PRISM: UNIFIED NEUTRINO EVENT RECONSTRUCTION VIA TASK-AWARE RETRIEVAL AND GEOMETRY-GROUNDED DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neutrino telescopes infer multiple latent physical quantities from the same sparse, irregular spatio-temporal pulse array, yet heterogeneous reconstruction targets depend on fundamentally distinct observational evidence. We present **PRISM**, a unified reconstruction framework that addresses this challenge through a three-level decoupling: shared event representation, task-conditioned information retrieval, and targeted structural specialization. PRISM employs a **Transport Skeletonizer (TSK)** to maintain a compact event-level latent memory while preserving fine-grained hit representations, and learns **task-aware queries** to independently extract target-dependent features from the combined event-level and hit-level representations. For interaction-vertex localization, a **geometry-aware coordinate anchor** grounds the decoder with an explicit sensor-coordinate reference prior to unconstrained residual refinement. Across the formal NuBench benchmark, PRISM offers a favorable accuracy–efficiency trade-off: it achieves competitive performance with published per-cell reference results in selected energy and topology regimes, while requiring approximately $5\times$ fewer parameters and providing 4.8–$5.0\times$ higher inference throughput than sequential multi-model execution. Rigorous equal-compute controls attribute a 32.65% reduction in median vertex error to coordinate anchoring, while ablations and controlled perturbation tests demonstrate task-dependent observation stability rather than universal robustness. Directional reconstruction, absolute vertex precision, and low-energy events remain key open challenges. Overall, our findings support a unified reconstruction strategy that shares expensive spatio-temporal representations, retrieves target-specific evidence, and introduces physically grounded specialization where uniform sharing is insufficient.

1 INTRODUCTION

Neutrino telescopes infer latent physical quantities from sparse, irregular spatio-temporal arrays of Cherenkov-light pulses recorded by distributed optical modules (Aartsen et al., 2017; Adrián-Martínez et al., 2016). A single event supports multiple reconstruction targets, including topology, energy, direction, interaction vertex, and visible inelasticity (Ørsøe et al., 2026). Although these targets describe the same underlying interaction, they rely on different observational evidence: energy depends strongly on distributed charge, topology on event morphology, direction on extended space–time structure, vertex localization on local geometry, and visible inelasticity on the partition of interaction energy.

A common strategy is to train a separate specialist for each target (Ørsøe et al., 2026), but this duplicates event encoding and inference whenever multiple quantities are reconstructed from the same event. Unified reconstruction instead encodes the event once and reuses the representation across tasks. However, the central challenge is not simply to attach multiple prediction heads to a shared encoder, but to determine what information should be shared, how each target should retrieve task-relevant evidence, and where explicit specialization is necessary. A pooled multi-task representation can discard local information before task-specific prediction, while additional task-specific capacity alone does not ensure access to the evidence required by each target.

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.

We introduce **PRISM**, a unified reconstruction framework organized around three levels of decoupling: separates shared event representation from task-aware retrieval, and targeted structural specialization (Fig. 1). The **Transport Skeletonizer (TSK)** maintains a compact event-level latent representation while preserving fine-grained hit memory, enabling global information exchange without collapsing the event into a single pooled feature. Learned **task-aware queries** then retrieve target-dependent evidence independently from the shared hit–skeleton memory. Finally, because interaction vertices are defined directly in detector coordinates, we equip the vertex branch with a **geometry-aware coordinate anchor** that provides an explicit spatial reference before unconstrained residual refinement.

Across the formal NuBench benchmark, PRISM exhibits a task-dependent accuracy–efficiency trade-off: it is most competitive in energy and topology while using a single shared representation for all five targets, with approximately $5\times$ fewer parameters and 4.8–$5.0\times$ higher compute-only throughput than sequential same-size model copies. Matched ablations support the roles of TSK and task-aware retrieval, and a strict equal-compute control isolates a 32.65% reduction in median vertex error from coordinate anchoring. Directional reconstruction and low-energy events remain the main limitations.

Our contributions are threefold:

- We introduce PRISM, a unified reconstruction architecture that separates shared event representation from task-conditioned retrieval and introduces targeted specialization only where task structure requires it.
- We develop the Transport Skeletonizer (TSK) and task-aware query decoders, which combine compact event-level communication with preserved hit-level memory and independent task-aware retrieval. Matched ablations validate both components under the same architecture and evaluation protocol.
- We introduce a geometry-aware coordinate anchor for vertex localization, grounding the decoder in observed sensor coordinates before residual refinement. A strict equal-compute control attributes a 32.65% reduction in median vertex error to coordinate anchoring.

2 Related Work

Neutrino event reconstruction. Deep-learning approaches to neutrino event reconstruction have used convolutional, sparse, graph-based, point-cloud, and attention-based architectures to infer physical quantities from irregular detector responses (Aiello et al., 2020; Søgaard et al., 2023; Yu et al., 2023). NuBench (Ørsøe et al., 2026) provides a common multi-geometry benchmark with representative models including ParticleNet (Qu & Gouskos, 2020), DynEdge (IceCube Collaboration, 2022), GRIT (Ørsøe et al., 2026), and DeepIce (Bukhari et al., 2024). Existing benchmark results are predominantly organized around task-specific reconstruction, whereas PRISM targets joint multi-quantity inference from a shared event representation.

Sparse point and latent-set representations. Sparse detector events naturally admit point- or set-based representations. PointNet (Qi et al., 2017a) established direct learning over unordered point sets, PointNet++ (Qi et al., 2017b) introduced hierarchical local aggregation, and DGCNN (Wang et al., 2019) dynamically constructs local graph neighborhoods for geometric feature learning. More recent transformer-based approaches include Point Transformer V3 (Wu et al., 2024), which provides efficient serialized local attention, and SPoTr (Park et al., 2023), which uses compact self-positioning points to mediate broader context. Related latent-set architectures such as Set Transformer (Lee et al., 2019) and Perceiver (Jaegle et al., 2021) similarly use compact latent representations for efficient global interaction. PRISM is related to these approaches but differs from latent-bottleneck designs in that its skeleton tokens do not replace the hit representation: TSK preserves the hit stream throughout the encoder, and downstream tasks retain joint access to both fine-grained hit memory and compact event-level tokens.

Task-conditioned readout and multi-task specialization. Query-conditioned readout has been used to extract structured outputs from shared representations, including Perceiver IO (Jaegle et al., 2022) and DETR (Carion et al., 2020). Multi-task methods address task interference at several levels. Optimization-based approaches use adaptive loss weighting (Chen et al., 2018; Kendall et al.,

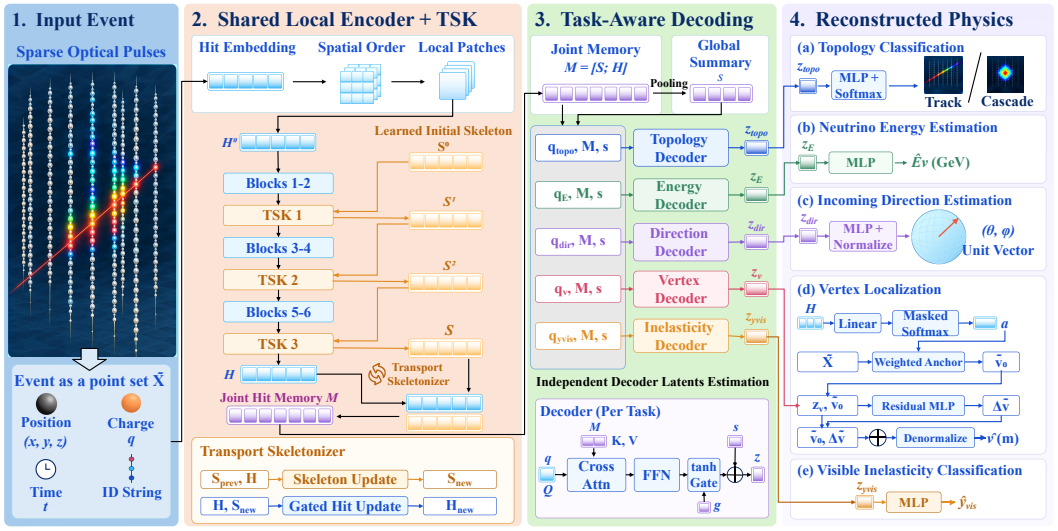

Figure 1: **Overview of PRISM.** A serialized local encoder constructs hit-level features; TSK mediates compact global exchange while preserving hit memory; task-aware queries retrieve target-specific representations from the joint memory; and the vertex branch adds a geometry-aware coordinate anchor before residual refinement.

2018) or gradient modification (Yu et al., 2020), while representation-sharing methods learn task-dependent combinations or attention over common features (Misra et al., 2016; Liu et al., 2019). Architectures such as MMoE (Ma et al., 2018) and PLE (Tang et al., 2020) further allocate partially shared and task-specific capacity. PRISM instead keeps a common event representation and specializes primarily at the retrieval stage: each task independently queries the same joint hit–skeleton memory, while additional task structure is introduced only when supported by the prediction target itself.

3 Method

Given a variable-length pulse set $\mathcal{X}$, PRISM constructs a shared hit–skeleton representation once and retrieves task-specific features for five reconstruction targets. The method consists of a local pulse encoder, the **Transport Skeletonizer (TSK)**, **task-aware retrieval**, and a **geometry-aware coordinate anchor** for vertex reconstruction (Fig. 1).

3.1 Event Representation and Local Encoding

We represent an event as an unordered pulse set

$$\mathcal{X} = \{(\mathbf{x}_i, t_i, q_i, s_i)\}_{i=1}^{n},$$

where $\mathbf{x}_i \in \mathbb{R}^3$ is the sensor position, t_i the arrival time, q_i the recorded charge, and s_i the detector-string identity. The five targets are $\mathcal{Y} = \{y_{\text{topo}}, E, \mathbf{d}, \mathbf{v}, y_{\text{vis}}\}$. After the NuBench feature normalization (Ørsøe et al., 2026) described in Appendix A.1, each pulse is embedded as

$$\mathbf{h}_i^0 = \phi_{\text{in}}(\mathbf{u}_i) + e(s_i).$$

Before global exchange, we encode the fine-grained pulse structure with a PTv3-style serialized local encoder (Wu et al., 2024). Pulses are Morton-ordered and partitioned into local patches of 64 hits, followed by six pre-normalized attention blocks with hidden dimension $d = 192$. Local attention includes a learned relative bias derived from spatial displacement, arrival-time difference, and charge difference; the exact relation features are given in Appendix A.4.

3.2 TRANSPORT SKELETONIZER

Local encoding preserves fine-grained pulse structure, but the reconstruction targets also require context across distant detector regions. Replacing the hit stream with a global pooled representation would provide compact context at the cost of local evidence. TSK instead maintains a persistent set of $K = 8$ skeleton tokens while retaining the full hit representation.

Let $\mathbf{H}^\ell \in \mathbb{R}^{n\times d}$ and $\mathbf{S}^{\ell-1} \in \mathbb{R}^{K\times d}$ denote the hit and skeleton representations at stage ℓ. Each TSK stage first updates the skeleton from the hits,

$$\begin{aligned}\widetilde{\mathbf{S}}^\ell &= \mathbf{S}^{\ell-1} + \mathrm{MHA}\left(\mathrm{LN}_S(\mathbf{S}^{\ell-1}), \mathrm{LN}_H(\mathbf{H}^\ell), \mathrm{LN}_H(\mathbf{H}^\ell)\right), \\ \mathbf{S}^\ell &= \widetilde{\mathbf{S}}^\ell + \mathrm{FFN}_S(\widetilde{\mathbf{S}}^\ell),\end{aligned} \tag{1}$$

and then transports the updated event context back to the hits,

$$\begin{aligned}\widehat{\mathbf{H}}^\ell &= \mathbf{H}^\ell + \tanh(\alpha_\ell)\mathrm{MHA}\left(\mathrm{LN}_H(\mathbf{H}^\ell), \mathrm{LN}_S(\mathbf{S}^\ell), \mathrm{LN}_S(\mathbf{S}^\ell)\right), \\ \mathbf{H}^{\ell+} &= \widehat{\mathbf{H}}^\ell + \tanh(\alpha_\ell)\mathrm{FFN}_H(\widehat{\mathbf{H}}^\ell).\end{aligned} \tag{2}$$

The first-stage skeleton consists of learned tokens, and subsequent TSK stages inherit the transformed skeleton state from the preceding stage. The scalar gate α_ℓ is initialized to zero, so TSK initially preserves the hit stream. Skeleton states persist across the three TSK stages inserted after encoder blocks 2, 4, and 6. Since $K \ll n$, hit–skeleton exchange scales as $\mathcal{O}(nKd)$ rather than the $\mathcal{O}(n^2d)$ cost of dense hit-level attention. Importantly, the skeleton does not replace the hits; both are retained for downstream retrieval.

3.3 TASK-AWARE RETRIEVAL

Sharing an event representation does not require all targets to consume the same summary. We therefore separate *representation sharing* from *task-aware retrieval*. The final hit and skeleton states form the joint memory

$$\mathbf{M} = [\mathbf{S}; \mathbf{H}], \qquad \mathbf{s} = \mathrm{MeanPool}(\mathbf{H}) + \tanh(g_S)\mathrm{Mean}(\mathbf{S}). \tag{3}$$

For each target τ, an independent learned query $\mathbf{q}_\tau$ retrieves information from the same memory:

$$\begin{aligned}\mathbf{a}_\tau &= \mathrm{MHA}_\tau\left(\mathrm{LN}_{q,\tau}(\mathbf{q}_\tau), \mathrm{LN}_{m,\tau}(\mathbf{M}), \mathrm{LN}_{m,\tau}(\mathbf{M})\right), \\ \mathbf{q}'_\tau &= \mathbf{q}_\tau + \mathbf{a}_\tau + \mathrm{FFN}_\tau(\mathbf{q}_\tau + \mathbf{a}_\tau).\end{aligned} \tag{4}$$

The retrieved representation modifies the shared summary through a learned residual gate,

$$\mathbf{z}_\tau = \mathbf{s} + \tanh(g_\tau)\mathbf{q}'_\tau.$$

Each task has independent normalization, cross-attention, and feed-forward parameters. With g_τ initialized to zero, every decoder starts from the shared representation and gradually learns task-aware retrieval. The joint memory allows each task to access both fine-grained hit evidence and compact event-level context.

3.4 GEOMETRY-AWARE COORDINATE ANCHOR

Task-aware retrieval provides target-specific latent features, but vertex localization has additional geometric structure: the prediction itself is defined in detector coordinates. A generic latent decoder does not explicitly ground its output in the observed sensor geometry. We therefore augment only the vertex branch with a learned coordinate reference followed by residual refinement.

Given the vertex representation $\mathbf{z}_v$ and final hit feature $\mathbf{h}_i$, we compute

$$e_i = \mathbf{w}_a^\top \mathbf{h}_i + b_a, \qquad a_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)},$$

where the softmax is taken over valid hits, and construct the normalized coordinate anchor

$$\widetilde{\mathbf{v}}_0 = \sum_i a_i \widetilde{\mathbf{x}}_i. \tag{5}$$

The final prediction is

$$\Delta\widetilde{\mathbf{v}} = g_\theta([\mathbf{z}_v; \widetilde{\mathbf{v}}_0]), \qquad \widehat{\widetilde{\mathbf{v}}} = \widetilde{\mathbf{v}}_0 + \Delta\widetilde{\mathbf{v}}.$$

The anchor is a permutation-invariant convex combination of observed sensor coordinates and serves as a geometric reference rather than the final estimate; the task-conditioned residual performs the final localization correction. This specialization is confined to the vertex branch.

3.5 Training and Inference

The five tasks are trained jointly using cross-entropy for topology, Smooth-L1 for energy and visible inelasticity, a cosine objective for direction, and coordinate-wise Smooth-L1 for vertex reconstruction, with weights $(1, 1, 1, 0.2, 0.2)$ for topology, energy, direction, vertex, and visible inelasticity, respectively. Coordinate Anchor refinement starts from the selected joint checkpoint and updates only the vertex pathway while freezing the shared encoder, TSK, and all non-vertex branches. During refinement, the joint vertex loss is replaced by the geometry-aligned objective defined in Appendix A.2. Full optimization settings are provided in Appendix A. At inference, the shared encoder and TSK are executed once for all five targets.

4 Experiments

4.1 NuBench Configurations and Evaluation Protocol

We evaluate PRISM on the seven NuBench configurations spanning six detector geometries (Ørsøe et al., 2026). We designate a configuration as *formal* when it follows the frozen training protocol, validation-based checkpoint selection, and complete official-test evaluation. Cluster, Flower-L, Flower-XL, and Hexagon satisfy this protocol and constitute the main evaluation set; Triangle, Flower-S, and Hexagon-Ice-LE are reported only as limited-budget breadth studies in the Appendix. Flower-XL additionally uses two optimization seeds for replication analyses.

We follow the official NuBench evaluation protocol (Ørsøe et al., 2026). The main text reports energy bias/resolution, topology ROC-AUC, directional opening-angle metrics, vertex localization error, and visible-inelasticity error; complete energy-binned and topology-specific metric definitions are provided in Appendix B.1.

4.2 Training and Comparison Setup

Formal models are trained jointly for three effective epochs using AdamW, with validation-based checkpoint selection and a maximum of 4096 pulses per event. The Coordinate Anchor uses the refinement protocol described in Sec. 3.4. Full optimization, augmentation, and checkpoint-selection details are provided in Appendix A.

We use three distinct comparison settings. **External accuracy** is compared with the published NuBench results for ParticleNet, DynEdge, GRIT, and DeepIce (Ørsøe et al., 2026); when summarizing across metric cells, we use the best published value in each cell as a reference envelope, which does not correspond to any single competing model. **Architectural contributions** are evaluated with matched internal ablations under the same training and official-test protocol. **Inference efficiency** is compared with sequential execution of five same-size model copies, which serves only as a storage/compute proxy and not as an accuracy comparison with independently optimized single-task specialists.

4.3 Accuracy–Efficiency Profile

Table 1 compares PRISM with published NuBench baselines (Ørsøe et al., 2026) on Flower-XL (a), summarizes reference coverage across four formal configurations (b), and reports inference efficiency (c). Panel (a) shows one metric family per task; complete official metrics appear in Appendix B.

The model-level comparison in Panel (a) shows that PRISM is strongest in energy and topology. Its energy resolution is lower than every listed published baseline in the intermediate- and high-energy regions, while its topology AUC is highest for the overall, low-energy, and intermediate-

Table 1: **Five-task reconstruction accuracy and the cost of unified inference.** **(a)** Flower-XL official-test results compared with named published NuBench baselines. Bold indicates the best available value in each row; the PRISM column is highlighted in blue, and ↑/↓ indicate preferred directions. **(b)** Cells reaching the published reference envelope across the four formal configurations, using all official metrics (not only those in (a)). **(c)** Parameter footprint and compute-only throughput against a sequential five-copy proxy; this is not a specialist-accuracy comparison.

(a) Flower-XL: absolute performance across five reconstruction targets

Metric	Regime	ParticleNet	DynEdge	GRIT	DeepIce	**PRISM**
Energy W (%) ↓	Low	**120.16**	203.08	122.63	–	123.25
	Mid	101.61	169.47	99.74	–	**69.87**
	High	90.11	99.10	90.42	–	**31.40**
Topology AUC ↑	Overall	0.8954	0.8890	0.9005	–	**0.9027**
	Low	0.6797	0.6780	0.6702	–	**0.6905**
	Mid	0.7969	0.7970	0.8108	–	**0.8295**
	High	0.9816	0.9796	**0.9840**	–	0.9836
Direction ψ (°) ↓	Low / CC	14.33	16.22	–	**14.23**	21.36
	Low / NC	**33.38**	33.47	–	33.60	37.71
	High / CC	1.60	2.03	–	**1.29**	3.58
	High / NC	3.81	4.07	–	**3.21**	9.02
Vertex D_{xyz} (m) ↓	Low	42.90	**11.98**	35.51	–	15.10
	High	36.82	**4.31**	30.00	–	7.69
Visible-y error ↓	Low	0.2490	**0.1810**	0.2467	–	0.1825
	Mid	0.1124	0.1399	**0.1088**	–	0.1356
	High	0.0398	0.0577	**0.0345**	–	0.0564

(b) Cross-configuration coverage of the published reference envelope

Configuration	Energy	Topology	Direction	Vertex	Visible-y	Total
Flower-XL	4/6	3/4	0/12	0/6	1/6	8/34
Flower-L	0/6	0/4	0/12	0/6	0/6	0/34
Hexagon	1/6	2/4	0/12	0/6	0/6	3/34
Cluster	1/6	0/4	0/12	0/6	0/6	1/34

(c) Inference efficiency: one shared pass versus five sequential copies

Metric	Five-copy proxy	Unified PRISM	Benefit
Parameters (M) ↓	39.225	**7.845**	80.0% fewer
Throughput, batch 1 (events/s) ↑	16.52	**82.19**	4.98×
Throughput, batch 32 (events/s) ↑	449.79	**2167.25**	4.82×

Energy uses percentile width $W = P_{84} - P_{16}$; direction and vertex use median errors, and visible-y uses median absolute error. For energy, topology and visible-y, Low/Mid/High denote 10–100 GeV, 100–1000 GeV and 1–100 TeV. For direction and vertex, Low/High denote $E \leq 1$ TeV/$E > 1$ TeV; CC/NC follow the official masks. A dash denotes an unavailable published result. The reference envelope in (b) takes the best published value per cell and is not a single model. Complete metrics are in Appendix B.

energy evaluations. In contrast, the best published results remain stronger for direction and vertex localization. Visible-inelasticity reconstruction is closer to the published results but does not show the same advantage as energy or topology.

Across the four configurations in Panel (b), PRISM reaches the published envelope in 12 of 136 cells: six energy, five topology, and one visible-inelasticity cell. Eight occur on Flower-XL, three on Hexagon, and one on Cluster; none occur on Flower-L or in direction and vertex. Competitiveness is therefore confined to selected tasks and configurations.

Panel (c) characterizes the corresponding efficiency trade-off. PRISM uses 7.845M parameters compared with 39.225M for sequential execution of five same-size model copies, corresponding to an 80.0% reduction in parameter count, while providing 4.82–4.98× higher compute-only throughput. This comparison quantifies the inference cost of shared versus repeated encoding and does not establish accuracy parity with five independently optimized task-specific specialists.

Table 2: **Matched architecture ablation on Flower-XL.** All variants use matched training budgets and the same official-test protocol for two optimization seeds. Higher is better for topology AUC; lower is better for all other metrics.

Variant	Seed	Topology AUC ↑	Energy $\lvert B\rvert$ (%) ↓	Energy width (%) ↓	Direction median (°) ↓	Vertex D_{xyz} (m) ↓	Visible-y median ↓	Visible-y width ↓
Base PRISM	42	**0.901**	**2.181**	**65.835**	**8.176**	**17.229**	**0.098**	**0.218**
Base PRISM	43	**0.894**	2.137	**66.763**	9.442	**19.976**	**0.105**	**0.220**
No task query	42	0.895	3.486	82.329	9.261	24.089	0.112	0.223
No task query	43	0.893	4.300	82.080	9.353	23.770	0.117	0.226
No TSK	42	0.881	2.255	81.574	10.866	27.089	0.114	0.226
No TSK	43	0.869	1.939	78.885	12.075	30.708	0.124	0.232
Pooled control	42	0.886	8.648	92.942	10.917	33.208	0.128	0.239
Pooled control	43	0.886	4.315	87.427	10.861	31.552	0.128	0.239

4.4 Shared Representation and Task-Aware Retrieval

We evaluate the two architectural mechanisms underlying PRISM—the Transport Skeletonizer and task-aware retrieval—using matched Flower-XL controls with two optimization seeds and the same official-test protocol. Table 2 compares Base PRISM with no-query, no-TSK, and pooled variants. All models use validation-selected checkpoints after three training epochs and are evaluated on the same 2,980,023-event official test set. The no-query variant removes task-aware retrieval, the no-TSK variant removes the skeleton pathway, and the pooled control serves as the shared pooled multi-task baseline.

Both components improve most headline metrics across the two seeds. Removing task-aware retrieval worsens energy resolution, direction, vertex localization, and visible-inelasticity metrics, together with smaller reductions in topology AUC. Removing TSK produces a similar degradation pattern, particularly for energy resolution, direction, vertex, and visible inelasticity. Energy bias is the main exception: the no-TSK model is slightly better for seed 43 but worse for seed 42. The pooled control shows the largest overall deterioration, supporting the benefit of structured shared memory and task-aware retrieval over naive pooled sharing.

These performance ablations are complemented by two diagnostics in the Appendix. Learned task queries exhibit stable task-dependent retrieval distributions over the shared hit–skeleton memory, consistent with differentiated information access across targets. A controlled perturbation study further shows lower clean-to-perturbed degradation with TSK for four of five task summaries, with direction as the explicit exception. Although TSK improves clean direction accuracy in Table 2, this perturbation result indicates that reconstruction accuracy and observation stability need not improve together. Overall, the evidence supports both structured shared memory and task-aware retrieval, while the effect of TSK remains task dependent rather than uniformly monotonic.

4.5 Geometry-Aware Vertex Specialization

We first isolate the effect of the Coordinate Anchor with a strict equal-compute Flower-XL control, and then examine whether the targeted improvement is reproduced across independent optimization seeds (Table 3). In the equal-compute setting, the Anchor is the only functional difference between the two refinement branches and reduces median D_{xyz} from 18.91 m to 12.74 m, a 32.65% reduction. This control therefore isolates the contribution of coordinate anchoring from additional vertex-only optimization.

The improvement is also reproduced in two Flower-XL optimization seeds under the full refinement setting, reducing median D_{xyz} by 44.22% and 44.83%, respectively. These rows provide within-seed replication of the targeted Anchor effect, but not equal-compute attribution because the total pre-refinement optimization histories are not identical. The larger gains in the full refinement runs should therefore be interpreted within their respective training histories.

To examine how this replicated improvement is distributed across events, Fig. 2 analyzes the seed-42 run. Coordinate anchoring shifts the vertex-error distribution toward smaller values, reducing the

Table 3: **Coordinate Anchor isolation and replication on Flower-XL.** D_{xyz} is the median three-dimensional vertex error in meters. Gain denotes the relative reduction in D_{xyz}. The equal-compute experiment isolates the Anchor under matched optimization; the remaining rows evaluate replication under the stated control setting.

Configuration	Comparison	Baseline	Anchor	Gain	Low-E gain	High-E gain	Max non-Vertex regression
Flower-XL	Equal-compute, 20k	18.911	12.738	**32.65%**	26.39%	37.94%	0%
Flower-XL	Seed 42	18.278	10.196	**44.22%**	34.34%	50.36%	1.71%
Flower-XL	Seed 43	18.927	10.443	**44.83%**	33.33%	51.96%	0%

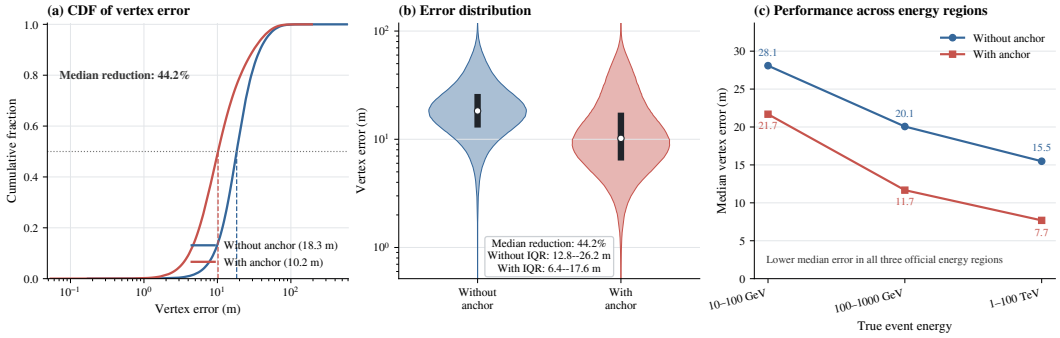

Figure 2: **Distribution of the Coordinate Anchor improvement on Flower-XL (seed 42). (a)** Cumulative distribution of the three-dimensional vertex error D_{xyz}. Coordinate anchoring reduces the median error from 18.3 m to 10.2 m. **(b)** Event-wise error distributions, showing reductions in both the median and interquartile range. **(c)** Median vertex error in the three official energy regions. The Anchor yields lower median error in all three regions. Equal-compute and cross-seed controls are reported separately in Table 3.

median D_{xyz} from 18.3 m to 10.2 m while also narrowing the interquartile range. The improvement is observed across all three official energy regions, although the absolute localization error remains largest at low energy. This distributional analysis complements, rather than replaces, the equal-compute isolation in Table 3.

Additional configuration-level results provide supporting evidence that the targeted vertex benefit extends beyond Flower-XL: Cluster shows formal effectiveness, while Hexagon and Flower-L require the provenance qualifications detailed in Appendix D.3. These results support broader effectiveness but do not establish geometry-independent improvement.

4.6 Anchor Mechanism Analysis

The learned Anchor provides a more informative geometric reference than simple sensor centroids, but it is not itself the final localization estimate. On Flower-XL, median D_{xyz} decreases from 64.18 m for the unweighted centroid and 59.95 m for the charge-weighted centroid to 45.60 m for the learned Anchor, before residual refinement reaches 12.74 m. Importantly, the Anchor alone remains worse than the learned no-Anchor baseline (18.91 m), confirming that its role is to provide a geometry-grounded reference rather than to solve localization directly. The same qualitative pattern is observed on Cluster, and residual refinement improves the Anchor estimate in more than 91% of events on both configurations. Full centroid and residual statistics are reported in Appendix D.4.

Figure 3 provides two illustrative event-level examples of the same reference–residual decomposition. In the extended track-like event, the charge centroid is strongly displaced from the interaction vertex, whereas the learned Anchor moves the reference substantially closer before residual correction. The more localized cascade-like event shows the same qualitative pattern at a smaller geometric scale.

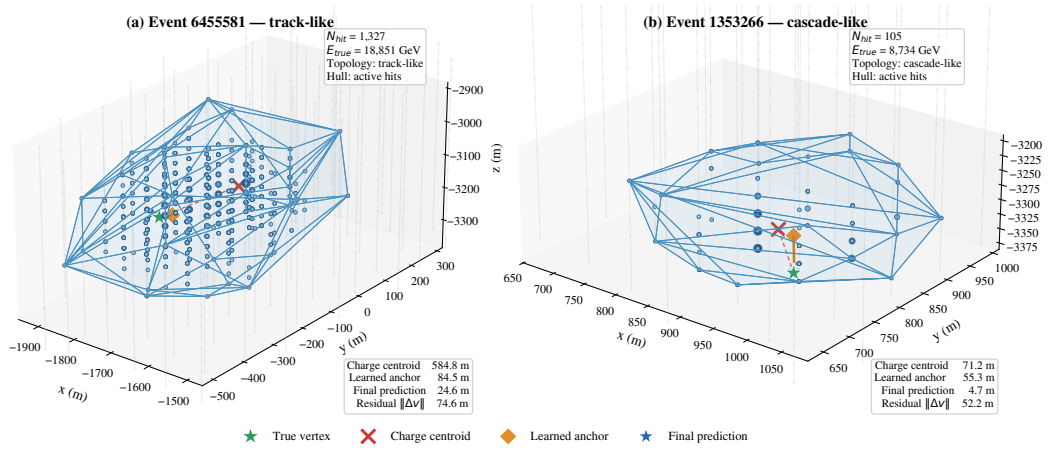

Figure 3: **Qualitative visualization of geometry-aware vertex refinement.** Two illustrative Flower-XL test events with **(a)** track-like and **(b)** cascade-like topologies are shown in detector coordinates. Blue markers denote activated sensors, and the translucent hull indicates their spatial support. We visualize the ground-truth interaction vertex, charge-weighted centroid, learned coordinate Anchor $\tilde{\mathbf{v}}_0$, and final prediction $\hat{\mathbf{v}} = \tilde{\mathbf{v}}_0 + \Delta\tilde{\mathbf{v}}$. The examples illustrate the intended decomposition of the vertex decoder: the Anchor provides a geometry-informed reference, while the learned residual displacement performs the final localization. These examples are qualitative and are not used for aggregate performance claims.

Taken together, the aggregate centroid comparison and the event-level visualizations support the intended role of the Coordinate Anchor: rather than directly solving the full localization problem, it grounds the prediction in the observed detector geometry, after which the residual decoder performs the final event-specific correction.

5 Conclusion

We presented PRISM, a unified five-target neutrino reconstruction framework that separates shared event representation, task-aware information retrieval, and targeted geometry-aware specialization. Matched ablations support both the Transport Skeletonizer and task-aware retrieval, while a strict equal-compute control isolates a 32.65% reduction in median vertex error from coordinate anchoring. Across NuBench, PRISM is competitive with published references in selected energy and topology regimes while using $5.00\times$ smaller footprint and achieving 4.8–$5.0\times$ higher compute-only throughput than sequential execution of five same-size model copies. Directional reconstruction, absolute vertex accuracy relative to published specialist references, and low-energy reconstruction remain the main open challenges.

Limitations. Performance remains uneven across tasks and regimes: direction is the clearest benchmark gap, and low-energy reconstruction remains difficult. Although the Coordinate Anchor substantially improves vertex localization under matched controls, its benefit is not uniform across configurations; the Hexagon-Ice-LE screen shows only a 5.83% gain, and the present experiments do not isolate whether this weaker effect is primarily associated with energy, hit multiplicity, or detector geometry. Because models are trained independently for each detector configuration, we do not evaluate held-out geometry generalization or robustness to realistic stochastic dark noise. Finally, the sequential five-copy baseline measures parameter and compute efficiency only and does not establish accuracy parity with independently optimized single-task specialists.

AI Use Statement

The authors acknowledge the use of large language models solely for grammar checking, code syntax verification, and stylistic polishing of the manuscript. The conceptual formulation, architecture

design, experimental execution, and analytical deductions were performed entirely by the authors without generative reliance.

REFERENCES

M. G. Aartsen et al. The IceCube neutrino observatory: Instrumentation and online systems. *Journal of Instrumentation*, 12(03):P03012, 2017. doi: 10.1088/1748-0221/12/03/P03012.

S. Adrián-Martínez et al. Letter of intent for KM3NeT 2.0. *Journal of Physics G: Nuclear and Particle Physics*, 43(8):084001, 2016. doi: 10.1088/0954-3899/43/8/084001.

S. Aiello et al. Event reconstruction for km3net/orca using convolutional neural networks. *Journal of Instrumentation*, 15(10):P10005, 2020. doi: 10.1088/1748-0221/15/10/P10005.

Habib Bukhari, Dipam Chakraborty, Philipp Eller, Takuya Ito, Maxim V. Shugaev, and Rasmus Ørsøe. Icecube–neutrinos in deep ice: The top 3 solutions from the public kaggle competition. *The European Physical Journal C*, 84(6):646, 2024. doi: 10.1140/epjc/s10052-024-12977-2.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, pp. 213–229. Springer, 2020.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 794–803. PMLR, 2018.

IceCube Collaboration. Graph neural networks for low-energy event classification & reconstruction in icecube. *Journal of Instrumentation*, 17(11):P11003, 2022. doi: 10.1088/1748-0221/17/11/P11003.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 2021.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018. doi: 10.1109/CVPR.2018.00781.

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 *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 2019.

Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1871–1880, 2019.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1930–1939, 2018. doi: 10.1145/3219819.3220007.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3994–4003, 2016. doi: 10.1109/CVPR.2016.433.

Rasmus Ørsøe, Stephan Meighen-Berger, Jeffrey Lazar, Jorge Prado, Iván Mozún-Mateo, Aske Rosted, Philip Weigel, and Arturo Llorente Anaya. NuBench: An open benchmark for deep learning-based event reconstruction in neutrino telescopes. *Journal of Instrumentation*, 21(05): T05001, 2026. doi: 10.1088/1748-0221/21/05/T05001.

Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunyang Xiong, and Hyunwoo J. Kim. Self-positioning point-based transformer for point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21814–21823, 2023.

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. doi: 10.1109/CVPR.2017.16.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017b.

Huilin Qu and Loukas Gouskos. ParticleNet: Jet tagging via particle clouds. *Physical Review D*, 101(5):056019, 2020. doi: 10.1103/PhysRevD.101.056019.

Andreas Søgaard, Rasmus F. Ørsøe, Morten Holm, Leon Bozianu, Aske Rosted, Troels C. Petersen, Kaare Endrup Iversen, Andreas Hermansen, Tim Guggenmos, Peter Andresen, Martin Ha Minh, Ludwig Neste, Moust Holmes, Axel Pontén, Kayla Leonard DeHolton, and Philipp Eller. Graphnet: Graph neural networks for neutrino telescope event reconstruction. *Journal of Open Source Software*, 8(85):4971, 2023. doi: 10.21105/joss.04971.

Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. Progressive layered extraction (PLE): A novel multi-task learning (MTL) model for personalized recommendations. In *Proceedings of the Fourteenth ACM Conference on Recommender Systems*, pp. 269–278, 2020. doi: 10.1145/3383313.3412236.

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*, 38(5):1–12, 2019. doi: 10.1145/3326362.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler Faster Stronger. In *Conference on Computer Vision and Pattern Recognition*, pp. 4840–4851, 2024. doi: 10.1109/CVPR52733.2024.00463. URL https://mlanthology.org/cvpr/2024/wu2024cvpr-point/.

Felix J. Yu, Jeffrey Lazar, and Carlos A. Argüelles. Trigger-level event reconstruction for neutrino telescopes using sparse submanifold convolutional neural networks. *Physical Review D*, 108(6): 063017, 2023. doi: 10.1103/PhysRevD.108.063017.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020.

A Implementation and Training Details

A.1 Input and coordinate normalization

Each pulse is represented by normalized spatial coordinates, time, and charge,

$$\tilde{\mathbf{x}} = (x/100,\, y/100,\, z/1000)\,, \qquad \tilde{t} = t/10^5, \qquad \tilde{q} = \log_{10}(q+1). \tag{6}$$

A learned embedding of the detector-string identity is added after projection of the continuous pulse features.

Vertex targets use the same anisotropic spatial normalization. Defining

$$\mathbf{A} = \mathrm{diag}\,(100\,\mathrm{m}, 100\,\mathrm{m}, 1000\,\mathrm{m})\,, \tag{7}$$

the normalized and physical coordinates are related by

$$\tilde{\mathbf{v}} = \mathbf{A}^{-1}\mathbf{v}, \qquad \hat{\mathbf{v}} = \mathbf{A}\hat{\tilde{\mathbf{v}}}. \tag{8}$$

The coordinate Anchor and its residual operate in this normalized coordinate system, whereas the vertex loss and all reported vertex metrics are evaluated after conversion back to physical units.

A.2 Training objectives

Topology classification uses cross-entropy. Energy and visible inelasticity use Smooth-L1 objectives, and direction is optimized with a cosine objective on the normalized direction vector.

The initial five-target stage uses coordinate-wise Smooth-L1 for vertex reconstruction. With

$$\mathbf{A} = \mathrm{diag}\left(100\,\mathrm{m}, 100\,\mathrm{m}, 1000\,\mathrm{m}\right),$$

the joint-training vertex objective is

$$\mathcal{L}_{\mathrm{vtx}}^{\mathrm{joint}} = \frac{1}{3B}\sum_{i=1}^{B}\sum_{c=1}^{3}\mathrm{SmoothL1}\left(\frac{\left[\mathbf{A}\left(\hat{\tilde{\mathbf{v}}}_i - \tilde{\mathbf{v}}_i\right)\right]_c}{100\,\mathrm{m}}\right). \tag{9}$$

The initial five-target model is therefore optimized with

$$\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_{\mathrm{topo}} + \mathcal{L}_E + \mathcal{L}_{\mathrm{dir}} + 0.2\mathcal{L}_{\mathrm{vtx}}^{\mathrm{joint}} + 0.2\mathcal{L}_{y_{\mathrm{vis}}}. \tag{10}$$

For the subsequent Coordinate Anchor refinement, we define the physical localization errors

$$D_{\mathrm{xyz},i} = \left\|\hat{\mathbf{v}}_i - \mathbf{v}_i\right\|_2, \tag{11}$$

$$D_{\mathrm{xy},i} = \left\|\hat{\mathbf{v}}_{i,xy} - \mathbf{v}_{i,xy}\right\|_2, \tag{12}$$

$$|D_{z,i}| = |\hat{v}_{i,z} - v_{i,z}|. \tag{13}$$

The vertex-only refinement objective is

$$\mathcal{L}_{\mathrm{vtx}}^{\mathrm{ref}} = \frac{1}{B}\sum_{i=1}^{B}\left[\frac{D_{\mathrm{xyz},i}}{100\,\mathrm{m}} + 0.1\frac{D_{\mathrm{xy},i}}{100\,\mathrm{m}} + 0.1\frac{|D_{z,i}|}{100\,\mathrm{m}}\right]. \tag{14}$$

$\mathcal{L}_{\mathrm{vtx}}^{\mathrm{ref}}$ replaces the joint-training vertex objective.

A.3 Optimization and checkpoint selection

The initial five-target model is trained with AdamW using a learning rate of 3×10^{-4} and weight decay 0.05. Gradients are clipped at 1.0 and automatic mixed precision is enabled. Formal scratch runs use cosine learning-rate decay with 3750 warm-up steps and a minimum learning-rate ratio of 0.01. Checkpoints are selected using validation loss.

Events are capped at 4096 pulses. When truncation is required, pulses are ranked deterministically by charge. Training uses a $1\,\mathrm{ns}$ time jitter and a $0.25\,\mathrm{p.e.}$ charge jitter; stochastic noise-pulse injection is disabled.

Coordinate Anchor refinement uses AdamW with a learning rate of 3×10^{-5}. The shared encoder, TSK, and all non-vertex branches are frozen. Trainable parameters are restricted to the vertex task-query row, vertex decoder/head, relevance head, and Anchor residual MLP. Anchor checkpoints are selected using the fixed validation composite score.

A.4 Local Relation Bias

Within each serialized local patch, we augment self-attention with a learned pairwise relative bias. For pulses i and j, we define the normalized relation feature

$$\mathbf{r}_{ij} = \left[\Delta\tilde{x}_{ij}, \Delta\tilde{y}_{ij}, \Delta\tilde{z}_{ij}, |\Delta\tilde{x}_{ij}|, |\Delta\tilde{y}_{ij}|, |\Delta\tilde{z}_{ij}|, \Delta\tilde{t}_{ij}, \Delta\tilde{q}_{ij}\right] \in \mathbb{R}^8, \tag{15}$$

where

$$\Delta\tilde{\mathbf{x}}_{ij} = \tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j, \qquad \Delta\tilde{t}_{ij} = \tilde{t}_i - \tilde{t}_j, \qquad \Delta\tilde{q}_{ij} = \tilde{q}_i - \tilde{q}_j. \tag{16}$$

Here, $\tilde{\mathbf{x}} = (x/100, y/100, z/1000)$, $\tilde{t} = t/10^5$, and $\tilde{q} = \log_{10}(q+1)$ follow the normalization defined in Appendix A.1. The three absolute spatial terms are computed element-wise rather than as a Euclidean distance. A learned relative-bias network maps $\mathbf{r}_{ij}$ to the attention bias used for each pulse pair within the local patch.

B Complete NuBench Evaluation

B.1 Task metrics

We follow the published NuBench evaluation protocol (Ørsøe et al., 2026). For energy reconstruction, the signed percentage residual is

$$R_E = 100\left(1 - \frac{E_{\mathrm{reco}}}{E_{\mathrm{true}}}\right), \tag{17}$$

so positive values denote underestimation and negative values denote overestimation. We report the median signed bias and the percentile width $P_{84} - P_{16}$ in 10–100 GeV, 100–1000 GeV, and 1–100 TeV.

Topology classification is evaluated by ROC-AUC overall and in the same three energy ranges. Direction reconstruction uses the median opening angle and the fractions of events reconstructed within 1° and 5°, separately for $E \leq 1$ TeV and $E > 1$ TeV and for CC/track and NC/cascade events.

Vertex reconstruction is evaluated with the median D_{xyz}, D_{xy}, and $|D_z|$, separately below and above 1 TeV. Visible inelasticity uses

$$R_y = |y_{\mathrm{reco}} - y_{\mathrm{vis}}|, \tag{18}$$

with its median and $P_{84} - P_{16}$ reported in the three energy ranges. Visible-inelasticity metrics are restricted to CC events.

B.2 Evaluation tiers and comparison convention

For completeness, we distinguish experimental results by training and provenance using the following tiers:

- **Formal**: frozen protocol, validation-selected checkpoint, complete official-test evaluation, and traceable result artifact.
- **Short adaptation**: limited-budget adaptation used as breadth evidence rather than as a formal benchmark result.
- **Screen**: controlled diagnostic or method-selection experiment.
- **Legacy**: experiment using a materially different architecture or protocol.
- **Non-comparable**: available evidence does not support the comparison under discussion.

Cluster, Flower-L, Flower-XL, and Hexagon constitute the formal configurations used in the main comparison. Triangle, Flower-S, and Hexagon-Ice-LE are reported only as short-adaptation breadth results.

The "published reference" used throughout the paper is the best published NuBench value in each metric cell. It is therefore a per-cell envelope over published models and does not correspond to a single competing model.

B.3 Continuous formal results and published baselines

Tables 4–8 provide the continuous values underlying the cell counts in Table 1. Every PRISM row is produced by one fixed, validation-selected checkpoint; values are never selected independently by cell. We reproduce the individual published NuBench baselines rather than presenting the per-cell envelope as if it were a single competing model. Abbreviations and metric directions are defined in the corresponding captions.

Table 4: **Formal energy-reconstruction comparison.** Signed median bias B and percentile width $W = P_{84} - P_{16}$ are percentages. Subscripts L, M, and H denote 10–100 GeV, 100–1000 GeV, and 1–100 TeV. Smaller $|B|$ and W are better.

Configuration	Model	B_{L}	W_{L}	B_{M}	W_{M}	B_{H}	W_{H}
Flower-XL	ParticleNeT	-8.92	120.16	18.18	101.61	3.78	90.11
	DynEdge	-70.08	203.08	-6.93	169.47	4.14	99.10
	GRIT	-9.37	122.63	21.15	99.74	9.33	90.42
	PRISM	28.81	123.25	1.83	69.87	-0.88	31.40
Flower-L	ParticleNeT	-159.37	279.99	-14.87	215.52	8.65	106.54
	DynEdge	-106.46	224.34	-8.03	189.24	3.13	107.41
	GRIT	-168.98	289.28	-13.81	212.77	6.77	108.47
	PRISM	189.56	369.62	73.47	293.05	27.94	176.34
Hexagon	ParticleNeT	-72.71	202.89	15.56	141.00	26.23	86.25
	DynEdge	-74.70	206.51	14.05	145.73	25.76	87.93
	GRIT	-85.85	226.39	6.04	156.10	32.44	94.11
	PRISM	85.27	219.28	-9.76	165.09	-21.64	97.45
Cluster	ParticleNeT	-48.44	145.94	11.91	151.16	15.84	90.17
	DynEdge	-46.42	145.23	12.57	148.77	15.66	90.30
	GRIT	-58.18	164.38	10.27	163.84	17.96	112.70
	PRISM	333.19	901.26	-2.86	175.46	-21.81	116.00

Table 5: **Formal direction-reconstruction comparison.** ψ is the median opening angle in degrees (lower is better); F_1 and F_5 are percentages reconstructed within 1° and 5° (higher is better). L/H denote $E \leq 1$ TeV/$E > 1$ TeV, and C/N denote CC/track and NC/cascade events.

Config.	Model	ψ_{LC}	ψ_{LN}	$F_{1,\mathrm{LC}}$	$F_{1,\mathrm{LN}}$	$F_{5,\mathrm{LC}}$	$F_{5,\mathrm{LN}}$	ψ_{HC}	ψ_{HN}	$F_{1,\mathrm{HC}}$	$F_{1,\mathrm{HN}}$	$F_{5,\mathrm{HC}}$	$F_{5,\mathrm{HN}}$
Flower-XL	ParticleNeT	14.33	33.38	1.56	0.30	21.75	6.54	1.60	3.81	27.23	6.74	91.05	61.80
	DynEdge	16.22	33.47	1.00	0.30	16.80	5.98	2.03	4.07	19.43	5.92	85.79	59.10
	DeepIce	14.23	33.60	2.12	0.39	24.11	7.01	1.29	3.21	36.65	9.98	93.23	67.15
	PRISM	21.36	37.71	0.43	0.12	9.05	2.68	3.58	9.02	7.08	1.12	67.20	22.43
Flower-L	ParticleNeT	13.21	20.66	1.26	0.60	20.45	11.54	2.12	3.50	18.36	8.27	83.66	64.89
	DynEdge	11.60	18.95	2.12	0.84	26.46	14.20	1.54	2.90	30.67	11.99	88.76	70.54
	GRIT	10.03	17.54	4.51	1.25	33.09	17.87	0.99	2.14	50.41	21.11	92.12	77.03
	DeepIce	9.92	17.19	5.81	1.32	33.98	18.47	0.66	1.86	65.13	28.14	92.87	78.44
	PRISM	15.09	22.86	0.90	0.41	16.46	8.69	3.77	5.74	5.67	2.64	65.38	42.74
Hexagon	ParticleNeT	16.91	32.44	1.35	0.45	19.80	8.38	2.03	3.86	19.93	8.10	82.15	59.55
	DynEdge	16.61	31.92	1.50	0.49	20.89	8.94	1.76	3.48	25.62	10.25	83.92	62.12
	GRIT	16.52	32.34	1.79	0.54	22.17	9.27	1.83	3.42	23.33	10.03	84.16	63.15
	DeepIce	15.02	30.73	3.44	0.82	27.15	11.39	0.89	2.53	54.46	20.08	88.23	68.74
	PRISM	20.12	33.67	0.56	0.26	11.15	5.50	3.11	5.58	10.27	3.85	69.95	45.51
Cluster	ParticleNeT	8.29	14.55	2.58	1.30	32.69	19.83	2.06	2.47	17.83	14.08	86.93	78.43
	DynEdge	7.66	13.82	3.25	1.70	36.03	22.12	1.88	2.12	21.11	18.67	88.02	81.07
	GRIT	7.66	13.74	2.88	1.48	35.14	21.62	2.18	2.60	17.21	13.64	84.45	77.34
	DeepIce	6.25	12.15	7.83	3.00	43.79	27.16	1.22	1.35	40.45	37.53	92.00	86.23
	PRISM	19.47	30.67	0.35	0.16	7.92	4.03	6.49	8.07	1.71	1.25	34.50	25.61

Table 6: **Formal topology-classification comparison.** ROC-AUC is reported overall and in the three energy ranges; higher is better.

Configuration	Model	Overall	10–100 GeV	100–1000 GeV	1–100 TeV
Flower-XL	ParticleNeT	0.8954	0.6797	0.7969	0.9816
	DynEdge	0.8890	0.6780	0.7970	0.9796
	GRIT	0.9005	0.6702	0.8108	0.9840
	PRISM	0.9027	0.6905	0.8295	0.9836
Flower-L	ParticleNeT	0.9308	0.5371	0.8012	0.9860
	DynEdge	0.9318	0.5327	0.8136	0.9859
	GRIT	0.9462	0.5462	0.8512	0.9909
	PRISM	0.9374	0.5021	0.8234	0.9885
Hexagon	ParticleNeT	0.9177	0.6560	0.7898	0.9786
	DynEdge	0.9340	0.6719	0.8273	0.9853
	GRIT	0.9219	0.6444	0.7980	0.9801
	PRISM	0.9354	0.6759	0.8131	0.9843
Cluster	ParticleNeT	0.9242	0.7536	0.8849	0.9908
	DynEdge	0.9223	0.7540	0.8852	0.9891
	GRIT	0.9014	0.7257	0.8360	0.9856
	PRISM	0.8456	0.5598	0.7446	0.9690

Table 7: **Formal vertex-reconstruction comparison.** Median errors are in meters and lower is better. L/H denote $E \leq 1$ TeV and $E > 1$ TeV.

Configuration	Model	$D_{\mathrm{xyz,L}}$	$\lvert D_{z,\mathrm{L}}\rvert$	$D_{\mathrm{xy,L}}$	$D_{\mathrm{xyz,H}}$	$\lvert D_{z,\mathrm{H}}\rvert$	$D_{\mathrm{xy,H}}$
Flower-XL	ParticleNeT	42.90	16.13	36.77	36.82	16.55	30.54
	DynEdge	11.98	3.04	10.06	4.31	1.80	3.35
	GRIT	35.51	7.68	32.86	30.00	6.91	28.09
	PRISM	15.10	3.87	13.00	7.69	2.76	6.27
Flower-L	ParticleNeT	17.47	6.15	14.09	10.20	4.36	7.84
	DynEdge	6.52	1.96	5.28	2.20	0.88	1.75
	GRIT	13.59	6.50	9.69	10.08	6.01	6.46
	PRISM	8.60	2.47	7.05	3.81	1.46	3.06
Hexagon	ParticleNeT	25.46	6.96	22.00	13.99	5.08	11.45
	DynEdge	12.02	2.09	10.33	2.37	0.87	1.93
	GRIT	41.30	26.43	25.72	38.20	28.14	22.06
	PRISM	15.83	2.95	13.84	5.38	1.71	4.60
Cluster	ParticleNeT	4.30	1.84	3.16	2.25	1.45	1.15
	DynEdge	2.97	0.93	2.52	1.29	0.44	1.09
	GRIT	16.43	5.04	14.23	13.06	4.31	10.89
	PRISM	13.13	3.73	11.30	4.84	1.51	4.10

Table 8: **Formal visible-inelasticity comparison on CC events.** Median absolute error M and percentile width $W = P_{84} - P_{16}$ are reported in the three energy ranges; lower is better.

Configuration	Model	M_{L}	W_{L}	M_{M}	W_{M}	M_{H}	W_{H}
Flower-XL	ParticleNeT	0.2490	0.2981	0.1124	0.1904	0.0398	0.0958
	DynEdge	0.1810	0.2370	0.1399	0.2423	0.0577	0.1348
	GRIT	0.2467	0.2932	0.1088	0.1843	0.0345	0.0836
	PRISM	0.1825	0.2369	0.1356	0.2474	0.0564	0.1244
Flower-L	ParticleNeT	0.5185	0.4921	0.1270	0.2239	0.0436	0.1053
	DynEdge	0.1812	0.2405	0.1107	0.2065	0.0435	0.1071
	GRIT	0.1821	0.2352	0.1231	0.2197	0.0464	0.1124
	PRISM	0.1928	0.2553	0.1164	0.2472	0.0559	0.1236
Hexagon	ParticleNeT	0.1861	0.2441	0.1193	0.2178	0.0446	0.1145
	DynEdge	0.1851	0.2438	0.1182	0.2171	0.0454	0.1167
	GRIT	0.4912	0.4840	0.1554	0.2798	0.0545	0.1310
	PRISM	0.1906	0.2463	0.1247	0.2443	0.0526	0.1248
Cluster	ParticleNeT	0.3591	0.4713	0.0770	0.1193	0.0345	0.0858
	DynEdge	0.1712	0.2568	0.0775	0.1468	0.0369	0.0908
	GRIT	0.3962	0.4659	0.1034	0.1512	0.0469	0.1123
	PRISM	0.2400	0.3498	0.1502	0.2924	0.0809	0.1659

Table 9: **Aggregate Full versus no-query ablation.** Entries report the number of optimization seeds for which the full task-query decoder performs better than the no-query variant.

Metric family	Full wins / 3 seeds
Topology AUC	3/3
Energy bias	3/3
Energy width	3/3
Direction	3/3
Vertex	3/3
Visible-y median error	2/3
Visible-y width	2/3

Table 10: **Task-query retrieval statistics** on 12,000 deterministic Flower-XL validation events. Skeleton and hit masses sum to one for each task.

Task	Skeleton mass	Hit mass	Entropy	Spread (m)
Topology	0.3034	0.6966	0.9828	105.84
Energy	0.4002	0.5998	0.9572	108.94
Direction	0.3207	0.6793	0.8769	115.25
Vertex	0.4620	0.5380	0.8707	99.64
Visible-y	0.5382	0.4618	0.8741	100.40

C Additional TSK and Task-Query Analysis

This section provides additional ablation and diagnostic results for the two components responsible for shared event representation and task-specific retrieval. The aggregate multi-seed ablations are retained as supporting evidence, while the controlled perturbation and retrieval analyses provide the directly verified mechanistic evidence used in the main text.

C.1 Task-query performance ablation

In addition to the current-architecture two-seed ablation reported in the main paper, we retain an earlier three-seed aggregate comparison between the full task-aware decoder and the corresponding no-query variant as secondary supporting evidence.

The corresponding per-seed prediction artifacts and complete checkpoint history are not fully preserved under the current standardized evaluation scope. We therefore treat Table 9 as supporting aggregate evidence rather than as an independently verified numerical replication study. In particular, we do not use these aggregate counts to claim formal multi-seed significance.

C.2 Task-query retrieval statistics

We further analyze how the five learned task queries retrieve information from the same shared hit–skeleton memory. Statistics are computed on a fixed set of 12,000 deterministic Flower-XL validation events using the seed-42 Anchor checkpoint. For each query, we measure the total attention mass assigned to skeleton and hit tokens, the normalized attention entropy, and the spatial spread of the attended detector observations.

The retrieval distributions differ systematically across tasks. Skeleton attention mass ranges from 0.3034 for topology to 0.5382 for visible inelasticity, indicating that different targets use the compact event-level representation to different degrees. Pairwise attention-profile similarities range from 0.6597 to 0.9125, with a mean of 0.8125. The least similar pair is topology–direction (0.6597), whereas energy–visible-y is the most similar pair (0.9125).

To test whether these differences are stable properties of the learned decoder rather than sampling fluctuations, we divide the event set into two disjoint halves and repeat the analysis. The resulting split-half correlations are 0.99995 for memory mass, 0.99999 for attention entropy, 0.99941 for spatial spread, and 0.99995 for pairwise similarity.

Table 11: **Aggregate Full versus no-TSK ablation.** Entries report the number of seeds for which the full TSK model performs better than the no-TSK variant.

Metric family	Full wins / 3 seeds
Topology AUC	1/3
Energy bias	3/3
Energy width	3/3
Direction	2/3
Vertex	3/3
Visible-y median error	2/3
Visible-y width	3/3

Table 12: **Clean-to-perturbed degradation** on the fixed 4096-event Cluster validation set. Lower degradation is better.

Task	Full TSK	no-TSK	Better
Topology	-0.07%	$+0.37\%$	Full
Energy	$+1.40\%$	$+4.10\%$	Full
Direction	$+7.76\%$	$+5.80\%$	no-TSK
Vertex	$+4.02\%$	$+4.83\%$	Full
Visible-y	$+0.15\%$	$+6.29\%$	Full

These results provide descriptive evidence that different task queries learn stable retrieval distributions within the evaluated checkpoint and event sample from the same shared representation. They should not be interpreted as causal physical explanations of individual attention weights, nor as evidence that each query isolates a unique physical interaction process.

C.3 TSK PERFORMANCE ABLATION

The aggregate three-seed Full versus no-TSK comparison is summarized in Table 11. The clearest aggregate gains occur for the energy and vertex metrics, while topology is less consistent across seeds.

As for the no-query comparison, the raw per-seed prediction artifacts are not fully retained under the standardized benchmark protocol. We therefore report these counts as supporting evidence and do not use them as the primary basis for the TSK claim in the main paper.

C.4 CONTROLLED OBSERVATION STABILITY

The primary audited TSK diagnostic uses the same fixed 4096-event Cluster validation set for the full and no-TSK models. We compare the relative clean-to-perturbed degradation for each reconstruction target.

This is an inference-only diagnostic, not a noise-background benchmark. The fixed events and perturbations are deterministic by event ID. The registered suite consists of 10/20/30% hit thinning, 10/30% string dropout, 10/30% peripheral cropping, 1/5 ns time smearing, and 0.25/0.50 p.e. charge jitter. For the cell-level analysis, drift is the eventwise change in the task output or task error induced by one condition; lower mean drift is preferred. We use six drift measures (topology probability and flip rate; energy log-error; direction angle; vertex distance; and CC inelasticity error), yielding 11 conditions $\times$ 6 measures = 66 cells. Full versus no-TSK differences use paired bootstrap 95% intervals with 1,000 resamples; an interval wholly below (above) zero is reported as lower (higher) drift, otherwise as unresolved. This definition accommodates metrics with different optimization directions without turning the diagnostic into a common physical-error scale.

The full model exhibits smaller degradation for four of the five task summaries, with direction as the explicit exception. Across the separately defined cell-level drift analysis, the full model has lower drift in 44 cells, higher drift in 3, with 19 intervals unresolved. The aggregate degradation summary and the cell-level drift analysis have different units and should not be conflated.

Table 13: **Equal-compute Coordinate-Anchor isolation on Flower-XL.** D_{xyz}, $|D_z|$, and D_{xy} are in meters. Gain denotes the relative reduction from the matched no-Anchor control.

Step	Model	Overall D_{xyz}	Low-E D_{xyz}	High-E D_{xyz}	Overall $\lvert D_z \rvert$	Overall D_{xy}
10k	No Anchor	18.889	23.416	16.206	10.328	12.821
10k	Anchor	13.545	18.310	10.816	4.639	11.170
10k	Gain	**28.29%**	**21.80%**	**33.26%**	**55.08%**	**12.87%**
20k	No Anchor	18.911	23.600	16.224	10.383	12.921
20k	Anchor	12.738	17.371	10.068	4.247	10.574
20k	Gain	**32.65%**	**26.39%**	**37.94%**	**59.09%**	**18.17%**

Table 14: **Flower-XL seed replication of Coordinate Anchor refinement.** Errors are median D_{xyz} values in meters.

Seed	Baseline	Anchor	Overall gain	Low-E gain	High-E gain	$\lvert D_z \rvert$ gain	D_{xy} gain	Max non-Vertex regression
42	18.278	10.196	**44.22%**	34.34%	50.36%	66.12%	33.57%	1.71%
43	18.927	10.443	**44.83%**	33.33%	51.96%	66.26%	34.73%	0%

We therefore interpret TSK as providing *task-dependent observation stability*. The experiment does not establish detector invariance, universal robustness, or uniform gains across all reconstruction targets.

D Extended Coordinate-Anchor Analysis

The main paper reports the headline Coordinate-Anchor results. Here, we provide the controlled isolation experiment, detailed replication evidence, and a mechanistic comparison with simple geometry-based references.

D.1 Equal-compute Anchor isolation

To determine whether the observed vertex gain is caused by coordinate anchoring rather than simply by additional vertex-only optimization, we construct a strict matched control on Flower-XL. Both branches start from the same validation-selected checkpoint and use identical event IDs, optimization seed, batch size, 4096-hit cap, augmentation, optimizer, learning-rate schedule, trainable vertex parameters, and total event exposure. The shared encoder, TSK, and all four non-vertex branches are frozen in both cases. The coordinate Anchor is therefore the only functional difference between the two refinement runs.

At 20k refinement steps, the Anchor reduces the overall median D_{xyz} by 32.65%, with improvements in both energy regions. Because all non-vertex branches are frozen and receive the same shared representation, their predictions are identical by construction in this experiment. This provides controlled evidence that the gain is not explained merely by additional vertex-only optimization time.

The equal-compute experiment isolates the contribution of the Anchor component itself. It does not imply that all gains observed in the longer formal refinement runs are attributable exclusively to the Anchor, since those runs use longer adaptation histories.

D.2 Flower-XL seed replication

The full Coordinate Anchor refinement is independently evaluated for two Flower-XL optimization seeds.

The improvement is consistent across the two optimization seeds. Seed 42 includes an additional documented low-learning-rate continuation before the Anchor stage relative to seed 43. The appropriate interpretation is therefore within-seed replication of the targeted Anchor effect, rather than identical total pretraining exposure across seeds.

Table 15: **Cross-configuration Coordinate-Anchor evidence.** The final column records the comparison scope required by the checkpoint history.

Configuration	Baseline role	Baseline	Anchor	Gain	Low-E	High-E	Max non-Vertex regression	Interpretation
Flower-XL s42	Converged independent	18.278	10.196	**44.22%**	34.34%	50.36%	1.71%	Formal pass
Flower-XL s43	Converged independent	18.927	10.443	**44.83%**	33.33%	51.96%	0%	Formal pass
Cluster	Independently trained baseline	43.884	8.045	**81.67%**	81.08%	81.11%	0%	Formal effectiveness
Hexagon	Referenced matched baseline	14.142	7.396	**47.70%**	28.49%	52.76%	0.0016%	Formal, qualified provenance
Flower-L	Fixed-start no-Anchor view	66.435	4.859	**92.69%**	86.94%	94.29%	0%	Supporting evidence

Table 16: **Anchor mechanism analysis.** All errors are in meters. The learned Anchor is evaluated before residual correction.

Dataset	Predictor	D_{xyz}	$\lvert D_z \rvert$	D_{xy}
Flower-XL	Learned baseline	18.91	10.38	12.92
Flower-XL	Mean centroid	64.18	28.89	50.76
Flower-XL	Charge centroid	59.95	25.81	49.03
Flower-XL	Learned Anchor $\tilde{\mathbf{v}}_0$	45.60	20.13	35.44
Flower-XL	Final $\tilde{\mathbf{v}}_0 + \Delta\tilde{\mathbf{v}}$	**12.74**	**4.25**	**10.57**
Cluster	Learned baseline	42.44	37.35	14.51
Cluster	Mean centroid	40.96	19.93	32.10
Cluster	Charge centroid	34.10	16.03	27.33
Cluster	Learned Anchor $\tilde{\mathbf{v}}_0$	27.47	11.28	22.67
Cluster	Final $\tilde{\mathbf{v}}_0 + \Delta\tilde{\mathbf{v}}$	**7.71**	**2.25**	**6.58**

D.3 Cross-configuration effectiveness

The Anchor is also evaluated on the additional formal configurations used in the main paper. The comparison roles differ across configurations and are therefore explicitly stated.

Cluster provides formal cross-configuration effectiveness, although the total optimization exposure differs between the baseline and final refinement. Hexagon is also a formal result, but the complete baseline parameter-level history is outside the present comparative focus. Flower-L uses a registered fixed-start no-Anchor view rather than a separately converged independent baseline and is therefore treated only as supporting evidence.

Accordingly, these experiments support cross-configuration effectiveness of the targeted vertex specialization, but they do not establish universal geometry-independent improvement or state-of-the-art vertex reconstruction.

D.4 What the Anchor provides

The coordinate Anchor is designed to provide an explicit spatial reference before residual refinement. To distinguish this mechanism from simple centroid localization, we compare the learned Anchor with unweighted and charge-weighted sensor centroids.

The learned Anchor improves over both unweighted and charge-weighted centroids on Flower-XL and Cluster. However, the Anchor alone should not be interpreted as an accurate final vertex estimator. On Flower-XL, for example, its D_{xyz} remains worse than the learned baseline before residual refinement.

The residual branch performs most of the final localization correction. Its median norm is 40.80 m on Flower-XL and 26.35 m on Cluster, and it improves the learned Anchor estimate in 93.63% and 91.03% of events, respectively. The corresponding fractions in which residual refinement worsens the Anchor estimate are 6.37% and 8.97%.

The supported mechanism-level interpretation is therefore that the learned Anchor provides a more informative geometry-based reference than simple centroids, while vertex residual refinement performs the majority of the final correction.

Table 17: **Controlled alternatives not retained in the final PRISM architecture.** These experiments are diagnostic and are not treated as primary benchmark results.

Intervention	Observation	Interpretation
Generic spatial adapter	Direction and vertex performance deteriorate substantially.	Generic early task branching is not sufficient; specialization must be aligned with the structure of the target.
XYZ-MSE direction objective	Direction opening angle changes from 54.67° to 56.31°, while energy width also deteriorates.	Direct Cartesian MSE does not resolve the direction-reconstruction limitation.
Time/Fourier direction features	No promotion-level direction improvement is observed.	Additional temporal encoding alone is insufficient to close the direction gap.
Remove-direction / four-task variants	Strict promotion criteria are not satisfied. Energy-only variants improve low-energy bias and width but introduce degradation in other energy regions.	The weak direction task is not shown to be the sole cause of broad negative transfer in the unified model.
Low-energy reweighting	Low-energy energy metrics improve, but middle/high-energy or pooled trade-offs remain.	The low-energy gap cannot be removed without introducing performance trade-offs under the tested weighting schemes.
Hexagon-Ice-LE Anchor	Vertex improvement is limited to 5.83%.	Anchor effectiveness is not universally strong in low-energy or low-hit regimes.

E Negative Results, Limitations, and Evidence Provenance

The final PRISM architecture is the result of several controlled comparisons, including alternatives that did not provide sufficiently stable improvements. We retain these negative results because they constrain the interpretation of the successful components and help distinguish targeted structured specialization from generic increases in task-specific capacity.

E.1 Controlled alternatives not retained

These experiments motivate the final design principle used in PRISM: task-specific capacity is introduced only when it is associated with a clear target structure and supported by controlled evidence.

E.2 Scope of the five-copy efficiency comparison

The main paper compares one unified PRISM forward pass with sequential execution of five same-size model copies. This experiment measures the storage and inference-compute consequences of using five separate networks instead of one shared representation.

No complete final-architecture suite of five independently optimized single-task specialists is available. The retained legacy single-task experiments cover only energy, visible inelasticity, and vertex; direction and topology specialists are missing. In addition, these legacy experiments use an earlier decoding configuration and do not incorporate the final Coordinate-Anchor vertex head.

Consequently, the five-copy comparison is an efficiency proxy only. It supports the parameter and throughput comparison reported in the main paper, but it does not support a claim that PRISM matches the reconstruction accuracy of five independently optimized specialists.

E.3 Evidence provenance and comparison scope

Several result groups in the paper differ in the strength of their experimental provenance. We therefore distinguish their intended use explicitly.

E.4 Remaining scope limitations and open questions

Beyond the controlled negative interventions in Table 17, several architectural and evaluation limitations further constrain the scope of the present results.

Table 18: **Evidence provenance for the main experimental claims.**

Evidence	Status	Appropriate interpretation
Unified efficiency	Directly verified	Parameter and compute-only throughput comparison against sequential execution of five same-size model copies.
Current-architecture task-query ablation	Matched two-seed official-test control	Primary evidence that task-aware retrieval improves most reported headline metrics relative to the no-query variant, with the stated seed-specific exceptions.
Current-architecture TSK ablation	Matched two-seed official-test control	Primary evidence that TSK improves most reported headline metrics relative to the no-TSK variant, with energy bias as the main seed-dependent exception.
Pooled multi-task control	Matched two-seed official-test control	Evidence that structured shared memory and task-aware retrieval outperform a naive pooled multi-task baseline under the matched setting.
TSK perturbation stability	Directly verified	Task-dependent observation stability; direction is the explicit exception.
Task-query retrieval analysis	Directly verified descriptive diagnostic	Stable task-dependent retrieval distributions within the evaluated checkpoint; not causal physical semantics.
Historical three-seed query/no-query aggregate	Aggregate supporting record	Secondary supporting evidence only; complete per-seed numerical artifacts are not preserved under the current standardized benchmark scope.
Historical three-seed TSK/no-TSK aggregate	Aggregate supporting record	Secondary supporting evidence with the same provenance limitation.
Flower-XL equal-compute Anchor	Matched controlled comparison	Strong evidence that coordinate anchoring contributes beyond additional vertex-only optimization.
Flower-XL two-seed Anchor replication	Within-seed replication	Targeted Anchor improvement is reproduced in two optimization seeds, with the training-history qualification described in Appendix D.2.
Cluster Anchor	Formal effectiveness	Cross-configuration effectiveness with unequal total optimization exposure.
Hexagon Anchor	Formal, qualified provenance	Cross-configuration effectiveness; historical baseline checkpoints are not preserved for per-step inspection.
Flower-L Anchor	Fixed-start supporting evidence	Supporting evidence only; not a comparison against a separately converged no-Anchor baseline.

Direction reconstruction. Direction remains the largest formal benchmark gap: PRISM reaches none of the 48 published-reference direction cells in Table 1. Several tested interventions, including an alternative Cartesian direction objective and additional temporal features, do not provide promotion-level improvements (Table 17). These experiments establish the persistence of the direction gap but do not isolate its architectural cause.

The current pipeline suggests several hypotheses for future isolation studies. Long events are deterministically truncated by retaining high-charge pulses, Morton serialization is defined from spatial coordinates, and global context is mediated through a compact set of $K = 8$ skeleton tokens. Whether these choices disproportionately affect timing-sensitive directional evidence remains unresolved; the present experiments do not support attributing the direction gap to any one of them.

Geometry-aware vertex specialization. The learned coordinate reference

$$\tilde{\mathbf{v}}_0 = \sum_i a_i \tilde{\mathbf{x}}_i, \qquad a_i \geq 0, \qquad \sum_i a_i = 1, \tag{19}$$

is a convex combination of the observed sensor coordinates and therefore lies within their convex hull. This constraint applies only to the reference $\tilde{\mathbf{v}}_0$: the final prediction $\hat{\tilde{\mathbf{v}}} = \tilde{\mathbf{v}}_0 + \Delta\tilde{\mathbf{v}}$ contains an unconstrained residual displacement.

Consistent with this role, the residual branch performs most of the final localization correction. Its median physical displacement norm is 40.80 m on Flower-XL and 26.35 m on Cluster, and it improves the learned reference in 93.63% and 91.03% of events, respectively (Section D.4). The Anchor gains are also anisotropic: for example, the Flower-XL equal-compute control reduces $|D_z|$ by 59.09% and D_{xy} by 18.17%. The current experiments do not isolate whether this difference originates from detector geometry, the optimization objective, or other factors.

Anchor effectiveness is not uniform across all tested regimes. In the Hexagon-Ice-LE short/screen setting, the observed vertex improvement is only 5.83% (Table 17). Because this experiment does not separately control hit multiplicity, energy, and detector geometry, we treat it as evidence of limited transfer in that setting rather than as a causal diagnosis of why the Anchor is less effective.

Energy reconstruction and optimization trade-offs. Energy performance is also regime dependent. PRISM is most competitive in the intermediate- and high-energy ranges, while the lowest-energy range remains more challenging. Under the tested weighting schemes, low-energy reweighting improves the targeted low-energy metrics but introduces middle/high-energy or pooled trade-offs (Table 17). This indicates a practical optimization trade-off under the evaluated static weighting schemes, rather than establishing a universal multi-task Pareto frontier.

Evaluation domain and robustness. The reported detector configurations are trained independently. The present results therefore do not establish zero-shot cross-geometry generalization, detector invariance, or detector-independent representation learning.

The study also does not include a dedicated stochastic noise-pulse robustness benchmark. The perturbation analysis in Table 12 should therefore be interpreted as a controlled observation-stability test rather than as a comprehensive evaluation of the robustness to realistic detector backgrounds. Evaluation of real detector observations and with realistic background contamination remains outside the scope of the present study.