

H Experiment 2: expanded results

H.1 Geometry and empirical distribution of the interventions

Table 11 reports geometric measurements shared across all seven checkpoints. The analysis script verifies that these entries agree across models before retaining one copy. Coverage-controlled sampling is designed to limit changes in support geometry while permitting substantial changes in the empirical observation distribution. It does not make the finite registrations identical. Hausdorff distance and coverage error therefore describe the support-level component of the intervention, whereas sliced W_2 describes its empirical-distribution component.

On the CAD-derived source, natural sampling at $\lambda = 4$ produces mean density-pair Hausdorff distance 0.4308 and coverage error 0.4487, compared with coverage error 0.1239 at $\lambda = 0$. Coverage control instead gives density-pair Hausdorff distance 0.1063 and coverage error 0.1192, compared

Source	Protocol	λ	Coverage	Rep. d_H	Rep. SW_2	Den. d_H	Den. SW_2
CAD	Natural	0	0.1239	0.1161	0.0217	0.0000	0.0000
CAD	Natural	1	0.1958	0.1700	0.0224	0.1784	0.1285
CAD	Natural	2	0.3167	0.2583	0.0239	0.2989	0.2421
CAD	Natural	4	0.4487	0.3662	0.0257	0.4308	0.3451
CAD	Coverage	0	0.0998	0.0908	0.0189	0.0000	0.0000
CAD	Coverage	1	0.1090	0.0939	0.0188	0.0969	0.0969
CAD	Coverage	2	0.1155	0.0937	0.0177	0.1029	0.1825
CAD	Coverage	4	0.1192	0.0910	0.0151	0.1063	0.2592
Pool	Natural	0	0.1135	0.1073	0.0209	0.0000	0.0000
Pool	Natural	1	0.1873	0.1644	0.0205	0.1719	0.1168
Pool	Natural	2	0.2725	0.2292	0.0197	0.2585	0.1873
Pool	Natural	4	0.2591	0.2292	0.0187	0.2437	0.2130
Pool	Coverage	0	0.0888	0.0838	0.0184	0.0000	0.0000
Pool	Coverage	1	0.0939	0.0869	0.0176	0.0875	0.0908
Pool	Coverage	2	0.0951	0.0874	0.0157	0.0913	0.1501
Pool	Coverage	4	0.0950	0.0877	0.0138	0.0913	0.1779

Table 11: Geometry by controlled source, protocol, and density strength. Coverage is reference-to-registration distance; Rep. compares observed/repeat inputs, and Den. compares λ to zero. SW_2 uses 32 projections. Coordinates use each source's frozen transform, so cross-source distances are descriptive rather than a common calibrated severity scale.

with 0.0998 at zero. Thus the coverage-controlled design substantially reduces, but does not eliminate, finite-support change.

On the pool-derived source, the corresponding natural density-pair Hausdorff distance is 0.2437, whereas coverage control gives 0.0913. The acquisition laws and geometric discrepancies differ between the two source populations even at a common λ. Cross-source changes in model response are therefore interpreted descriptively rather than attributed to source representation alone.

H.2 Reading the full fixed-cardinality tables

Tables 12–15 report every fixed-cardinality condition. Accuracy scores the observed registration. Density flips and density R compare it with the coupled $\lambda = 0$ registration; repeatability compares it with a new registration at the same λ. Hence density response is zero at $\lambda = 0$ by construction, while repeatability can remain nonzero. Accuracy and representation stability answer different questions, particularly near class boundaries.

Model	λ	Acc. (%)	Repeatability R [95% CI]	Density R [95% CI]	Flip (%)
PointNet	0	89.65	0.240 [0.237, 0.244]	0.000 [0.000, 0.000]	0.00
PointNet	1	87.30	0.337 [0.333, 0.341]	0.356 [0.351, 0.360]	6.97
PointNet	2	74.31	0.545 [0.540, 0.550]	0.684 [0.677, 0.692]	22.81
PointNet	4	42.53	0.730 [0.724, 0.736]	1.021 [1.014, 1.029]	55.85
DGCNN	0	88.99	0.534 [0.530, 0.537]	0.000 [0.000, 0.000]	0.00
DGCNN	1	59.62	0.589 [0.585, 0.592]	0.805 [0.800, 0.809]	38.61
DGCNN	2	16.06	0.561 [0.558, 0.565]	1.093 [1.087, 1.099]	82.64
DGCNN	4	6.05	0.554 [0.550, 0.557]	1.250 [1.245, 1.255]	93.17
PointMLP	0	89.94	0.242 [0.238, 0.246]	0.000 [0.000, 0.000]	0.00
PointMLP	1	68.37	0.372 [0.367, 0.376]	0.567 [0.559, 0.576]	28.04
PointMLP	2	16.95	0.352 [0.348, 0.356]	1.057 [1.048, 1.065]	80.25
PointMLP	4	7.13	0.353 [0.349, 0.357]	1.242 [1.235, 1.249]	90.53
PointNet++	0	88.32	0.182 [0.180, 0.185]	0.000 [0.000, 0.000]	0.00
PointNet++	1	87.49	0.214 [0.211, 0.216]	0.235 [0.233, 0.238]	5.97
PointNet++	2	79.43	0.336 [0.331, 0.340]	0.442 [0.435, 0.448]	16.32
PointNet++	4	49.10	0.532 [0.525, 0.538]	0.805 [0.796, 0.814]	48.87
PointConv	0	86.05	0.116 [0.114, 0.118]	0.000 [0.000, 0.000]	0.00
PointConv	1	82.71	0.144 [0.142, 0.147]	0.243 [0.238, 0.248]	9.90
PointConv	2	63.98	0.234 [0.230, 0.238]	0.570 [0.560, 0.580]	31.97
PointConv	4	29.19	0.297 [0.292, 0.302]	0.963 [0.952, 0.974]	69.39
PointMamba	0	91.27	0.129 [0.125, 0.133]	0.000 [0.000, 0.000]	0.00
PointMamba	1	81.78	0.226 [0.220, 0.231]	0.362 [0.352, 0.370]	15.30
PointMamba	2	38.32	0.333 [0.327, 0.338]	0.914 [0.902, 0.927]	60.14
PointMamba	4	14.86	0.375 [0.370, 0.381]	1.213 [1.204, 1.222]	84.67
PointGPT-S	0	89.57	0.167 [0.163, 0.171]	0.000 [0.000, 0.000]	0.00
PointGPT-S	1	76.31	0.299 [0.293, 0.306]	0.400 [0.391, 0.409]	21.65
PointGPT-S	2	35.52	0.421 [0.415, 0.426]	0.925 [0.913, 0.935]	63.79
PointGPT-S	4	12.80	0.423 [0.417, 0.428]	1.214 [1.206, 1.221]	86.86

Table 12: CAD-derived source (HDF5 frame), natural sampling, $n = 1024$. All 2,468 objects and three registration seeds are included. Brackets contain pointwise 95% class-stratified object-cluster bootstrap intervals. Flip is the density-pair class-flip rate in percent.

Model	λ	Acc. (%)	Repeatability R [95% CI]	Density R [95% CI]	Flip (%)
PointNet	0	89.75	0.159 [0.157, 0.161]	0.000 [0.000, 0.000]	0.00
PointNet	1	89.45	0.154 [0.152, 0.157]	0.175 [0.172, 0.177]	3.23
PointNet	2	88.88	0.134 [0.132, 0.136]	0.215 [0.212, 0.218]	3.70
PointNet	4	88.34	0.098 [0.097, 0.100]	0.254 [0.251, 0.257]	4.62
DGCNN	0	90.78	0.478 [0.474, 0.481]	0.000 [0.000, 0.000]	0.00
DGCNN	1	83.95	0.507 [0.504, 0.510]	0.652 [0.647, 0.656]	14.28
DGCNN	2	58.13	0.488 [0.485, 0.492]	0.877 [0.871, 0.883]	40.60
DGCNN	4	35.98	0.422 [0.418, 0.425]	1.035 [1.029, 1.040]	63.24
PointMLP	0	92.22	0.184 [0.181, 0.188]	0.000 [0.000, 0.000]	0.00
PointMLP	1	89.56	0.215 [0.211, 0.219]	0.276 [0.270, 0.281]	7.16
PointMLP	2	82.59	0.239 [0.235, 0.242]	0.433 [0.425, 0.441]	14.10
PointMLP	4	75.03	0.215 [0.212, 0.218]	0.547 [0.538, 0.554]	22.69
PointNet++	0	88.70	0.148 [0.146, 0.149]	0.000 [0.000, 0.000]	0.00
PointNet++	1	88.60	0.144 [0.142, 0.145]	0.160 [0.158, 0.162]	3.93
PointNet++	2	88.38	0.129 [0.127, 0.130]	0.179 [0.177, 0.181]	4.73
PointNet++	4	88.64	0.102 [0.101, 0.104]	0.194 [0.192, 0.197]	5.05
PointConv	0	86.57	0.100 [0.098, 0.102]	0.000 [0.000, 0.000]	0.00
PointConv	1	85.98	0.101 [0.099, 0.102]	0.145 [0.142, 0.148]	5.79
PointConv	2	84.18	0.097 [0.095, 0.098]	0.222 [0.217, 0.226]	8.77
PointConv	4	81.85	0.081 [0.080, 0.083]	0.285 [0.279, 0.290]	12.10
PointMamba	0	92.10	0.101 [0.097, 0.104]	0.000 [0.000, 0.000]	0.00
PointMamba	1	89.94	0.114 [0.111, 0.117]	0.174 [0.169, 0.179]	6.66
PointMamba	2	85.41	0.131 [0.128, 0.134]	0.317 [0.308, 0.325]	12.03
PointMamba	4	78.78	0.130 [0.127, 0.133]	0.453 [0.443, 0.464]	19.30
PointGPT-S	0	91.29	0.135 [0.131, 0.138]	0.000 [0.000, 0.000]	0.00
PointGPT-S	1	88.13	0.172 [0.168, 0.176]	0.214 [0.209, 0.220]	9.13
PointGPT-S	2	82.12	0.221 [0.217, 0.226]	0.347 [0.339, 0.355]	16.38
PointGPT-S	4	75.85	0.235 [0.230, 0.240]	0.452 [0.443, 0.460]	22.81

Table 13: CAD-derived source (HDF5 frame), coverage-controlled sampling, $n = 1024$. Repeatability is conditional on 256 shared anchors. Density-pair comparisons retain those anchors across strengths. All other definitions match Table 12.

Model	λ	Acc. (%)	Repeatability R [95% CI]	Density R [95% CI]	Flip (%)
PointNet	0	89.47	0.198 [0.195, 0.200]	0.000 [0.000, 0.000]	0.00
PointNet	1	88.45	0.301 [0.297, 0.305]	0.275 [0.272, 0.279]	4.90
PointNet	2	81.48	0.464 [0.459, 0.468]	0.544 [0.538, 0.550]	14.45
PointNet	4	76.92	0.502 [0.495, 0.508]	0.580 [0.569, 0.590]	19.10
DGCNN	0	88.76	0.517 [0.514, 0.520]	0.000 [0.000, 0.000]	0.00
DGCNN	1	62.07	0.571 [0.568, 0.574]	0.771 [0.767, 0.776]	35.35
DGCNN	2	24.95	0.563 [0.560, 0.565]	1.019 [1.014, 1.024]	74.32
DGCNN	4	22.72	0.552 [0.549, 0.556]	1.059 [1.051, 1.066]	76.39
PointMLP	0	90.95	0.221 [0.217, 0.225]	0.000 [0.000, 0.000]	0.00
PointMLP	1	76.32	0.345 [0.340, 0.349]	0.489 [0.482, 0.496]	20.35
PointMLP	2	27.58	0.366 [0.362, 0.370]	0.963 [0.955, 0.971]	70.58
PointMLP	4	26.78	0.332 [0.328, 0.337]	0.982 [0.969, 0.995]	71.31
PointNet++	0	92.30	0.163 [0.161, 0.165]	0.000 [0.000, 0.000]	0.00
PointNet++	1	91.64	0.187 [0.185, 0.189]	0.169 [0.167, 0.172]	3.12
PointNet++	2	90.28	0.250 [0.247, 0.253]	0.300 [0.296, 0.304]	6.23
PointNet++	4	87.06	0.299 [0.294, 0.303]	0.383 [0.377, 0.390]	9.51
PointConv	0	91.76	0.095 [0.093, 0.096]	0.000 [0.000, 0.000]	0.00
PointConv	1	90.45	0.116 [0.114, 0.118]	0.172 [0.169, 0.175]	3.93
PointConv	2	85.29	0.161 [0.159, 0.164]	0.372 [0.365, 0.379]	10.94
PointConv	4	79.19	0.185 [0.182, 0.189]	0.435 [0.425, 0.444]	17.26
PointMamba	0	92.26	0.110 [0.106, 0.113]	0.000 [0.000, 0.000]	0.00
PointMamba	1	86.24	0.192 [0.187, 0.196]	0.291 [0.284, 0.298]	10.09
PointMamba	2	56.90	0.310 [0.305, 0.315]	0.729 [0.717, 0.740]	40.82
PointMamba	4	46.88	0.302 [0.298, 0.308]	0.822 [0.806, 0.836]	51.01
PointGPT-S	0	92.42	0.140 [0.136, 0.143]	0.000 [0.000, 0.000]	0.00
PointGPT-S	1	84.32	0.247 [0.241, 0.251]	0.315 [0.307, 0.322]	13.24
PointGPT-S	2	51.54	0.407 [0.401, 0.412]	0.754 [0.743, 0.764]	47.20
PointGPT-S	4	42.21	0.401 [0.395, 0.408]	0.859 [0.844, 0.872]	56.86

Table 14: Released 10,000-row pool-derived source, natural sampling without replacement, $n = 1024$. Results are summarized separately from the CAD-derived source. Interval and metric definitions match Table 12.

Model	λ	Acc. (%)	Repeatability R [95% CI]	Density R [95% CI]	Flip (%)
PointNet	0	90.14	0.127 [0.125, 0.129]	0.000 [0.000, 0.000]	0.00
PointNet	1	89.75	0.126 [0.124, 0.127]	0.107 [0.106, 0.109]	1.85
PointNet	2	89.52	0.116 [0.114, 0.118]	0.157 [0.155, 0.159]	2.54
PointNet	4	89.40	0.109 [0.108, 0.111]	0.177 [0.174, 0.180]	3.15
DGCNN	0	90.86	0.463 [0.460, 0.466]	0.000 [0.000, 0.000]	0.00
DGCNN	1	81.59	0.491 [0.488, 0.494]	0.626 [0.622, 0.630]	15.45
DGCNN	2	64.36	0.488 [0.486, 0.491]	0.821 [0.816, 0.826]	33.60
DGCNN	4	55.69	0.472 [0.469, 0.475]	0.888 [0.882, 0.894]	42.86
PointMLP	0	92.71	0.155 [0.152, 0.158]	0.000 [0.000, 0.000]	0.00
PointMLP	1	90.60	0.173 [0.170, 0.176]	0.219 [0.215, 0.223]	4.98
PointMLP	2	86.25	0.195 [0.192, 0.198]	0.353 [0.347, 0.360]	10.28
PointMLP	4	82.43	0.200 [0.197, 0.202]	0.431 [0.422, 0.440]	14.49
PointNet++	0	92.65	0.124 [0.123, 0.126]	0.000 [0.000, 0.000]	0.00
PointNet++	1	92.59	0.122 [0.121, 0.124]	0.099 [0.097, 0.100]	1.89
PointNet++	2	92.30	0.113 [0.111, 0.114]	0.132 [0.130, 0.134]	2.40
PointNet++	4	92.18	0.108 [0.107, 0.109]	0.142 [0.139, 0.144]	2.59
PointConv	0	92.17	0.067 [0.066, 0.069]	0.000 [0.000, 0.000]	0.00
PointConv	1	91.60	0.068 [0.067, 0.070]	0.093 [0.091, 0.095]	2.47
PointConv	2	90.98	0.066 [0.065, 0.068]	0.156 [0.153, 0.159]	4.00
PointConv	4	90.14	0.066 [0.064, 0.067]	0.187 [0.182, 0.191]	5.15
PointMamba	0	92.95	0.064 [0.062, 0.066]	0.000 [0.000, 0.000]	0.00
PointMamba	1	91.57	0.070 [0.068, 0.072]	0.128 [0.124, 0.132]	4.21
PointMamba	2	88.88	0.077 [0.075, 0.079]	0.233 [0.226, 0.239]	7.56
PointMamba	4	86.70	0.083 [0.080, 0.085]	0.291 [0.283, 0.299]	9.66
PointGPT-S	0	93.08	0.071 [0.069, 0.073]	0.000 [0.000, 0.000]	0.00
PointGPT-S	1	90.79	0.079 [0.077, 0.082]	0.145 [0.140, 0.149]	5.33
PointGPT-S	2	87.67	0.091 [0.089, 0.093]	0.250 [0.243, 0.257]	9.45
PointGPT-S	4	85.35	0.094 [0.092, 0.097]	0.306 [0.298, 0.314]	12.17

Table 15: Released 10,000-row pool-derived source, coverage-controlled sampling, $n = 1024$. Extra rows exclude the anchor indices. Repeatability is conditional on the common scaffold, and source-specific normalization is frozen using the full pool.

H.3 NESTED CARDINALITY STRESS

Tables 16–19 report the two nested designs. Refinement at $n = 1024$ compares with the 512-point prefix, and refinement at $n = 2048$ compares with the 1024-point prefix. There is no smaller parent in the grid for $n = 512$, so its refinement entry is absent rather than zero. All density strengths here are zero. Fixed hierarchy counts are part of the condition, not a property guaranteed by every model's unmodified inference entry point.

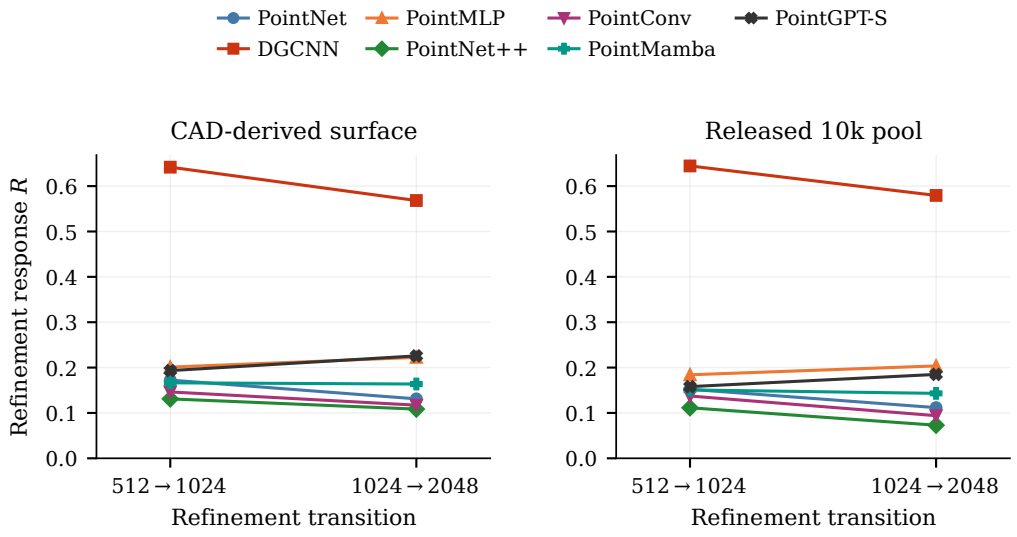


Figure 7: Nested coverage refinement at the two tested transitions. PointMLP and PointGPT-S show larger mean relative-logit changes at the second transition in both controlled sources; several other models show reductions. Two transitions do not establish convergence or divergence as cardinality tends to infinity.

Under coverage-controlled $1024 \to 2048$ refinement, PointNet, PointNet++, and PointConv have smaller mean R than DGCNN, PointMLP, PointMamba, and PointGPT-S on both sources, with every corresponding pointwise paired interval excluding zero. PointNet minus PointMamba is -0.0327 $[-0.0374, -0.0276]$ on the CAD-derived source and -0.0314 $[-0.0356, -0.0270]$ on the pool-derived source. PointNet++ and PointConv retain the mean ordering across both nested protocols and transitions; PointNet does not. The full tables give each transition's effect size, including the increases for PointMLP and PointGPT-S under nested coverage.

The nonmonotonic patterns do not imply that every accuracy metric deteriorates with refinement. A representation can change while a predicted class stays fixed, and a change can correct a previously incorrect class. Conversely, nested set inclusion does not require the learned representation itself to be monotone. The framework asks whether finite approximations to the same designated source become stable in the chosen geometry, a stronger question than whether a specific two-step numerical trend happens to decrease.

H.4 SELECTED PAIRED CONTRASTS AND LIMITATIONS

The paired summary compares models using the same object and exact stored registrations. Under uniform natural sampling, DGCNN minus PointNet has mean relative-logit repeatability difference 0.294 $[0.290, 0.297]$ on the CAD-derived source and 0.319 $[0.316, 0.322]$ on the pool-derived source. At coverage-controlled $\lambda = 4$, the corresponding density-response differences are 0.781 $[0.775, 0.786]$ and 0.711 $[0.705, 0.716]$. These intervals concern paired differences, not subtraction of marginal interval endpoints.

Not every apparent difference is resolved on this scale. For example, PointGPT-S minus PointMamba under CAD-derived coverage control at $\lambda = 4$ has density-response difference about -0.002 with

interval $[-0.011, 0.007]$. A table ordering alone would overstate the evidence for that particular contrast. The full paired CSV is included in the source package so that additional comparisons can be specified explicitly rather than inferred from plot positions.

The audit concerns released checkpoints and does not identify which architectural component causes a response. With one density-field family and one field per object, its intervals do not cover retraining or alternative field designs. The primary suite tests geometric resampling rather than exact duplicate insertion, so its metrics are not exact support-fiber diameters. They instead quantify finite near-idempotence and empirical-distribution sensitivity under the specified interventions. These observations establish neither global Lipschitz bounds nor asymptotic convergence.

Model	n	Acc. (%)	Repeatability R [95% CI]	Refinement R [95% CI]
PointNet	512	86.78	0.372 [0.367, 0.377]	–
PointNet	1024	89.26	0.242 [0.239, 0.245]	0.279 [0.275, 0.283]
PointNet	2048	90.17	0.153 [0.150, 0.155]	0.176 [0.173, 0.178]
DGCNN	512	76.39	0.624 [0.620, 0.627]	–
DGCNN	1024	88.72	0.534 [0.531, 0.537]	0.692 [0.688, 0.696]
DGCNN	2048	87.87	0.469 [0.466, 0.472]	0.586 [0.583, 0.590]
PointMLP	512	88.64	0.286 [0.282, 0.291]	–
PointMLP	1024	89.90	0.243 [0.239, 0.247]	0.269 [0.264, 0.274]
PointMLP	2048	89.86	0.244 [0.240, 0.248]	0.246 [0.242, 0.250]
PointNet++	512	87.82	0.218 [0.215, 0.220]	–
PointNet++	1024	88.34	0.182 [0.180, 0.185]	0.150 [0.149, 0.152]
PointNet++	2048	88.30	0.168 [0.166, 0.170]	0.118 [0.117, 0.119]
PointConv	512	83.23	0.169 [0.166, 0.172]	–
PointConv	1024	86.16	0.116 [0.114, 0.118]	0.176 [0.173, 0.179]
PointConv	2048	86.94	0.093 [0.092, 0.095]	0.126 [0.124, 0.128]
PointMamba	512	86.94	0.175 [0.170, 0.180]	–
PointMamba	1024	90.90	0.129 [0.125, 0.133]	0.224 [0.218, 0.231]
PointMamba	2048	90.52	0.139 [0.135, 0.143]	0.177 [0.172, 0.182]
PointGPT-S	512	85.95	0.199 [0.194, 0.203]	–
PointGPT-S	1024	89.79	0.166 [0.162, 0.170]	0.234 [0.228, 0.239]
PointGPT-S	2048	87.17	0.198 [0.193, 0.202]	0.234 [0.227, 0.240]

Table 16: CAD-derived source (HDF5 frame), nested IID uniform surface sequence. Non-1024 inputs are cardinality stress; internal hierarchy counts remain at their reference values.

Model	n	Acc. (%)	Repeatability R [95% CI]	Refinement R [95% CI]
PointNet	512	89.02	0.171 [0.168, 0.173]	–
PointNet	1024	89.76	0.158 [0.156, 0.160]	0.172 [0.170, 0.175]
PointNet	2048	90.21	0.122 [0.120, 0.124]	0.131 [0.129, 0.133]
DGCNN	512	84.32	0.502 [0.499, 0.505]	–
DGCNN	1024	91.22	0.479 [0.476, 0.482]	0.642 [0.638, 0.646]
DGCNN	2048	88.63	0.446 [0.443, 0.448]	0.568 [0.565, 0.572]
PointMLP	512	92.09	0.152 [0.149, 0.155]	–
PointMLP	1024	91.99	0.184 [0.180, 0.187]	0.201 [0.197, 0.205]
PointMLP	2048	90.87	0.217 [0.213, 0.221]	0.222 [0.218, 0.227]
PointNet++	512	88.71	0.149 [0.147, 0.151]	–
PointNet++	1024	88.25	0.148 [0.146, 0.150]	0.131 [0.129, 0.133]
PointNet++	2048	88.48	0.142 [0.140, 0.144]	0.108 [0.107, 0.110]
PointConv	512	84.29	0.114 [0.112, 0.116]	–
PointConv	1024	86.55	0.099 [0.098, 0.101]	0.146 [0.144, 0.149]
PointConv	2048	87.18	0.086 [0.085, 0.088]	0.117 [0.115, 0.119]
PointMamba	512	89.95	0.087 [0.084, 0.089]	–
PointMamba	1024	92.03	0.100 [0.097, 0.103]	0.166 [0.161, 0.171]
PointMamba	2048	90.92	0.122 [0.118, 0.125]	0.164 [0.159, 0.169]
PointGPT-S	512	89.15	0.131 [0.128, 0.135]	–
PointGPT-S	1024	91.18	0.135 [0.131, 0.139]	0.193 [0.188, 0.199]
PointGPT-S	2048	88.24	0.179 [0.174, 0.183]	0.226 [0.220, 0.232]

Table 17: CAD-derived source (HDF5 frame), nested coverage sequence with 256 retained anchors. Refinement compares adjacent cardinalities, and repeatability compares separately sampled sequences sharing their scaffold.

Model	n	Acc. (%)	Repeatability R [95% CI]	Refinement R [95% CI]
PointNet	512	88.10	0.324 [0.320, 0.328]	–
PointNet	1024	89.47	0.198 [0.195, 0.200]	0.249 [0.246, 0.253]
PointNet	2048	90.18	0.116 [0.115, 0.118]	0.151 [0.149, 0.153]
DGCNN	512	77.69	0.612 [0.608, 0.615]	–
DGCNN	1024	88.76	0.517 [0.514, 0.520]	0.691 [0.688, 0.695]
DGCNN	2048	85.32	0.453 [0.450, 0.456]	0.593 [0.590, 0.596]
PointMLP	512	90.42	0.253 [0.248, 0.257]	–
PointMLP	1024	90.95	0.221 [0.217, 0.225]	0.245 [0.240, 0.249]
PointMLP	2048	90.06	0.232 [0.228, 0.236]	0.241 [0.236, 0.245]
PointNet++	512	92.18	0.193 [0.191, 0.195]	–
PointNet++	1024	92.30	0.163 [0.161, 0.165]	0.136 [0.134, 0.138]
PointNet++	2048	92.34	0.149 [0.147, 0.151]	0.107 [0.106, 0.109]
PointConv	512	90.48	0.144 [0.141, 0.146]	–
PointConv	1024	91.76	0.095 [0.093, 0.096]	0.163 [0.160, 0.166]
PointConv	2048	92.03	0.077 [0.075, 0.078]	0.109 [0.107, 0.111]
PointMamba	512	89.53	0.138 [0.134, 0.142]	–
PointMamba	1024	92.26	0.110 [0.106, 0.113]	0.188 [0.183, 0.193]
PointMamba	2048	91.29	0.124 [0.121, 0.128]	0.162 [0.157, 0.167]
PointGPT-S	512	90.25	0.163 [0.159, 0.166]	–
PointGPT-S	1024	92.42	0.140 [0.136, 0.143]	0.195 [0.190, 0.200]
PointGPT-S	2048	89.87	0.184 [0.180, 0.188]	0.219 [0.213, 0.224]

Table 18: Released 10,000-row pool-derived source, nested uniform sampling without replacement (the implementation's `nested_iid` protocol label). The finite reservoir does not establish a continuum limit.

Model	n	Acc. (%)	Repeatability R [95% CI]	Refinement R [95% CI]
PointNet	512	89.42	0.144 [0.142, 0.146]	–
PointNet	1024	90.14	0.127 [0.125, 0.129]	0.152 [0.150, 0.154]
PointNet	2048	90.30	0.089 [0.087, 0.090]	0.112 [0.110, 0.113]
DGCNN	512	83.74	0.495 [0.492, 0.498]	–
DGCNN	1024	90.86	0.463 [0.460, 0.466]	0.644 [0.640, 0.649]
DGCNN	2048	86.67	0.430 [0.428, 0.433]	0.579 [0.576, 0.583]
PointMLP	512	92.52	0.133 [0.130, 0.135]	–
PointMLP	1024	92.71	0.155 [0.152, 0.158]	0.184 [0.181, 0.188]
PointMLP	2048	90.73	0.182 [0.179, 0.185]	0.204 [0.200, 0.207]
PointNet++	512	92.52	0.132 [0.131, 0.134]	–
PointNet++	1024	92.65	0.124 [0.123, 0.126]	0.111 [0.110, 0.113]
PointNet++	2048	92.49	0.114 [0.112, 0.115]	0.073 [0.072, 0.074]
PointConv	512	91.13	0.089 [0.087, 0.091]	–
PointConv	1024	92.17	0.067 [0.066, 0.069]	0.137 [0.135, 0.140]
PointConv	2048	92.07	0.052 [0.051, 0.053]	0.094 [0.092, 0.096]
PointMamba	512	91.13	0.062 [0.060, 0.064]	–
PointMamba	1024	92.95	0.064 [0.062, 0.066]	0.151 [0.146, 0.155]
PointMamba	2048	91.76	0.074 [0.072, 0.077]	0.143 [0.139, 0.148]
PointGPT-S	512	91.40	0.068 [0.065, 0.070]	–
PointGPT-S	1024	93.08	0.071 [0.069, 0.073]	0.158 [0.153, 0.163]
PointGPT-S	2048	90.46	0.094 [0.092, 0.097]	0.185 [0.179, 0.190]

Table 19: Released 10,000-row pool-derived source, nested coverage sequence. All sizes, models, and paired measurements use the same frozen reservoir transform.

H.5 RELIABILITY CONTRASTS UNDER CARDINALITY CHANGES

Table 20 compares three geometry-oriented baselines with the other four checkpoints under coverage-controlled $1024 \to 2048$ refinement. Every listed difference is negative and its pointwise 95% paired interval excludes zero. Thus PointNet, PointNet++, and PointConv have smaller mean R in this condition on both controlled sources. This is a paired result at the object level, not an inference from overlap of marginal intervals.

Row model	Comparison	CAD difference [95% CI]	Pool difference [95% CI]
PointNet	DGCNN	-0.4372 [-0.4405, -0.4337]	-0.4677 [-0.4707, -0.4645]
PointNet	PointMLP	-0.0914 [-0.0953, -0.0870]	-0.0919 [-0.0956, -0.0878]
PointNet	PointMamba	-0.0327 [-0.0374, -0.0276]	-0.0314 [-0.0356, -0.0270]
PointNet	PointGPT-S	-0.0947 [-0.1005, -0.0888]	-0.0732 [-0.0786, -0.0677]
PointNet++	DGCNN	-0.4598 [-0.4631, -0.4565]	-0.5065 [-0.5097, -0.5033]
PointNet++	PointMLP	-0.1140 [-0.1178, -0.1099]	-0.1307 [-0.1346, -0.1270]
PointNet++	PointMamba	-0.0553 [-0.0599, -0.0503]	-0.0702 [-0.0746, -0.0658]
PointNet++	PointGPT-S	-0.1174 [-0.1230, -0.1114]	-0.1120 [-0.1174, -0.1066]
PointConv	DGCNN	-0.4511 [-0.4549, -0.4475]	-0.4854 [-0.4888, -0.4822]
PointConv	PointMLP	-0.1053 [-0.1095, -0.1009]	-0.1097 [-0.1133, -0.1060]
PointConv	PointMamba	-0.0466 [-0.0512, -0.0416]	-0.0492 [-0.0533, -0.0452]
PointConv	PointGPT-S	-0.1087 [-0.1149, -0.1029]	-0.0909 [-0.0961, -0.0859]

Table 20: Coverage-controlled $1024 \to 2048$ refinement: mean paired difference in R, row model minus comparison model, and pointwise 95% class-stratified object-cluster bootstrap interval. Negative values indicate smaller R for the row model. Each source uses 2,468 objects; intervals are not simultaneous.

Across both sources, both nested protocols, both transitions, and the four comparison models, PointNet++ has a negative interval in all 32 pairings, as does PointConv. PointNet's ordering is less uniform: at the IID $512 \to 1024$ transition it has larger mean R than PointMLP, PointMamba, and PointGPT-S on both sources. At coverage-controlled $512 \to 1024$, PointMamba has lower mean R on the CAD-derived source, while the corresponding pool-derived interval includes zero. These exceptions preclude an unconditional three-versus-four architecture ordering. The individual intervals also do not form a multiple-comparison-adjusted global test.

	CAD-derived			Released 10k-pool		
Model	1024	2048	Δ	1024	2048	Δ
PointNet	89.76	90.21	+0.45	90.14	90.30	+0.16
DGCNN	91.22	88.63	-2.59	90.86	86.67	-4.19
PointMLP	91.99	90.87	-1.12	92.71	90.73	-1.97
PointNet++	88.25	88.48	+0.23	92.65	92.49	-0.16
PointConv	86.55	87.18	+0.63	92.17	92.07	-0.09
PointMamba	92.03	90.92	-1.11	92.95	91.76	-1.19
PointGPT-S	91.18	88.24	-2.94	93.08	90.46	-2.62

Table 21: Coverage-controlled accuracy before and after $1024 \to 2048$ refinement. Accuracy is in percent and Δ is a signed change in percentage points. Logit reliability and accuracy answer different questions.

For PointMLP, PointMamba, and PointGPT-S the accuracy changes are measured effects of finite refinement, not evidence of asymptotic failure. Under CAD-derived coverage control, their $1024 \to 2048$ changes are approximately -1.12, -1.11, and -2.94 percentage points, respectively; on the pool-derived source the corresponding changes are -1.97, -1.19, and -2.62 points. Their relative-logit responses provide a second scale on which to inspect these shifts. The fixed-geometry and learned-aggregation analysis in Appendices D.6–D.9 explains why a finite stable regime can occur without a structural guarantee.

I EXPERIMENT 3: DUPLICATE-TRACK SENSITIVITY IN FORECASTING

I.1 QUESTION AND STATISTICAL ESTIMAND

The experiment measures whether redundant observations of a known pedestrian change forecast errors for the original agents. The physical entity configuration and ground-truth futures are fixed, while the input registration is changed. This is a reliability study in the measurement-error sense (Bartlett & Frost, 2008): the primary quantities are paired within-window absolute changes, with signed changes and task error reported alongside them. It is not a study of the frequency of duplicate detections in a deployed tracker.

Three released forecasting systems are evaluated: Social GAN, Social-STGCNN, and AgentFormer (Gupta et al., 2018; Mohamed et al., 2020; Yuan et al., 2021). Each has five benchmark folds, ten conditions, and three target roles, yielding 180 summary rows per model. The main results are descriptive five-fold macro-averages. The inference coupling is specified separately for each model, because stochastic candidate generation and selection are part of the implemented predictor.

I.2 DATA, WINDOWS, AND EVALUATION ROLES

The five test folds are ETH and HOTEL from the ETH benchmark (Pellegrini et al., 2009), and UNIV, ZARA1, and ZARA2 from UCY (Lerner et al., 2007). The evaluation recipe uses the processed Social GAN test files, with frame number, pedestrian identifier, and two spatial coordinates in meters. Coordinates are rounded to four decimal places before window construction. Each window comprises eight observed and twelve future frames; an eligible pedestrian has all 20 entries across the window. Windows advance by one frame in the processed sequence and contain at least two eligible original pedestrians. The benchmark's nominal sampling interval is 0.4 seconds.

Fold	Original agent–windows	Source agent–windows	Other agent–windows
ETH	181	70	111
HOTEL	1,053	301	752
UNIV	24,334	947	23,387
ZARA1	2,253	602	1,651
ZARA2	5,833	921	4,912
Total	33,654	2,841	30,813

Table 22: Evaluation counts. One source pedestrian is chosen per scene window, so the source column also counts windows. Overlapping windows and repeated appearances of a pedestrian are not independent observations.

One source pedestrian is chosen uniformly from the eligible agents in each window using a fixed random stream with seed 7. The `source` role scores that original pedestrian after copies are added; `other` scores the other original pedestrians; `all` scores their union. Added records never become targets. Model comparisons use each pipeline's own baseline and within-window pairing; no cross-model identity of the random source selections is required by the reported macro statistics.

I.3 CORRUPTION AND COUPLING ACROSS CONDITIONS

For source history $x_{1:8}^{(j)}$, additional record ℓ has observed positions

$$\widetilde{x}_t^{(\ell)} = x_t^{(j)} + \sigma U_{\ell,t}, \qquad U_{\ell,t} \sim \mathcal{N}(0, I_2), \quad t = 1, \ldots, 8. \tag{50}$$

A seed-0 stream generates one standardized noise array per window up to the largest duplicate count. Its prefixes are shared across k, and changing σ scales the same array. Thus the first added track at $k = 1$ is retained at $k = 2$ and 4. All original observations and ground-truth futures remain unchanged.

The ten conditions are the baseline $(k, \sigma) = (0, 0)$ and $k \in \{1, 2, 4\}$ crossed with $\sigma \in \{0, 0.02, 0.05\}$. Exact duplication uses $\sigma = 0$. For near duplicates, σ is the coordinatewise standard deviation in meters. Independent jitter at each observed time perturbs both positions and apparent velocities. It

is not a rigid translation or a deterministic history-distance bound. The known source association makes these additional tracks redundant records of one entity by construction.

I.4 Baseline selection and fixed inference paths

Twenty candidate futures are used for each baseline. ADE- and FDE-minimizing indices are selected separately and can differ. After perturbing the scene, these indices are held fixed rather than selecting a new best candidate. This prevents the intervention from being partly hidden by changing the selected hypothesis. Social GAN selects a candidate jointly for a scene; the other two pipelines select per original agent. Accordingly, the baseline and sensitivity statistics refer to their respective evaluation conventions, not a common candidate-selection rule imposed retrospectively.

Model	Baseline candidate selection	Quantity fixed for corrupted input
Social GAN	Scene-total ADE and FDE, separately	Selected scene latent codes; original targets
Social-STGCNN	Per-agent ADE and FDE, separately	Gaussian innovations for original agents; selected indices
AgentFormer	Per-agent ADE and FDE, separately	Baseline DLow proposals for original agents; selected indices

Table 23: The coupling defines the sensitivity estimand. Each pipeline recomputes predictions using the corrupted context while preserving the stated baseline quantities.

I.5 Social GAN: released 20VP evaluator and scene coupling

The Social GAN baseline is obtained through the released 20VP evaluator and its data loader. The 20 forward passes and their latent draws are recorded from the official evaluation batches. Relative predictions are accumulated into absolute positions for metric computation. Within each scene, the baseline index minimizing total ADE and the index minimizing total FDE are retained. The corresponding scene latent codes are then reused when predicting from each corrupted observation array, and outputs are restricted to the original agents. The generator configuration is the released checkpoint's configuration. If the two metric indices coincide, one conditional prediction supplies both metrics.

The use of a common scene code is relevant to duplicate consistency: a duplicate history does not acquire an independent latent identity under this coupling. The measured error changes for exact duplication are approximately one to two millionths of a meter at the reporting scale. The structural analysis below explains why max context aggregation is compatible with this behavior, without turning rounded error equality into a claim of tensor equality.

I.6 Social-STGCNN: graph and common Gaussian innovations

The model outputs bivariate Gaussian parameters for future displacements. Let $r_{t,i}$ denote observed displacement. The evaluation graph has weights

$$W_{t,ij} = \begin{cases} 1, & i = j, \\ \|r_{t,i} - r_{t,j}\|_2^{-1}, & i \neq j,\ r_{t,i} \neq r_{t,j}, \\ 0, & i \neq j,\ r_{t,i} = r_{t,j}, \end{cases} \qquad L_t = D_t^{-1/2}(D_t - W_t)D_t^{-1/2}, \tag{51}$$

where D_t contains row sums. Thus exact copies do not cause a division by zero, but they can change the degree normalization and the interactions with other agents.

For each original window, a seed-1 stream generates standard-normal innovations of shape $20 \times 12 \times n \times 2$. Gaussian sampling uses the predicted means and a two-dimensional Cholesky factor; sampled displacements are accumulated from each original agent's last observed position. The baseline ADE- and FDE-minimizing candidate indices are chosen per agent. After corruption, the graph and distribution parameters are recomputed, the original agents' parameter blocks are extracted, and the same innovations and indices are used. This controls sampling randomness while permitting the interaction graph and predicted motion distribution to respond to the registration change.

I.7 AGENTFORMER: CONTEXT CHANGES WITH FROZEN DLOW PROPOSALS

The baseline context encoder supplies the input to the DLow proposal network (Yuan & Kitani, 2020). Its mean-proposal path constructs 20 latent proposals per agent; call the resulting tensor z_0. For every corrupted scene, the context is recomputed. Original agents retain their proposal blocks from z_0, and each added record inherits its source agent's block. The autoregressive future decoder predicts from this new context and the fixed latent proposals. Baseline ADE- and FDE-minimizing indices are chosen separately per original agent and reused.

This is a conditional decoder/context reliability estimand: it measures sensitivity along the baseline proposal paths when redundant records modify the social context. Recomputing the context-dependent DLow proposals would define a different experiment. Common proposals also do not by themselves equate the complete predictive laws of two scenes; the distinction between pathwise and distributional consistency is given in Appendix A.

I.8 WHY A MAX INTERACTION CAN PRESERVE EXACT DUPLICATES

A useful idealization of social max pooling is

$$P_i(\mathbf{h}, \mathbf{x}) = \max_j \phi(h_j, x_j - x_i), \tag{52}$$

with shared pairwise feature map ϕ and coordinatewise max. If a source record is copied with the same position and hidden feature, every original agent receives a repeated copy of an already-present pairwise feature. Its pooled context therefore remains unchanged. A shared deterministic encoder of equal observed histories produces equal features, and a shared scene latent code preserves that equality at decoder initialization.

If the decoder is deterministic conditional on this state and code, with context pooled only at initialization, original-agent predictions remain equal by identical recurrent updates. If the same max interaction is recomputed at later times, the argument extends by induction provided copies retain the same state and position as their sources at each step. Independent per-copy noise, multiplicity-dependent normalization, or an exclusion rule that introduces a previously absent pairwise feature can invalidate the premise. The identity is conditional on this complete inference structure.

For near duplicates, the pairwise features can differ and the exact max identity no longer applies. A continuity bound would require Lipschitz control of the encoder, interaction features, and recurrent evolution, and boundedness where products occur. Thus the increase in Social GAN's measured sensitivity under jitter is compatible with exact-duplicate stability. Conversely, normalized graph weights and attention weights can depend on multiplicity, but neither operator alone establishes the response of every trained forecasting pipeline.

I.9 METRICS AND RELATIVE CHANGES

For a selected forecast $\widehat{y}_{1:T}$ and target $y_{1:T}$, with $T = 12$,

$$\mathrm{ADE} = \frac{1}{T}\sum_{t=1}^{T} \|\widehat{y}_t - y_t\|_2, \qquad \mathrm{FDE} = \|\widehat{y}_T - y_T\|_2. \tag{53}$$

Both are measured in meters. For metric M, original agent–window i in fold f has paired change $\Delta M_{fi} = M_{1,fi} - M_{0,fi}$. For each role, define

$$B_f = \frac{1}{N_f}\sum_i M_{0,fi}, \qquad S_f = \frac{1}{N_f}\sum_i \Delta M_{fi}, \qquad A_f = \frac{1}{N_f}\sum_i |\Delta M_{fi}|. \tag{54}$$

The five-fold macro-averages are $B = \frac{1}{5}\sum_f B_f$, $S = \frac{1}{5}\sum_f S_f$, and $A = \frac{1}{5}\sum_f A_f$. Counts in the `AVG` rows are summed, while metric columns are unweighted fold means. We report

$$\mathrm{RelAbs}(M) = 100\,\frac{A}{B}\,\%. \tag{55}$$

The ratio is computed before rounding the summaries. It is a ratio of macro-averages, not a signed percentage deterioration, the mean of fold percentages, or the mean of individual percentages. The source/other roles use their own baseline denominators. Equal fold weighting prevents the much larger UNIV count from determining the macro result.

Model	Baseline ADE	Baseline FDE
Social GAN	0.533	1.044
Social-STGCNN	0.454	0.760
AgentFormer	0.234	0.403

Table 24: Five-fold macro-average baseline errors in meters for original agents under the stated candidate-selection conventions.

I.10 Error sensitivity, displacement, and interpretation

The absolute value is taken before aggregation to avoid cancellation between increased and decreased errors. Thus $A \geq |S|$, and a positive relative absolute change is not itself a percentage loss of predictive accuracy. The expanded tables retain signed changes as well.

For baseline and perturbed forecasts of the same target, the reverse triangle inequality gives

$$|\Delta\mathrm{ADE}| \leq \frac{1}{T}\sum_{t} \|\widehat{y}_t^1 - \widehat{y}_t^0\|_2, \tag{56}$$

$$|\Delta\mathrm{FDE}| \leq \|\widehat{y}_T^1 - \widehat{y}_T^0\|_2. \tag{57}$$

Consequently a nonzero error change certifies a change in the selected prediction and lower-bounds its corresponding displacement, but equal errors do not imply equal forecasts. Two forecasts on opposite sides of a target can have identical distance to it. The reported ratios therefore measure error reliability on the selected paths, not a distance between full forecast distributions.

Two exact duplicates yield AgentFormer relative absolute ADE/FDE changes of $14.77\%/22.68\%$, despite baseline errors of $0.234/0.403$ meters. Four raise these values to $23.53\%/35.58\%$. Social-STGCNN has $25.63\%/33.32\%$ for two exact copies, whereas Social GAN is near the reporting floor. For two copies with $\sigma = 0.05$, Social GAN changes by $2.19\%/2.51\%$, Social-STGCNN by $24.20\%/31.69\%$, and AgentFormer by $15.24\%/23.61\%$. This separates strong baseline accuracy from agreement under registration changes.

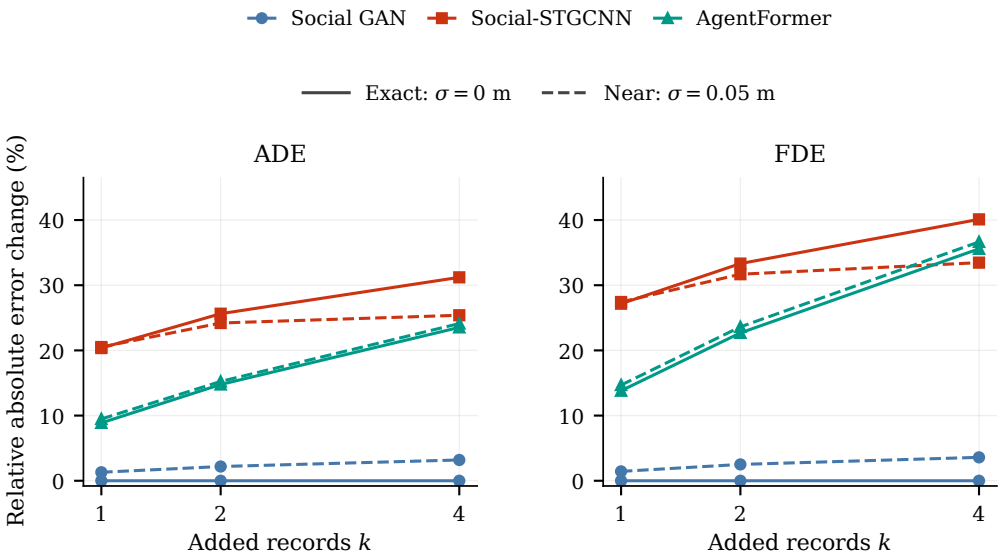


Figure 8: Relative absolute ADE/FDE changes versus duplicate count. Solid lines denote exact copies; dashed lines use $\sigma = 0.05$ meters. Curves show descriptive five-fold macro-averages. Each model's baseline denominator is given in Table 24.

The estimates describe the five benchmark folds under the stated source-selection and inference couplings. Overlapping windows are not treated as independent replicates, and no population

significance claim is made from their count. The experiment isolates downstream response to known redundant records, rather than determining real-world duplication prevalence or attributing sensitivity to one layer. Its conclusion is that registration reliability is a separate empirical property from clean forecasting accuracy.

I.11 ALL CONDITIONS AND ROLES: SOCIAL GAN

Role	k	σ	ΔADE	ΔFDE	$\lvert\Delta\rvert$ADE	$\lvert\Delta\rvert$FDE	Rel. ADE	Rel. FDE
all	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
all	1	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
all	1	0.02	0.0002	0.0008	0.0030	0.0069	0.57	0.66
all	1	0.05	0.0013	0.0034	0.0069	0.0150	1.30	1.44
all	2	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
all	2	0.02	0.0006	0.0021	0.0049	0.0114	0.92	1.09
all	2	0.05	0.0034	0.0082	0.0116	0.0262	2.19	2.51
all	4	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
all	4	0.02	0.0008	0.0030	0.0071	0.0162	1.34	1.55
all	4	0.05	0.0049	0.0128	0.0170	0.0375	3.19	3.59
source	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
source	1	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
source	1	0.02	0.0005	0.0027	0.0079	0.0177	1.55	1.78
source	1	0.05	0.0038	0.0104	0.0173	0.0366	3.41	3.68
source	2	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
source	2	0.02	0.0015	0.0059	0.0127	0.0292	2.50	2.94
source	2	0.05	0.0092	0.0235	0.0292	0.0652	5.78	6.56
source	4	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
source	4	0.02	0.0022	0.0084	0.0181	0.0406	3.58	4.09
source	4	0.05	0.0135	0.0356	0.0417	0.0919	8.23	9.25
other	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
other	1	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
other	1	0.02	0.0001	-0.0001	0.0011	0.0025	0.20	0.23
other	1	0.05	0.0002	0.0000	0.0028	0.0063	0.52	0.59
other	2	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
other	2	0.02	0.0002	0.0001	0.0017	0.0037	0.31	0.35
other	2	0.05	0.0006	0.0005	0.0043	0.0095	0.79	0.89
other	4	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
other	4	0.02	0.0002	0.0003	0.0026	0.0057	0.48	0.53
other	4	0.05	0.0008	0.0015	0.0067	0.0141	1.23	1.32

Table 25: Social GAN macro-averages. Δ columns are mean signed changes; $\lvert\Delta\rvert$ columns are means of individual absolute changes. Relative columns are percentages using the role-specific macro baseline. Zeros are rounded from the stored precision.

The role-specific percentages need not combine into the all-agent percentage by the total source/other counts: the fold means are first computed within role and then macro-averaged. They should be read with their own denominators.

I.12 ALL CONDITIONS AND ROLES: SOCIAL-STGCNN

Role	k	σ	ΔADE	ΔFDE	$\lvert\Delta\rvert$ADE	$\lvert\Delta\rvert$FDE	Rel. ADE	Rel. FDE
all	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
all	1	0.00	0.0465	0.0797	0.0923	0.2063	20.36	27.16
all	1	0.02	0.0459	0.0832	0.0947	0.2143	20.88	28.22
all	1	0.05	0.0463	0.0821	0.0930	0.2083	20.52	27.42
all	2	0.00	0.0684	0.1210	0.1162	0.2531	25.63	33.32
all	2	0.02	0.0615	0.1116	0.1125	0.2502	24.80	32.94
all	2	0.05	0.0609	0.1088	0.1098	0.2407	24.20	31.69
all	4	0.00	0.0895	0.1638	0.1415	0.3046	31.21	40.10
all	4	0.02	0.0652	0.1196	0.1188	0.2665	26.20	35.09
all	4	0.05	0.0643	0.1167	0.1152	0.2542	25.39	33.46
source	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
source	1	0.00	0.0623	0.1061	0.1249	0.2852	26.54	35.83
source	1	0.02	0.0368	0.0719	0.1041	0.2461	22.12	30.92
source	1	0.05	0.0413	0.0760	0.1051	0.2471	22.35	31.04
source	2	0.00	0.1029	0.1793	0.1681	0.3699	35.72	46.48
source	2	0.02	0.0566	0.1052	0.1289	0.2957	27.39	37.15
source	2	0.05	0.0610	0.1105	0.1277	0.2909	27.13	36.55
source	4	0.00	0.1338	0.2387	0.2036	0.4438	43.27	55.76
source	4	0.02	0.0587	0.1121	0.1351	0.3161	28.71	39.71
source	4	0.05	0.0635	0.1190	0.1344	0.3088	28.56	38.79
other	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
other	1	0.00	0.0411	0.0729	0.0787	0.1710	17.60	22.87
other	1	0.02	0.0520	0.0928	0.0943	0.2068	21.07	27.66
other	1	0.05	0.0503	0.0893	0.0899	0.1940	20.08	25.94
other	2	0.00	0.0551	0.1020	0.0949	0.2031	21.21	27.17
other	2	0.02	0.0654	0.1195	0.1098	0.2387	24.55	31.93
other	2	0.05	0.0623	0.1126	0.1048	0.2231	23.42	29.85
other	4	0.00	0.0718	0.1381	0.1160	0.2449	25.91	32.76
other	4	0.02	0.0700	0.1287	0.1175	0.2555	26.25	34.18
other	4	0.05	0.0660	0.1204	0.1106	0.2367	24.71	31.67

Table 26: Social-STGCNN macro-averages, with the same signed/absolute distinction as Table 25. Relative changes use the absolute-change convention in equation 55.

Near-duplicate effects need not increase monotonically with σ in a finite nonlinear audit. In particular, a small perturbation can change graph weights or neighbor relations differently from an exact copy. Monotonicity is not assumed in the analysis.

I.13 ALL CONDITIONS AND ROLES: AGENTFORMER

Role	k	σ	ΔADE	ΔFDE	$\lvert\Delta\rvert$ADE	$\lvert\Delta\rvert$FDE	Rel. ADE	Rel. FDE
all	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
all	1	0.00	0.0063	0.0176	0.0208	0.0556	8.88	13.81
all	1	0.02	0.0064	0.0182	0.0210	0.0563	9.00	13.98
all	1	0.05	0.0071	0.0198	0.0222	0.0591	9.48	14.68
all	2	0.00	0.0157	0.0407	0.0346	0.0913	14.77	22.68
all	2	0.02	0.0157	0.0410	0.0348	0.0920	14.86	22.86
all	2	0.05	0.0162	0.0427	0.0356	0.0951	15.24	23.61
all	4	0.00	0.0327	0.0822	0.0551	0.1433	23.53	35.58
all	4	0.02	0.0327	0.0826	0.0553	0.1442	23.65	35.82
all	4	0.05	0.0333	0.0849	0.0564	0.1475	24.12	36.65
source	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
source	1	0.00	0.0066	0.0219	0.0258	0.0665	10.75	16.42
source	1	0.02	0.0067	0.0229	0.0261	0.0671	10.88	16.57
source	1	0.05	0.0074	0.0254	0.0275	0.0701	11.46	17.30
source	2	0.00	0.0146	0.0439	0.0390	0.1015	16.26	25.06
source	2	0.02	0.0146	0.0445	0.0393	0.1022	16.39	25.23
source	2	0.05	0.0151	0.0468	0.0403	0.1056	16.81	26.06
source	4	0.00	0.0290	0.0815	0.0578	0.1492	24.09	36.83
source	4	0.02	0.0290	0.0823	0.0581	0.1500	24.22	37.03
source	4	0.05	0.0295	0.0851	0.0592	0.1530	24.67	37.76
other	0	0.00	0.0000	0.0000	0.0000	0.0000	0.00	0.00
other	1	0.00	0.0064	0.0149	0.0194	0.0531	8.39	13.10
other	1	0.02	0.0066	0.0154	0.0197	0.0539	8.51	13.30
other	1	0.05	0.0072	0.0166	0.0208	0.0568	8.97	14.01
other	2	0.00	0.0163	0.0380	0.0336	0.0896	14.51	22.10
other	2	0.02	0.0164	0.0384	0.0338	0.0904	14.58	22.29
other	2	0.05	0.0171	0.0399	0.0347	0.0933	14.95	23.02
other	4	0.00	0.0343	0.0808	0.0550	0.1436	23.71	35.44
other	4	0.02	0.0344	0.0813	0.0552	0.1447	23.82	35.70
other	4	0.05	0.0352	0.0834	0.0564	0.1484	24.31	36.61

Table 27: AgentFormer macro-averages under frozen baseline DLow proposals and baseline-selected sample indices. These results concern the conditional inference path described above. Relative columns use absolute changes and role-specific baselines.

The strong baseline remains visible in Table 24. The ratios show sensitivity relative to that baseline without concealing that the raw changes are measured in the same spatial units as ADE and FDE.

I.14 FOLD-LEVEL EXACT-DUPLICATE RESULTS

The following tables use two exact duplicates. They expose how a macro-average can combine different baseline scales and different sensitivity across folds. All other fold/condition/role combinations are retained in the machine-readable summaries.

Fold	Role	N	Base ADE	Base FDE	$\lvert\Delta\rvert$ADE	$\lvert\Delta\rvert$FDE	Rel. ADE	Rel. FDE
ETH	all	181	0.7786	1.4268	0.0000	0.0000	0.00	0.00
HOTEL	all	1053	0.4353	0.8805	0.0000	0.0000	0.00	0.00
UNIV	all	24334	0.7465	1.4974	0.0000	0.0000	0.00	0.00
ZARA1	all	2253	0.3476	0.6959	0.0000	0.0000	0.00	0.00
ZARA2	all	5833	0.3555	0.7198	0.0000	0.0000	0.00	0.00
ETH	source	70	0.6955	1.2489	0.0000	0.0000	0.00	0.00
HOTEL	source	301	0.4278	0.8574	0.0000	0.0000	0.00	0.00
UNIV	source	947	0.7220	1.4650	0.0000	0.0000	0.00	0.00
ZARA1	source	602	0.3283	0.6678	0.0000	0.0000	0.00	0.00
ZARA2	source	921	0.3567	0.7285	0.0000	0.0000	0.00	0.00
ETH	other	111	0.8310	1.5391	0.0000	0.0000	0.00	0.00
HOTEL	other	752	0.4383	0.8897	0.0000	0.0000	0.00	0.00
UNIV	other	23387	0.7475	1.4987	0.0000	0.0000	0.00	0.00
ZARA1	other	1651	0.3546	0.7062	0.0000	0.0000	0.00	0.00
ZARA2	other	4912	0.3553	0.7182	0.0000	0.0000	0.00	0.00

Table 28: Social GAN, $k = 2, \sigma = 0$, by fold and role. Relative columns are fold-specific percentages; they are not averaged to form equation 55.

Fold	Role	N	Base ADE	Base FDE	$\lvert\Delta\rvert$ADE	$\lvert\Delta\rvert$FDE	Rel. ADE	Rel. FDE
ETH	all	181	0.7379	1.2206	0.3101	0.6072	42.03	49.74
HOTEL	all	1053	0.4053	0.6637	0.0653	0.1735	16.11	26.14
UNIV	all	24334	0.4887	0.9113	0.0186	0.0429	3.81	4.70
ZARA1	all	2253	0.3326	0.5208	0.1184	0.2785	35.59	53.48
ZARA2	all	5833	0.3031	0.4815	0.0688	0.1634	22.69	33.95
ETH	source	70	0.7698	1.2794	0.3997	0.8381	51.92	65.51
HOTEL	source	301	0.4208	0.6960	0.0743	0.1975	17.66	28.38
UNIV	source	947	0.5090	0.9530	0.0291	0.0642	5.71	6.74
ZARA1	source	602	0.3304	0.5152	0.2193	0.4820	66.36	93.55
ZARA2	source	921	0.3224	0.5364	0.1180	0.2679	36.60	49.94
ETH	other	111	0.7178	1.1836	0.2536	0.4615	35.34	38.99
HOTEL	other	752	0.3991	0.6508	0.0617	0.1639	15.46	25.18
UNIV	other	23387	0.4879	0.9096	0.0182	0.0420	3.73	4.62
ZARA1	other	1651	0.3335	0.5228	0.0816	0.2043	24.47	39.08
ZARA2	other	4912	0.2995	0.4711	0.0595	0.1439	19.88	30.53

Table 29: Social-STGCNN, $k = 2, \sigma = 0$. The count N is original agent–windows, not independent people.

Fold	Role	N	Base ADE	Base FDE	$\lvert\Delta\rvert$ADE	$\lvert\Delta\rvert$FDE	Rel. ADE	Rel. FDE
ETH	all	181	0.4563	0.7972	0.0417	0.1132	9.14	14.20
HOTEL	all	1053	0.1423	0.2218	0.0348	0.0871	24.43	39.26
UNIV	all	24334	0.2542	0.4539	0.0176	0.0485	6.92	10.68
ZARA1	all	2253	0.1761	0.3031	0.0543	0.1432	30.81	47.24
ZARA2	all	5833	0.1410	0.2371	0.0244	0.0647	17.35	27.30
ETH	source	70	0.4706	0.7647	0.0454	0.1168	9.64	15.27
HOTEL	source	301	0.1461	0.2307	0.0349	0.0852	23.90	36.95
UNIV	source	947	0.2595	0.4654	0.0270	0.0678	10.41	14.57
ZARA1	source	602	0.1675	0.2929	0.0548	0.1477	32.75	50.42
ZARA2	source	921	0.1559	0.2721	0.0329	0.0901	21.11	33.11
ETH	other	111	0.4473	0.8177	0.0394	0.1110	8.81	13.57
HOTEL	other	752	0.1407	0.2183	0.0347	0.0878	24.66	40.24
UNIV	other	23387	0.2540	0.4534	0.0172	0.0477	6.77	10.52
ZARA1	other	1651	0.1793	0.3068	0.0540	0.1415	30.15	46.14
ZARA2	other	4912	0.1382	0.2305	0.0229	0.0600	16.55	26.02

Table 30: AgentFormer, $k = 2$, $\sigma = 0$, under the conditional proposal coupling. Fold-specific relative changes vary substantially; the macro-average is not a claim that every scene has the same response.

For the all-agent role, AgentFormer's relative absolute ADE/FDE changes are approximately $9.14\%/14.20\%$ on ETH, $24.43\%/39.26\%$ on HOTEL, $6.92\%/10.68\%$ on UNIV, $30.81\%/47.24\%$ on ZARA1, and $17.35\%/27.30\%$ on ZARA2. These descriptive differences motivate reporting both fold-level values and the clearly defined macro statistic.

J REPRODUCIBILITY AND STATISTICAL INTERPRETATION

J.1 SUPPLEMENTARY MATERIAL

The supplementary archive contains separate directories for the three experiments, each with its implementation, execution instructions, and numerical summaries. The supplied tables permit inspection of the reported numerical results without repeating training or inference. Experiment 1 provides model-, run-, support-, and cardinality-level summaries and figures. Experiment 2 provides marginal reliability summaries, paired checkpoint contrasts, completion metadata, and a CPU-only script for regenerating the density and refinement figures. Experiment 3 provides fold-, condition-, and role-specific error summaries for the three forecasting models. Installation, resource retrieval, configurations, and execution commands are documented in the experiment-specific READMEs.

Full reruns obtain public datasets and released pretrained weights where required and regenerate the underlying training and audit artifacts. Individual multiplicity predictions, stored point-cloud registrations, and per-agent trajectory records are produced by these workflows. Analyses requiring these records, including recomputation of the point-cloud bootstrap intervals, require the corresponding run outputs. Large downloaded resources and intermediate artifacts are excluded from the compact archive. Seeds and execution controls specify the reproduction procedure; exact floating-point agreement can depend on software environments and compiled operators.

J.2 STATISTICAL UNITS AND INTERPRETATION

Table 31 summarizes the units of aggregation and the reported variation. Checkpoint selection and support aggregation are specified in Appendices F.2–F.5; point-cloud estimands, bootstrap intervals, and numerical checks in Appendices G.7–G.9; and trajectory evaluation, stochastic couplings, and relative changes in Appendices I.2–I.10.

Experiment	Aggregation unit	Reported variation
Maximum regression	Multiplicity profiles within support, supports within training run	Mean and sample standard deviation across ten training seeds
Point clouds	Three registration seeds averaged within each object	Pointwise class-stratified object-bootstrap intervals and paired checkpoint contrasts
Trajectory prediction	Original agent–windows within fold and role, then an equally weighted five-fold mean	Descriptive signed and absolute error changes and role-specific relative changes

Table 31: Statistical units and reported variation. Multiple registrations of an object and overlapping trajectory windows are not additional independent replicates.

The point-cloud intervals describe object-level variation conditional on the frozen checkpoints, calibration, and realized registrations; they are pointwise intervals across the reported conditions. The trajectory summaries are descriptive fold statistics. Neither the number of registrations nor the number of overlapping agent–windows supplies an independent sample size for a population significance claim.

The measurement-error literature distinguishes agreement of repeated measurements from reliability relative to between-subject variation (Bartlett & Frost, 2008). Here, *registration reliability* concerns stability under the declared registration interventions. Our metrics quantify paired agreement and task-error responses, rather than a variance-ratio reliability coefficient. Predictive accuracy is reported alongside these metrics; Experiment 2 additionally distinguishes resampling variability from repeated evaluation of an identical input.