EQUAL INFORMATION IS NOT ENOUGH: ENCODER RANKINGS DEPEND ON INTERFACE REALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Encoder comparisons are often interpreted architecturally once the information supplied to each model has been matched. We show that information matching is necessary but not sufficient: the same information can be realized through heterogeneous model interfaces in ways that induce different learning problems and different encoder rankings. We separate declared information $\phi(X)$ from its architecture-specific realization ρ_A, writing performance as $Y(A, \rho_A(\phi(X)); \mathcal{T})$ under a fixed learning protocol $\mathcal{T}$. In camera–LiDAR correspondence, all controlled inputs encode exactly the same occupied-cell set; an audit over $13{,}935$ validation frames finds zero mismatched cells. Holding architecture, parameter count, optimization, comparator, gallery and evaluation fixed, changing only the realization reverses SparseConv3D's ranking against a raster encoder on RADIATE ($-0.0729 \rightarrow +0.1206$) and Boreas ($-0.0115 \rightarrow +0.0618$); all four comparisons are resolved under the preregistered criterion. The effect extends to PointPillars on both datasets: on RADIATE an unresolved $+0.0073$ becomes a resolved $+0.0569$, and on Boreas an already-resolved $+0.0032$ becomes $+0.0451$, with direct realization effects of $+0.0496$ and $+0.0419$. The SparseConv3D–PointPillars ordering reverses on both datasets as well. We therefore distinguish information equivalence from interface realization and introduce a realization-aware Declare–Realize–Audit protocol. Architectural attribution requires controlling not only what information is available, but how that information is exposed to the learner.

1 INTRODUCTION

Claims that one encoder family outperforms another are claims about computation, but the families being compared rarely receive the same input. Each arrives with a native representation, and those representations carry different information about the same signal. The natural correction is to declare a common statistic of the raw observation, give it to every architecture, and read what survives as architectural. That correction is necessary, but not sufficient.

Two tensors can carry precisely the same declared information while exposing it through different representation geometries. A quantity can be explicit in a feature channel or implicit in the structure of the input; an architecture's inductive bias interacts with those two situations differently even though neither adds information. Matching *what* a model is told therefore leaves open *how* it is told, and that remaining freedom can decide the comparison.

Concretely: a submanifold-sparse encoder given the occupied cells of a bird's-eye grid, with feature channels canonicalized, *trails* a raster CNN by 0.0729 P@1. The same encoder, same parameters, same schedule, given the *same* occupied cells with the cell-centre coordinates those cells determine written into its native feature fields, *leads* the same raster CNN by 0.1206. Both comparisons are resolved under the preregistered criterion ($S = 13.61$ and 26.35). The reversal reproduces independently on a second platform, sensor suite and city ($-0.0115 \rightarrow +0.0618$). In PointPillars the same intervention does not reverse a sign but changes the verdict, from an unresolved $+0.0073$ to a resolved $+0.0569$.

The quantity an information-controlled comparison measures is therefore not $Y(A, \phi(X))$ but

$$Y\big(A,\ \rho_A(\phi(X));\ \mathcal{T}\big),$$

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

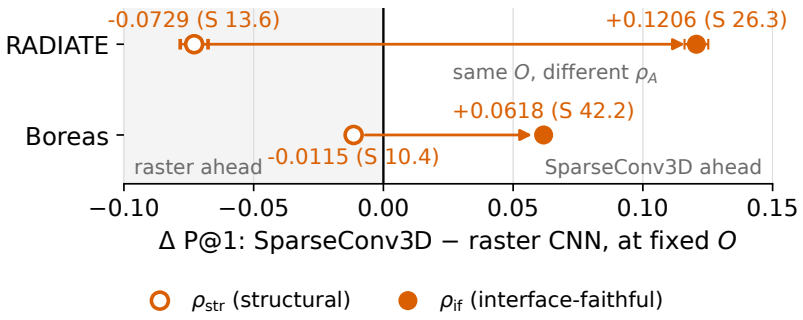

Figure 1: **The same declared information, two realizations, different conclusions.** Δ P@1 between SparseConv3D and a fixed raster comparator under each realization of the same occupied-cell set. Under a structural realization of the occupied-cell set the sparse encoder trails; under an interface-faithful realization of the *same* set it leads. Both realizations are deterministic from that set and preserve it, both pass the identical frame-wise audit at zero mismatched cells, and both use identical architectures and parameter counts. Bars are the pooled floor.

where ρ_A is the architecture-specific realization that turns the declared statistic into an object the interface accepts, and $\mathcal{T}$ is the fixed learning protocol.

Contributions.

- **Information matching is underdetermined.** We exhibit two realizations of one audited information statistic that select opposite encoders, both comparisons resolved, on two independent datasets. Fixing what an architecture is told does not fix the comparison (§5).
- **A realization-aware estimand and protocol.** We give the controlled quantity a realization argument, define a realization policy that makes ρ_A a stated choice rather than an implementation accident, and extend the audit from one property to two: no information outside ϕ survives, *and* the ϕ-determined native fields that each interface consumes are documented and classified rather than being silently discarded (§3).
- **The realization term can dominate.** Along the intervention path we measured, changing the realization moves P@1 by $+0.1936$ while removing the information ϕ genuinely discards moves it by -0.053, and the realization effect exceeds the spread of every SparseConv3D configuration we tested (§6).

2 Related Work

Fair comparison and hidden comparison variables. A line of work shows that architectural attribution depends on what is held fixed. Goyal et al. (2021) find that evaluation scheme and training details can obscure architecture effects in point-cloud classification, and that a simple projection baseline is competitive with elaborate architectures once those are matched. Liu et al. (2020) report substantially smaller differences among local aggregation operators after controlling the surrounding architecture and feature dimensions. Qian et al. (2022) raise PointNet++ from 77.9% to 86.1% on ScanObjectNN through improved training strategies alone, with further gains from scaling, and Wu et al. (2024) show that a simplified point-transformer design can scale to stronger performance. These works establish the broader methodological lesson that attribution depends on what is controlled; the variables they control are optimization, evaluation and scale. We address a distinct one: even after information content itself is matched, heterogeneous interfaces can realize that information differently.

Positional accessibility and representation geometry. That a model *can* recover a quantity does not mean it can use it equally well. Liu et al. (2018) show that convolutional networks fail at coordinate transforms that become easy once coordinates are supplied as explicit channels. Kayhan & van Gemert (2020) show that ordinary convolution can exploit absolute location through boundary effects, and Islam et al. (2020) that CNN representations encode substantial absolute position. These studies ask whether and how a model encodes or exploits position. We ask an attribution question

instead: when two realizations contain the same declared positional information, can changing only whether that information is explicit or structural alter the conclusion of an architecture comparison? Our result is not a new positional encoding module; it is a failure of comparison invariance under information-equivalent realizations.

Representation equivalence and 3D co-design. A separate literature asks when two representations carry compatible information or can be mapped into one another; Lenc & Vedaldi (2015) study equivalence and equivariance through learned transformations between representations. Our question is upstream and experimental: information-equivalent *inputs* can induce different learning geometry. In 3D systems, representation and computation are designed together: pillar encoders couple a learned point summary to a 2D trunk (Lang et al., 2019), sparse convolution operates over active sites (Graham et al., 2018; Choy et al., 2019a), hybrids retain a point branch alongside voxels (Tang et al., 2020; Shi et al., 2020), and recent detectors make sparsity and spatial support integral to the architecture (Chen et al., 2022; Fan et al., 2022; Chen et al., 2023). These systems optimize end-to-end representation–computation co-design; they do not aim to identify an architecture contrast under controlled, information-equivalent interface realizations.

Cross-modal correspondence. We train camera–LiDAR correspondence with a contrastive objective in the CPC/InfoNCE lineage (van den Oord et al., 2018). Related 2D-to-3D representation-learning methods such as SLidR and ST-SLidR contrast aligned image and LiDAR regions (Sautier et al., 2022; Mahmoud et al., 2023), while multimodal BEV systems fuse camera and LiDAR representations in a shared spatial space (Liang et al., 2022). We use correspondence as a measurement task rather than proposing a new cross-modal representation-learning method.

Interface conventions and usable information. Constant unit features on occupied sites are an established sparse-tensor convention (Choy et al., 2019b), while voxel encoders in the VoxelNet lineage aggregate measured point attributes into per-cell features (Zhou & Tuzel, 2018); the two conventions differ in exactly the fields our realization policy governs. Separately, a line of work formalizes information as usable only relative to a constrained predictor family (Xu et al., 2020; Ethayarajh et al., 2022). We borrow the intuition and none of the machinery: our claim is empirical and about interfaces, not a computational-information bound.

Positioning. Prior work has separately shown that fair comparison requires controlling training and evaluation variables, that positional accessibility changes what is learnable, and that representations with related information can differ geometrically. We are not aware of work connecting these into the attribution question studied here: whether an encoder ranking is invariant to multiple realizations of the same exactly audited input information while architecture, capacity, optimization and evaluation remain fixed.

3 Information equivalence is not learning equivalence

Declared information and its realization. Let X be the raw observation and $I_\phi = \phi(X)$ the statistic declared for comparison; here $I_\phi = O$, the set of occupied cells of a fixed grid. An architecture does not consume I_ϕ directly: it consumes an object of the type its interface accepts, produced by an architecture-specific realization $Z_A = \rho_A(I_\phi)$. Performance is therefore

$$Y\big(A,\ \rho_A(\phi(X));\ \mathcal{T}\big),$$

with $\mathcal{T}$ the fixed learning protocol (objective, optimizer, schedule, sampling and budget). Let $\rho = \{\rho_A\}_A$ denote a realization *policy*, the collection of architecture-specific realization maps it prescribes. The admissible class $\mathcal{R}_\phi$ is deliberately narrow rather than the set of all information-preserving bijections: an arbitrary bijection could encrypt O into channels no interface reads, which is information-equivalent but not a representation any implementer would write. We admit only realizations that write O-derived values into fields the target interface natively consumes. The controlled architecture contrast is then

$$\Delta_{A_1,A_2}(\phi,\rho;\mathcal{T}) \ = \ Y\big(A_1,\rho_{A_1}(\phi(X));\mathcal{T}\big) - Y\big(A_2,\rho_{A_2}(\phi(X));\mathcal{T}\big).$$

Information-equivalent realizations. Call $Z_A = \rho_A(I_\phi)$ *information-equivalent* with respect to I_ϕ when Z_A is a deterministic function of I_ϕ and I_ϕ remains recoverable from Z_A. The realization then neither introduces information outside ϕ nor destroys the declared statistic. Both realizations we compare satisfy this, by construction and by audit.

We study two admissible policies. The *structural* realization ρ_{str} makes cell identity available through the lattice, scatter or grid structure of the input while canonicalizing feature channels not required by O. The *interface-faithful* realization ρ_{if} additionally reconstructs, in the encoder's native fields, those quantities that are deterministic functions of O; fields requiring information outside O remain canonicalized. Neither changes the architecture or its dimensionality.

Realization invariance. Fix ϕ, a learning protocol $\mathcal{T}$, and an admissible class $\mathcal{R}_\phi$ of information-equivalent realizations. Call an architecture ordering *realization-invariant* over $\mathcal{R}_\phi$ if the sign of $\Delta_{A_1,A_2}(\phi, \rho; \mathcal{T})$ is the same for every $\rho \in \mathcal{R}_\phi$. Realization invariance is the property an information-matched comparison implicitly assumes whenever it reports a single ordering: without it, the ordering is a property of the pipeline as much as of the architectures. Our experiments exhibit two admissible realizations with opposite signs on each of two datasets, so the SparseConv3D–raster ordering is *not* realization-invariant even though information, architecture, capacity and protocol are fixed. PointPillars remains sign-invariant but is not verdict-invariant under the preregistered resolution criterion.

Why equivalence of information is weaker than equivalence of learning. Two realizations can carry the same information while placing it in different computational coordinates, and a finite learning protocol is not invariant to that choice. Information equivalence therefore constrains what is recoverable from the realized input, not what a particular architecture will learn from it under $\mathcal{T}$.

The protocol. *Declare* the statistic and state it. *Realize* it under a stated policy: reconstruct native-interface fields determined by ϕ, canonicalize fields requiring information outside ϕ, leave the architecture unchanged. *Audit* two properties rather than one: an *information audit*, that every non-constant quantity any encoder receives is deterministic from ϕ and that the declared statistic is preserved exactly, and a *realization audit*, that the native fields each interface consumes are documented as reconstructed or canonicalized. The second is what distinguishes this protocol from information matching alone.

4 A REALIZATION-AWARE CONTROLLED COMPARISON

Task and data. Camera–LiDAR patch correspondence retrieval on RADIATE (Sheeny et al., 2021), with an independent replication on Boreas (Burnett et al., 2023), a different platform, sensor suite and city, retrained in full with nothing transferred. Every arm consumes the same LiDAR returns, rasterized or pillarized by us from the same points. Protocol details in Appendix A.

Architectures. A raster CNN (dense and sparsity-invariant masked variants (Uhrig et al., 2017), $165{,}632$ BEV parameters), PointPillars ($206{,}720$), SparseConv3D ($206{,}840$) and a sparse transformer ($206{,}384$). The BEV parameter counts of the three sparse families agree within 0.3% of one another; the raster comparator is the smallest model in the study. All arms share an identical camera tower ($140{,}160$ parameters), optimizer and data order, and within every adjudicated comparison both arms use the identical schedule. Every arm reported here uses the same registered schedule, two Stage-0 and six Stage-1 epochs, so each adjudicated contrast and each ρ_{str}–ρ_{if} intervention is at equal training budget on both sides.

The intervention. Figure 2 states the protocol. Holding the architecture fixed, we compare ρ_{str} against ρ_{if}. The two differ only in the values written on occupied cells: under ρ_{if} the cell-centre coordinates that O determines are placed in the encoder's native feature fields; height and sub-cell offsets, which O does not determine, remain canonical constants under both. The raster comparator's input tensor is byte-identical under both.

The canonical constant, and what is scored. Under ρ_{str} each of the five per-site feature channels of the sparse arms carries the canonical constant 1.0 on an occupied cell; under ρ_{if} the two position fields carry cell-centre coordinates and the remaining three keep that constant. Token-level P@1 is scored on 300 RADIATE validation frames, fixed by index before any result existed, and on all $6{,}433$ Boreas validation frames. The audits cover the full $13{,}935$-frame validation population, a superset of the scored frames. Table 1 lists what the intervention holds fixed and Table 2 is the realization audit itself.

The transformer as an interface reference. Self-attention without positional encoding is permutation-equivariant and therefore does not by itself encode absolute token location. Its na-

Table 1: **What the realization intervention holds fixed.** The comparison in §5 varies exactly one thing. "Identical" means equal by construction and verified, not matched to a tolerance.

quantity	under ρ_{str} vs ρ_{if}
declared information	identical occupied-cell set O, audited frame-wise
architecture	identical computation graph
BEV parameters	identical (206,840 sparse, 206,720 pillar)
feature dimensionality	identical (five input channels)
optimizer, schedule, data order	identical
comparator	identical raster arm, never retrained
gallery and candidate support	identical
what changes	values of the native O-derivable positional fields

Table 2: **Realization audit.** Every non-constant quantity any encoder receives is a deterministic function of the occupied-cell set O: no measured height, intensity, point multiplicity or sub-cell position survives in any controlled arm. What differs is *where* the O-determined position appears, and whether the interface's own positional fields carry it. The final column classifies each field as O-derived, $g(O)$, or as a fixed constant carrying no positional content.

family	structural route	native fields	fields under this realization	source
Raster CNN	grid index	occupancy	binary occupancy	$g(O)$
PointPillars, ρ_{str}	scatter to (r, c)	x, y, z, intensity, sub-cell offset magnitude	constant	constant
PointPillars, ρ_{if}	scatter to (r, c)	as above	cell-centre x, y + constants	$g(O)$
SparseConv3D, ρ_{str}	sparse lattice index	x, y, z, intensity, sub-cell offset magnitude	constant	constant
SparseConv3D, ρ_{if}	sparse lattice index	as above	cell-centre x, y + constants	$g(O)$
Transformer	token cell index	five per-site channels + PE(r, c)	five constants + PE(r, c)	$g(O)$

tive interface consequently retains an O-derived sinusoidal encoding of the cell index. Removing that encoding would not yield a second admissible realization: the token multiset would then be a set of identical vectors from which O is not recoverable, violating the recoverability condition of §3. The transformer has no structural route to position that survives canonicalization, whereas lattice indices and scatter targets carry O exactly for the sparse and pillar families. It is therefore retained as a reference point, and the intervention is adjudicated on the two families where both policies are admissible.

Statistics. Token-level P@1 against a fixed gallery of $T = 576$ BEV cells, five seeds per arm; Boreas scored on its complete 6,433-pair validation set. For a comparison we report Δ, the pooled floor $\sqrt{f_a^2 + f_b^2}$ with $f_x = s_x/\sqrt{n}$, and $S = |\Delta|/\text{pooled}$, calling a comparison *resolved* iff $S > 2$. For equal n, S is algebraically $|t_{\mathrm{Welch}}|$. We retain the preregistered $S > 2$ rule; its near-threshold band ($1.5 \leq S \leq 2$) is kept in the supplement and unused here. We report Welch tests and bootstrap 95% intervals, from a single deterministic script, as conventional sensitivity analyses; they agree with the $S > 2$ verdict on every headline comparison. All primary adjudication criteria were registered before the corresponding results existed.

5 Interface realization changes the comparison

The direct realization effect. The cleanest quantity is $\Delta_\rho(A) = Y(A, \rho_{\mathrm{if}}(\phi(X)); \mathcal{T}) - Y(A, \rho_{\mathrm{str}}(\phi(X)); \mathcal{T})$, in which the same architecture appears on both sides. Table 3 gives Δ_ρ for both families on both datasets; all four are resolved (S from 6.26 to 51.00), with a magnitude that is family-dependent.

The consequence for the comparison. Because the realization effect can exceed the gap to the comparator, it can change the conclusion. For SparseConv3D the ordering reverses: -0.0729 becomes $+0.1206$ on RADIATE, and -0.0115 becomes $+0.0618$ on Boreas, with all four comparisons resolved. For PointPillars the sign does not change, but the verdict does: under ρ_{str} the correct state-

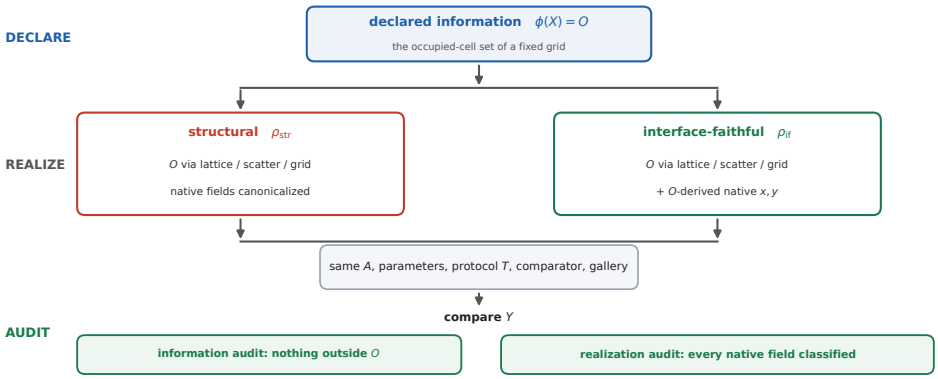

Figure 2: **Declare, realize, audit.** Two admissible realizations of one declared statistic; everything downstream is fixed. The audit checks information (nothing outside O) and realization (every native field classified).

Table 3: **Main result.** Δ is against the fixed raster comparator; Δ_ρ is the direct within-architecture realization effect, the same encoder on both sides. S in parentheses; 95% bootstrap intervals and Welch p are in supplement Table 4. RAD = RADIATE, BOR = Boreas.

family	data	ρ_{str} (S)	ρ_{if} (S)	Δ_ρ (S)	outcome
PointPillars	RAD	$+0.0073$ (1.35)	$+0.0569$ (8.61)	$+0.0496$ (6.26)	unresolved $\rightarrow$ resolved
SparseConv3D	RAD	-0.0729 (13.61)	$+0.1206$ (26.35)	$+0.1936$ (30.97)	**ranking reverses**
PointPillars	BOR	$+0.0032$ (3.00)	$+0.0451$ (32.61)	$+0.0419$ (31.88)	resolved $\rightarrow$ resolved
SparseConv3D	BOR	-0.0115 (10.44)	$+0.0618$ (42.19)	$+0.0732$ (51.00)	**ranking reverses**

ment is that no architectural advantage is resolved ($+0.0073$, $S = 1.35$), and under ρ_{if} it is that an advantage of $+0.0569$ is resolved ($S = 8.61$); on Boreas an already-resolved $+0.0032$ becomes $+0.0451$ ($S = 32.61$), with no reversal on either dataset, as registered. Same ϕ, same architecture, same capacity, different answers to whether a positive architectural contrast is statistically resolved.

Identification. Table 1 lists what the intervention holds fixed. Every model is scored from the final checkpoint of the registered schedule, with no early stopping and no validation-based selection. Controls on candidate support, gallery size, capacity and comparator choice (Appendix H) move no verdict; Table 4 lists the alternative explanations we tested. On Boreas, the SparseConv3D reversal and the PointPillars ρ_{if} advantage hold within each validation traversal separately (winter and autumn; supplement Table 5).

The structural realization is a standard interface convention. ρ_{str} is a legitimate realization rather than a degraded one. It satisfies the declared ϕ, passes the exact audit, and is what a careful implementer writes: constant unit features on occupied sites are used in established sparse-convolution pipelines such as FCGF (Choy et al., 2019b), which takes a 1-vector as its only input feature. Its value here is that it is information-valid. The claim is not that two equally correct implementations disagree, but that information equality alone admits realizations of differing interface fidelity, and the ranking changes across them.

The ranking, not just the gap. Figure 3 shows all four families under both policies, tabulated in the supplement. On RADIATE, under ρ_{str} the raster comparator leads SparseConv3D and is not resolved from PointPillars; under ρ_{if} it trails both, and SparseConv3D closes on the transformer from 0.204 to 0.0105. The raster comparison is not the only ordering that reverses: SparseConv3D trails PointPillars under ρ_{str} and leads it under ρ_{if}, on RADIATE (-0.0803, $S = 11.61$, to $+0.0637$, $S = 8.66$) and on Boreas (-0.0146, $S = 14.44$, to $+0.0167$, $S = 10.04$), resolved on both sides in both cases.

6 Representation geometry and architectural inductive bias

Geometry, and its interaction with inductive bias. Under ρ_{str}, cell identity lives in the structure of the input: which lattice sites are active, which scatter targets are written. Under ρ_{if} that identity

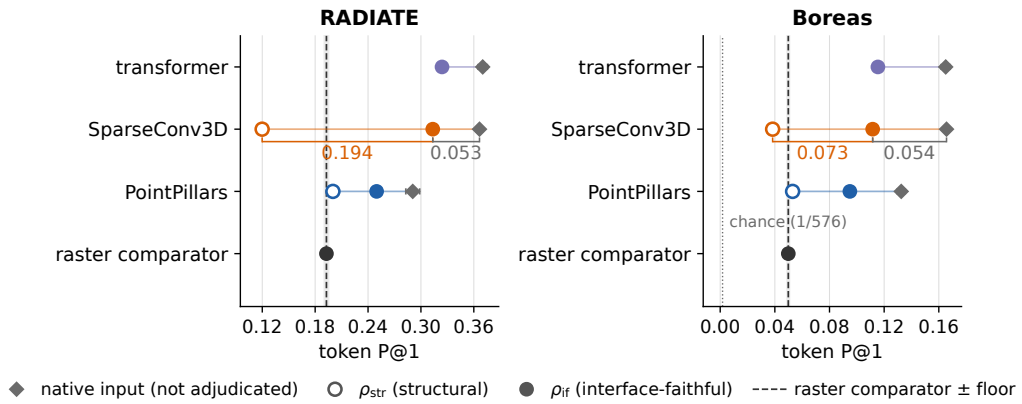

Figure 3: **Rankings are not preserved across realizations.** Token P@1 for all four families under the structural (open) and interface-faithful (filled) realizations of the same occupied-cell set; native-input values (grey diamonds) are shown for scale and are not adjudicated. Dashed line: raster comparator ± floor; dotted line on Boreas: chance ($1/576$). Brackets give step magnitudes for SparseConv3D: grey, native $\rightarrow \rho_{\mathrm{if}}$ (information ϕ discards); orange, $\rho_{\mathrm{if}} \rightarrow \rho_{\mathrm{str}}$ (realization at fixed O). On both datasets SparseConv3D moves from last to second, overtaking PointPillars as well as the comparator, and its realization step is at least as large as its information step. The transformer's native interface already carries an O-derived positional encoding and we define no second admissible realization under the present policy. Bars are ± 1 pooled floor; on Boreas they are smaller than the markers.

is additionally a value in a native feature channel. No bit of declared information distinguishes the two. What differs is the geometry through which learning must reach the information, and therefore how much computation stands between the input and a usable positional signal. Dense convolution, submanifold sparse convolution, scatter-then-convolve pillar encoding and attention differ in locality, in equivariance, and in how they can access absolute position at all, so a realization can align well with one inductive bias and poorly with another while adding nothing to the information content. That is what makes ρ an architecture-interacting variable rather than a benign preprocessing detail. Parameter matching fixes the architecture, but $f_\theta \circ \rho_{\mathrm{str}}$ and $f_\theta \circ \rho_{\mathrm{if}}$ induce different hypothesis families over O, so the two realizations do not present the learner with the same function class; we treat that difference as a candidate mediator rather than a mechanism established here.

Family-dependent sensitivity. The realization effect is not numerically uniform across architectures. On RADIATE, Δ_ρ is $+0.1936$ for SparseConv3D against $+0.0496$ for PointPillars, and the difference between the two is itself resolved in a post-hoc contrast of the two intervention effects ($+0.1440$, $S = 14.27$, CI $[+0.1262, +0.1612]$). The same contrast on Boreas, named in advance rather than post-hoc, replicates it ($+0.0313$, $S = 16.08$, CI $[+0.0278, +0.0347]$). Realization sensitivity is therefore architecture-dependent in magnitude on both datasets, not a fixed benefit of exposing coordinates.

A candidate mediator, stated but not established. The ρ_{str} ordering coincides with each interface's structural route to absolute position: an explicit positional encoding (transformer), a dense trunk whose zero-padding exposes location (PointPillars and the raster comparator; Kayhan & van Gemert, 2020; Islam et al., 2020), and a submanifold lattice with no such route (SparseConv3D). Consistent with this, the only configuration changes that raise SparseConv3D under ρ_{str} by a resolved margin are those that widen its route to position: regular sparse convolution ($0.120 \rightarrow 0.158$) and dilation ($\rightarrow 0.179$; supplement I). We treat positional accessibility as the leading candidate mediator and its identification as the next study.

Scale, against information and against architecture. Two comparisons place the effect in context. Moving SparseConv3D from its native input to the interface-faithful realization removes information that ϕ genuinely discards, namely height, intensity and sub-cell offsets, and costs -0.0530 on RADIATE against -0.0540 on Boreas. Changing only the realization at fixed O costs -0.1936 and -0.0732 respectively: a factor of 3.65 on RADIATE and 1.36 on Boreas along this path. The realization effect is also larger than the within-family configuration spread in our RADIATE sweep: six

Table 4: **Robustness summary.** Alternative explanations, and the control addressing each. Controls were run under ρ_{str} unless stated, so each row bounds an explanation for the ρ_{str} ordering rather than for Δ_ρ, which the within-architecture intervention identifies instead. No control restores a resolved advantage for the sparse families under ρ_{str}. Exact values in the supplement.

potential explanation	control	result
The declared information was not equal	frame-wise exact audit of O, all 13,935 validation frames	zero mismatched cells, both realizations
Gallery cells the sparse families cannot represent	gallery restricted per frame to occupied cells	all ρ_{str} verdicts unchanged; magnitudes move
The configurations chosen favour the comparator	six SparseConv3D variants against three raster configurations	disjoint envelopes, 0.120–0.179 against 0.192–0.201; $5/6$ resolved below the comparator
Gallery size or parameter budget sets the outcome	$T \in \{144, \ldots, 2304\}$; budget matched both directions	ρ_{str} gap resolved at no T ($S \leq 1.41$); matching the budget downward leaves a resolved disadvantage, -0.0101 ($S = 2.77$); matching upward leaves it unresolved, -0.0007 ($S = 0.12$)
Comparator choice	the dense raster arm throughout (post-hoc)	all signs and all $S > 2$ verdicts unchanged
Specific to one dataset	full repetition on Boreas	the reversal replicates (§5)

SparseConv3D configurations spanning aggregation, sparse connectivity and receptive-field structure occupy a range of 0.059 in P@1 (0.120 to 0.179 under ρ_{str}), against $+0.1936$ for the realization change alone. For PointPillars the two steps are instead comparable: 1.21 on RADIATE and 1.11 on Boreas. The realization term therefore dominates for one family and merely matches the information term for the other, which is why we report these ratios as measurements on this intervention path and claim no general ratio.

7 Implications for representation learning

For comparison methodology. Matching information is a real improvement over comparing native pipelines, and we continue to recommend it. But a declared statistic leaves a degree of freedom that can decide the answer, so it should be reported with the same care as the statistic itself. The realization audit is straightforward: for each architecture, list the native fields its interface consumes, mark each as reconstructed from ϕ or canonicalized, and publish the table. Where a ϕ-derived field is canonicalized for one family and retained for another, that asymmetry should be justified or removed. A claim of the form "A_1 beats A_2 at equal information" is incomplete as stated; it is a claim about $(\phi, \rho, \mathcal{T})$.

Applying the protocol elsewhere. The same degree of freedom exists wherever heterogeneous models consume different encodings of one signal. Comparing a patch-token transformer against a pixel CNN, for instance, one would declare ϕ as the image at a fixed resolution; realize it as pixels for the CNN and as patches for the transformer; and then audit not only that both realizations are lossless in ϕ, but that any ϕ-derived quantity one interface exposes explicitly, such as the patch's position in the positional embedding, is either exposed to the other or its absence declared. Event-versus-frame representations and alternative tokenizations admit the same treatment. We do not claim the ranking effects seen here recur in those settings; the protocol identifies the variable that must be controlled before an architectural attribution is made.

8 Scope and limitations

The estimand is conditional. What we identify is an architecture contrast conditional on $(\phi, \rho, \mathcal{T})$. It is not a representation-independent causal effect of architecture, and this paper is in part an argument that no such quantity is recoverable from an information-matched comparison alone. A different admissible realization class defines a different estimand, which is precisely why the class should be declared rather than left implicit.

The protocol is fixed, not optimized per architecture. Models are compared under a matched registered budget with no per-architecture tuning and no early stopping. We make no claim about an

ordering among independently optimized asymptotic solutions; the comparison is a fixed-compute one and is reported as such.

The evidence is bounded. Two automotive camera–LiDAR datasets give independent platform, sensor and environmental replication, which is cross-platform rather than cross-domain. The task is token-level correspondence rather than downstream detection or segmentation. The protocol already applies explicit coordinate supervision identically to every arm (Stage 0 regresses cell coordinates from each tower), and the realization effect persists after it; whether stronger positional supervision through a detection or segmentation head would remove it is untested here. The encoders are compact controlled representatives of their families, chosen so that the comparison is interpretable, rather than production-scale systems (Chen et al., 2023; Wu et al., 2024).

The mediator is not identified. We establish realization sensitivity causally, by intervention, but do not decompose its family-dependent magnitude into positional accessibility, inductive-bias alignment and optimization geometry. Separating those would require interventions on each in turn and is the natural next study.

A structural effect we identify but do not test. Height is not in O, so under both realizations the submanifold encoder's occupied voxels fall in a single height bin; the 3D graph is unchanged but degenerate in z, identically on both sides of the intervention.

What the intervention does not establish. The design covers both families on both datasets, and PointPillars shows no ranking reversal on either, because its ρ_{str} contrast is already non-negative on both datasets and Δ_ρ is positive. We also do not claim that ρ_{if} yields *the* correct ordering. We test two admissible policies rather than exhaust $\mathcal{R}_\phi$; other members of the class, such as coordinate-augmented raster inputs (Liu et al., 2018), would be further points in it.

9 Conclusion

We studied whether matching input information is sufficient to support architectural attribution. It is not. Across two information-equivalent interface realizations, PointPillars exhibits a resolved change in architectural contrast, while SparseConv3D changes winner; the latter ranking reversal independently reproduces on RADIATE and Boreas, as does the reversal of the SparseConv3D–PointPillars ordering.

The result exposes a second variable in controlled encoder comparison. The information statistic ϕ specifies what evidence is available, while the realization ρ determines how that evidence is exposed to the learner and therefore how it interacts with architectural inductive bias. Equal information need not induce equivalent learning problems.

To attribute a performance difference to architecture, control both the information available to the model and the interface through which that information is realized.

Reproducibility statement

Complete interface definitions, information and realization audits, per-seed statistics, direct realization contrasts, Boreas replication, robustness controls and run-level validity checks are provided in the supplementary material. Exact configurations, scoring code, canonical statistics scripts and per-seed result files are available in the anonymous repository at `https://anonymous.4open.science/r/equal-information-is-not-enough`. Every primary adjudication criterion was recorded, with a hash chain linking each amendment to its predecessor, before the corresponding result existed.

The use of large language models

Generative AI tools were used to support literature discovery, language editing, code debugging and checking, and feedback on experimental analysis. All implementations were independently checked and verified, and full responsibility is taken for the final work.

REFERENCES

Burnett, K., Yoon, D. J., Wu, Y., Li, A. Z., Zhang, H., Lu, S., Qian, J., Tseng, W.-K., Lambert, A., Leung, K. Y. K., Schoellig, A. P., and Barfoot, T. D. Boreas: A multi-season autonomous driving dataset. *IJRR*, 42(1–2):33–42, 2023.

Chen, Y., Li, Y., Zhang, X., Sun, J., and Jia, J. Focal sparse convolutional networks for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chen, Y., Liu, J., Zhang, X., Qi, X., and Jia, J. VoxelNeXt: Fully sparse VoxelNet for 3D object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Choy, C., Gwak, J., and Savarese, S. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Choy, C., Park, J., and Koltun, V. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Ethayarajh, K., Choi, Y., and Swayamdipta, S. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

Fan, L., Wang, F., Wang, N., and Zhang, Z. Fully sparse 3D object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Goyal, A., Law, H., Liu, B., Newell, A., and Deng, J. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning (ICML)*, 2021.

Graham, B., Engelcke, M., and van der Maaten, L. 3D semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018.

Islam, M. A., Jia, S., and Bruce, N. D. B. How much position information do convolutional neural networks encode? In *International Conference on Learning Representations (ICLR)*, 2020.

Kayhan, O. S. and van Gemert, J. C. On translation invariance in CNNs: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., and Beijbom, O. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Lenc, K. and Vedaldi, A. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., and Tang, Z. BEVFusion: A simple and robust LiDAR-camera fusion framework. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Liu, R., Lehman, J., Molino, P., Petroski Such, F., Frank, E., Sergeev, A., and Yosinski, J. An intriguing failing of convolutional neural networks and the CoordConv solution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Liu, Z., Hu, H., Cao, Y., Zhang, Z., and Tong, X. A closer look at local aggregation operators in point cloud analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Mahmoud, A., Hu, J. S. K., Kuai, T., Harakeh, A., Paull, L., and Waslander, S. L. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., and Ghanem, B. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sautier, C., Puy, G., Gidaris, S., Boulch, A., Bursuc, A., and Marlet, R. Image-to-LiDAR self-supervised distillation for autonomous driving data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Sheeny, M., De Pellegrin, E., Mukherjee, S., Ahrabian, A., Wang, S., and Wallace, A. RADIATE: A radar dataset for automotive perception in bad weather. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., and Li, H. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., and Han, S. Searching efficient 3D architectures with sparse point-voxel convolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., and Geiger, A. Sparsity invariant CNNs. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2017.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Wu, X., Jiang, L., Wang, P.-S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., and Zhao, H. Point Transformer V3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. A theory of usable information under computational constraints. In *International Conference on Learning Representations (ICLR)*, 2020.

Zhou, Y. and Tuzel, O. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.