VECMALT: FIXED-CAPACITY GLOBAL–LOCAL REPRESENTATIONS FOR GEOSPATIAL VECTOR GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Encoding geospatial vector geometry for reuse requires a fixed-size code to retain both overall form and spatially localized detail. We introduce VecMALT, which stores a global embedding and a fixed number of local chart records, each pairing a feature with its spatial frame. Coverage-constrained measure allocation bounds token participation, jointly assigns evidence to charts, and supervises their support along sampled boundaries. Across five code widths, local capacity improves complex-polygon reconstruction, while six interventions reveal allocation-dependent trade-offs. We also freeze the codes and train independent readers. Accurate reconstruction does not ensure easy attribute access: for VecMALT, chart-record reading reduces window-occupancy MAE by 31.8% relative to a flat-code MLP, and feature–frame reassignment raises error by 16.3% after reader retraining. On three public polygon cohorts, with geometries available during representation learning and task labels held out, record-aware readers reduce boundary-count MAE in all nine local-method/dataset comparisons. These results distinguish geometric fidelity, spatial support, and independent access as complementary tests of reusable geometry codes.

1 INTRODUCTION

Geospatial vector geometry carries the shape information on which spatial measurement and geometric analysis depend. Building footprints, land-use regions, and transportation curves contain both overall form and localized structure: area depends on the whole object, while a query over a small region may depend on a recess, hole, or sharp bend. A grand challenge for GeoAI is to make this heterogeneous information reusable across learning tasks without losing the structure and detail those tasks need (Chen et al., 2025; Mai et al., 2023).

Learned geometry encoders offer a common interface between vector data and downstream models (Mai et al., 2023; Siampou et al., 2025; Chu & Shahabi, 2026). We study a fixed-capacity interface: each object is encoded once into a code that can be cached and passed to different models. Yet the same budget must represent simple footprints and intricate boundaries, and even similar vertex counts can conceal very different distributions of detail. The difficulty is therefore not just what to retain, but how to organize limited capacity across an object.

Reconstruction alone cannot settle this design question. A jointly trained decoder may recover a detailed boundary from a code that a small, independently trained model finds difficult to use. Area and query-window occupancy provide controlled tests: their targets are computed from the original geometry, but new readers predict them from frozen codes without invoking that decoder. We ask: *How should geometric evidence be allocated within a fixed code budget to support both faithful reconstruction and independent access to global and local information?*

We propose Vector geometry with Measure-Allocated Local Tokens (VecMALT), which combines a global embedding with spatially addressed *local chart records*. Each record pairs a feature with an anchor and metric, all counted within one budget. At fixed chart count, bounded token participation, joint transport, and boundary supervision reconcile *coverage and specialization* (Figure 1), controlling both which evidence forms each record and where its support reaches.

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.

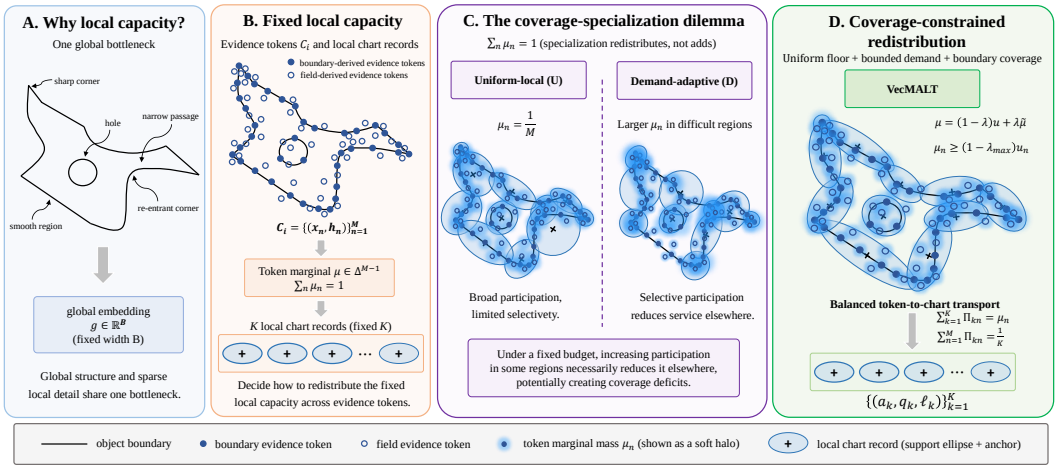


Figure 1: From heterogeneous geometry to coverage-constrained allocation. (A) Overall shape and sparse local detail share a fixed global bottleneck. (B) Adding a fixed number of local records creates an allocation problem: which evidence should each record represent? (C) Concentrating participation on salient regions reduces it elsewhere, exposing a coverage–specialization trade-off. (D) VecMALT bounds redistribution, jointly assigns evidence, and supervises spatial support. Halos denote token mass; ellipses show chart support. The panels are schematic.

We develop Uniform-local and Demand-adaptive as comparison configurations in the same framework. At 512 dimensions, VecMALT raises IoU on the most complex external polygons from Global-only's 91.2% to 96.2%, with Uniform-local performing similarly. Reading VecMALT's code as records lowers window MAE by 31.8% relative to a flat MLP; feature–frame reassignment raises error by 16.3% after reader retraining. Allocation interventions and public tasks reveal trade-offs.

Our contributions are:

1. **A fixed-budget global–local code** that stores object-level context and spatially addressed detail, counting all feature and frame coordinates.
2. **Coverage-constrained evidence allocation** that combines bounded participation, joint transport, and sampled spatial supervision at fixed chart count.
3. **An evaluation of geometric-code reuse** that distinguishes reconstruction, spatial support, and independent readout, with controls for feature–frame correspondence.

2 Related Work

Geometric codes. Polygon encoders address invariance, parts, and holes (Mai et al., 2023); Poly2Vec uses Fourier-based encoding across vector types (Siampou et al., 2025). Geo2Vec jointly optimizes distance-field decoders and object codes with geometry-adaptive sampling (Chu & Shahabi, 2026). Fixed-width codes also occur in vector graphics, including SketchINR (Bandyopadhyay et al., 2024). VecMALT studies how spatially organized evidence forms a reusable code, rather than treating fixed width alone as the contribution.

Locality and allocation. Global-code fields (Park et al., 2019; Mescheder et al., 2019) are complemented by spatial features and local grids (Chibane et al., 2020; Peng et al., 2020; Chabra et al., 2020; Jiang et al., 2020). SIF, LDIF, and PatchNets use structured local elements or explicit frames (Genova et al., 2019; 2020; Tretschk et al., 2020); LDIF counts geometric and latent parameters together (Figure 4 of Genova et al., 2020). DCC-DIF learns continuous code positions, while ACORN and OctField adapt spatial decompositions (Li et al., 2022; Martel et al., 2021; Tang et al., 2021). VecMALT instead controls token participation at fixed chart count and constructs frames through joint transport and conditional moments. Boundary supervision guides their spatial reach.

Independent use. We use established set aggregation and attention (Zaheer et al., 2017; Lee et al., 2019) to test frozen-code use, not as new reader architectures. Same-code comparisons and reassignment test record organization. Appendix L expands these relationships.

3 Fixed-Capacity Geometry Encoding

To preserve overall shape and local detail under one budget, VecMALT first encodes geometric observations, then allocates them to spatial records (Figure 2). The resulting global–local code supports reconstruction and separately trained readers.

3.1 Code and Geometric Evidence

Let $\bar{G}_i$ be geometry G_i in normalized object-local coordinates. We store

$$z_i = \left[g_i; \mathrm{vec}_{k=1}^{K}(a_{ik}, q_{ik}, \ell_{ik})\right], \qquad B = d_g + K(2+3+d_\ell). \tag{1}$$

Here B is code width, $g_i \in \mathbb{R}^{d_g}$ summarizes the object, and vec concatenates K records. Each contains an anchor $a_{ik} \in \mathbb{R}^2$, metric parameters $q_{ik} \in \mathbb{R}^3$ for scale, orientation, and anisotropy, and a feature $\ell_{ik} \in \mathbb{R}^{d_\ell}$. Charts may overlap; Global-only instead uses $z_i = g_i$.

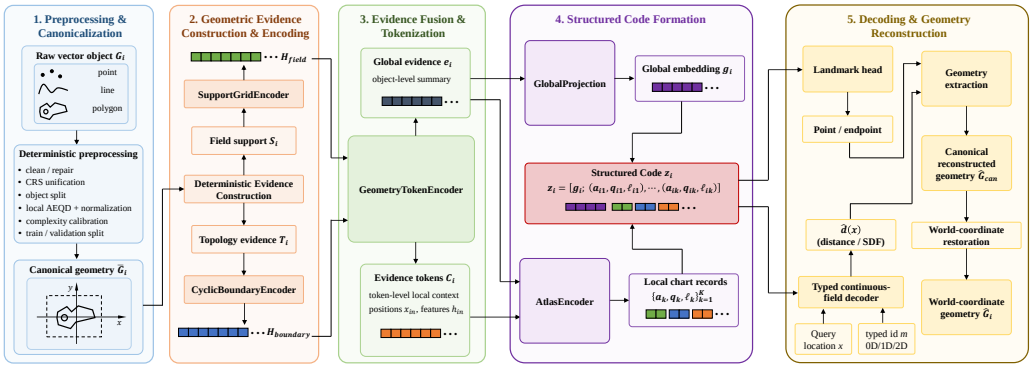


Figure 2: VecMALT architecture, read from left to right. (1) Preprocessing puts each object in normalized coordinates. (2–3) Field and boundary encoders turn distance samples and ordered geometry into spatial evidence tokens and a pooled summary. (4) The global branch projects the summary; the local branch uses both tokens and summary to form spatially addressed records. (5) From the code, field and landmark predictions yield canonical geometry; coordinate restoration follows. Dataset-specific preprocessing is detailed in Appendices A.1 and K.1.

Field samples describe the geometry around an object; boundary evidence supplies its contour and connectivity. The support-grid and cyclic boundary encoders (Figure 2) produce M_i valid tokens, $\mathcal{C}_i = \{(x_{in}, h_{in})\}_{n=1}^{M_i}$, with canonical positions x_{in} and 192-dimensional features h_{in}. Pooling boundary, ring, part, and field features yields the 768-dimensional summary e_i. The global branch projects e_i to g_i; the local branch retains spatial context by reading $(\mathcal{C}_i, e_i)$ before global compression. Appendix A details inputs and supervision.

3.2 Allocation and Chart Construction

The token marginal μ_i specifies total target participation across charts. From uniform mass $u_{in} = 1/M_i$ and a normalized geometric-salience prior p_i, form

$$\begin{aligned} \eta_{in} &= \mathrm{clip}(\log(p_{in}/u_{in}), -c, c), \qquad \delta_{in} = L \tanh f_{\mathrm{imp}}(h_{in}), \\ \widetilde{\mu}_{in} &= \frac{u_{in} e^{\alpha\eta_{in}+\delta_{in}}}{\sum_j u_{ij} e^{\alpha\eta_{ij}+\delta_{ij}}}, \qquad \mu_i(t) = (1-\lambda(t))u_i + \lambda(t)\widetilde{\mu}_i. \end{aligned} \tag{2}$$

Here c clips prior contrast, α scales it, and L bounds the residual from the learned scalar predictor f_{imp}. At training step t, the mixing schedule satisfies $0 \le \lambda(t) \le \lambda_{\max} = 0.25$, so $\mu_{in} \ge 0.75 u_{in}$

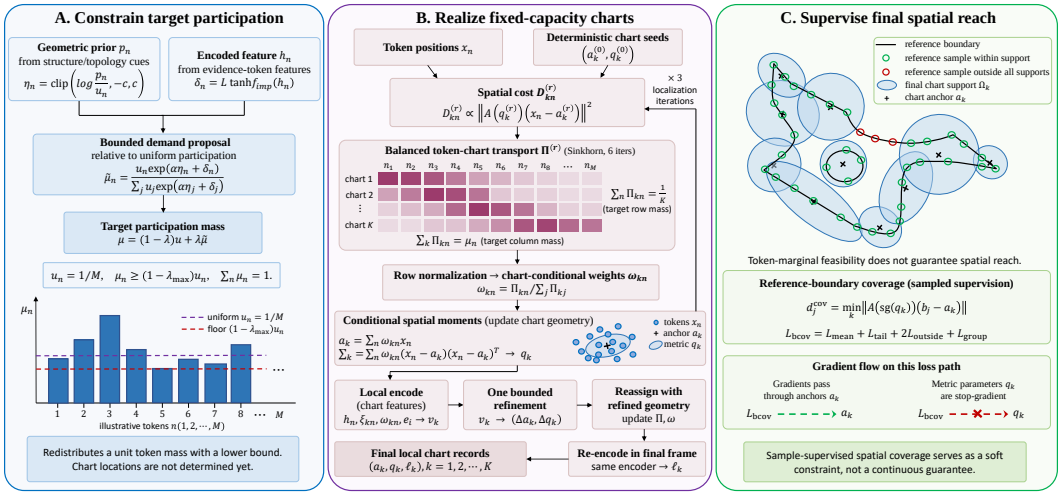


Figure 3: Coverage-constrained allocation. (A) Bounded demand is mixed with uniform token mass. (B) Transport and moments localize charts; reassignment precedes initial local encoding and follows frame refinement, before final re-encoding. (C) Reference-boundary supervision differentiates through anchors but stops metric gradients on this loss path.

and $\frac{1}{2}\|\mu_i - u_i\|_1 \leq \lambda(t)$. These are bounds on target token mass, not spatial coverage. Appendix A.6 gives the derivation and no-floor intervention.

At iteration r, squared chart-relative distance defines $D_{ikn}^{(r)} = \frac{1}{2}(x_{in} - a_{ik}^{(r)})^\top Q(q_{ik}^{(r)})(x_{in} - a_{ik}^{(r)})$, where $A(q)$ maps offsets to chart coordinates and $Q(q) = A(q)^\top A(q)$ is positive definite. With entropy weight $\tau > 0$, joint transport solves (Cuturi, 2013)

$$\Pi_i^* = \arg\min_{\Pi_i \geq 0} \left[\langle \Pi_i, D_i \rangle + \tau \sum_{k,n} \Pi_{ikn}(\log \Pi_{ikn} - 1) \right], \qquad \Pi_i \mathbf{1}_{M_i} = K^{-1}\mathbf{1}_K, \qquad \Pi_i^\top \mathbf{1}_K = \mu_i. \tag{3}$$

Here $\mathbf{1}_d$ is the all-ones vector of length d. Shared column constraints coordinate evidence use across equal-mass rows. Six log-domain Sinkhorn steps give $\widehat{\Pi}_i$. Row-normalized weights $\omega_{ikn} = \widehat{\Pi}_{ikn} / \sum_j \widehat{\Pi}_{ikj}$ drive three assignment–moment iterations from deterministic seeds:

$$a_{ik}^{(r+1)} = \sum_n \omega_{ikn}^{(r)} x_{in}, \qquad \Sigma_{ik}^{(r+1)} = \sum_n \omega_{ikn}^{(r)} (x_{in} - a_{ik}^{(r+1)})(x_{in} - a_{ik}^{(r+1)})^\top. \tag{4}$$

The covariance Σ gives bounded radii and orientation through a stabilized map. After the final update, assignments are recomputed before local encoding. A shared encoder aggregates features and relative coordinates, conditioned on e_i and chart identity. One bounded frame correction is followed by reassignment, re-encoding, and projection to ℓ_{ik} (Figure 3). Unindexed frames denote this final refined state. Appendix A.3 specifies parameterization and finite-iteration behavior.

3.3 Spatial Support and Geometry Decoding

Mass constraints alone cannot place compact supports along the boundary. We supervise final charts at reference samples b_{ij}, grouped by ring or part, using

$$d_{ij}^{\mathrm{cov}} = \min_k \sqrt{(b_{ij} - a_{ik})^\top Q(\mathrm{sg}(q_{ik}))(b_{ij} - a_{ik})}. \tag{5}$$

Here sg stops gradients. The objective combines group-balanced, tail, outside-support, and group-worst penalties (Appendix A.5). Stopping metric gradients blocks direct support enlargement through this loss; other losses train the metrics. The loss acts on boundary samples.

For a query x, let $\xi_{ik} = A(q_{ik})(x - a_{ik})$ and $r_{ik} = \|\xi_{ik}\|_2$. A compact kernel κ vanishes at $r \geq 1$. With $Z_i = \sum_k \kappa(r_{ik})$, $c_i = \min(1, Z_i)$, and blending weights $\bar{\omega}_{ik} = \kappa(r_{ik})/Z_i$ when $Z_i > 0$ (zero otherwise), the field of type m (signed for polygons, unsigned for curves) is

$$\begin{aligned}\widehat{d}_i(x, m) = f_{\text{coarse}}(x, g_i, m) \\ + c_i(x) \left[f_{\text{detail}}(x, g_i, m) + \sum_k \bar{\omega}_{ik}(x) f_{\text{loc}}(\ell_{ik}, \xi_{ik}(x), m) \right].\end{aligned} \tag{6}$$

Thus support gates both global detail and local residuals; outside it only the coarse field contributes. Global-only keeps global detail active everywhere. Polygon SDF contours and curve UDF bands yield canonical predictions; coordinate restoration follows extraction. Field, geometry, regularity, and chart losses train the code jointly (Appendix A.4).

3.4 Independent Readers

After freezing codes and representation weights, new readers predict $A_i = \text{Area}(\bar{G}_i)$ or window occupancy $\rho_i(W) = \text{Area}(\bar{G}_i \cap W)/(wh)$ for a rectangle $W = (c_x, c_y, w, h)$ with center (c_x, c_y) and width/height $w, h > 0$. Area uses a flat MLP. Windows compare a flat MLP, attention over 25 nonspatial code chunks, and attention over a global token plus complete chart records. A window query selects record information without the decoder's support mask. Appendix H.2 specifies normalization and reader architectures.

To test feature–frame correspondence, an object-specific permutation replaces records by $(a_{ik}, q_{ik}, \ell_{i,\pi_i(k)})$, preserving g_i and both sets. We transform every split and retrain the same reader. Public Edge uses an object-query record reader; Shape uses a global-conditioned query and direct global path. Appendix K.4 retains its exploratory revision and original results.

4 Experimental Design

Data and tasks. Overture-derived buildings, land-use regions, and transportation segments provide complementary polygon and curve geometry. We use 355,820 training and 39,536 internal-validation objects, with 39,473 polygons and 14,906 curves from external sources. Source composition and training-frozen complexity strata are in Appendix B. External area/window readers use 50,000/10,000/20,000 polygons for training/validation/test; each test polygon has 64 shared windows, totaling 1.28 million queries. These test independent access on new objects.

Public Shape (5,010 polygons), Singapore (109,877), and NYC (20,000 polygons) complement these tasks with shape classification and stored boundary-segment counting (Edge). Here representations learn from *all cohort geometries without task labels*; readers then use common label splits. This transductive protocol is distinct from external new-object evaluation. Edge includes hole boundaries and collinear vertices, so it tests source discretization as well as shape (Appendix K.1).

Comparisons within our framework. Global-only is the global control. Our three local configurations share the record layout: Uniform-local sets $\mu_i = u_i$; Demand-adaptive uses demand weighting; VecMALT bounds redistribution and supervises support. Demand-adaptive also differs in cost, refinement, and regularization (Appendix A.7); it is not the no-floor ablation. Local methods share $d_g = B/4$, $d_\ell = 11$, and $K = 6, 12, 24, 36, 48$ at $B = 128, 256, 512, 768, 1024$. Thus chart count is matched *within* each budget. The 20 runs and six $B = 512$ interventions use common evidence, seed 42, 100,000 steps, and internal Area-priority selection (Appendix A.4). Code width is matched, not compute or network size (VecMALT: 7.65M parameters at $B = 512$; Table K2).

Our external baseline, adapted Geo2Vec, jointly learns a decoder and training-object codes, then fits new external codes for 1,000 steps with the decoder frozen, without task labels. Public Geo2Vec instead exports jointly learned cohort codes. Encoder methods use forward passes over prepared evidence. Readers are selected by validation MAE or accuracy (Appendices H–K).

Evaluation. Reconstruction uses 512×512 fields without contour refinement. Vector IoU retains empty predictions at zero; boundary distances use finite predictions. Curve extraction differs across method families. Source-stratified spatial-block bootstrap intervals condition on the fitted models, not training-seed variation. Internal allocation penalties and external sampled-support fractions are distinct diagnostics; Appendix B defines metrics and aggregation.

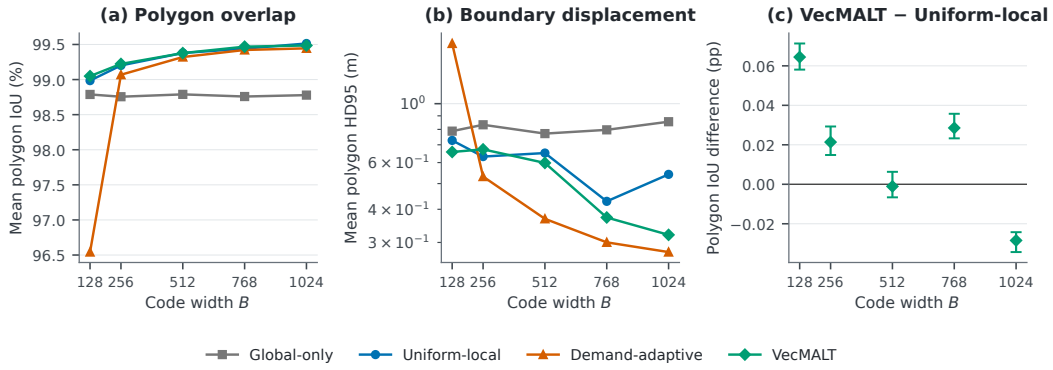


Figure 4: Code-width effects on external polygons: (a) mean IoU; (b) mean boundary HD95 in meters (log scale); (c) paired VecMALT-minus-Uniform-local IoU differences, in percentage points. Intervals are 95% source-stratified spatial-block bootstrap intervals for the fitted models. Width includes features and frame coordinates.

5 Results

5.1 Reconstruction under Fixed Budgets

At $B = 512$, VecMALT reaches 96.19% IoU on the 3,377 highest-complexity polygons versus Global-only's 91.23%, with Uniform-local performing similarly (Table C1). The shared gain supports explicit local capacity; allocation determines the trade-offs within this family. Across widths, VecMALT exceeds Demand-adaptive in polygon IoU; against Uniform-local it is essentially tied at 512 and lower at 1024 (Figure 4).

Table 1: External reconstruction at $B = 512$ on 39,473 polygons and 14,906 curves. Global-only, Uniform-local, Demand-adaptive, and VecMALT are configurations of our framework; Geo2Vec is the external baseline. CD denotes Chamfer distance; HD95 mean and median aggregate per-object boundary HD95. Uniform-local and Demand-adaptive each have one empty polygon prediction, included with IoU zero and undefined boundary distance. Appendix J gives canonical distances and extraction protocols. Bold denotes the best unrounded value in each metric column.

Method	Polygon IoU (%) ↑	Polygon HD95 mean (m) ↓	Polygon HD95 median (m) ↓	Curve CD mean (m) ↓	Curve HD95 mean (m) ↓
Geo2Vec (adapted)	98.5285	0.401	0.030	0.409	1.318
Global-only	98.7891	0.771	0.009	**0.362**	**0.895**
Uniform-local	**99.3783**	0.652	**0.009**	0.411	0.937
Demand-adaptive	99.3210	**0.368**	0.009	0.499	1.216
VecMALT	99.3772	0.598	0.009	0.406	0.968

Figure 6 illustrates these differences on selected objects: VecMALT follows recessed and irregular outlines while retaining holes and disconnected parts. Global-only merges holes or joins parts; Geo2Vec shows contour oscillations and extra components. These cases complement, rather than replace, the full-set comparison in Table 1.

Demand-adaptive leads meter-valued polygon HD95, while Global-only leads curve Chamfer distance (Table 1). VecMALT's canonical polygon HD95 is 55.0% lower than Geo2Vec's, but its meter-valued mean is higher. One large land-use polygon contributes 53.7% of VecMALT's meter HD95 sum, showing how scale conversion amplifies large-object errors (Appendix J.2).

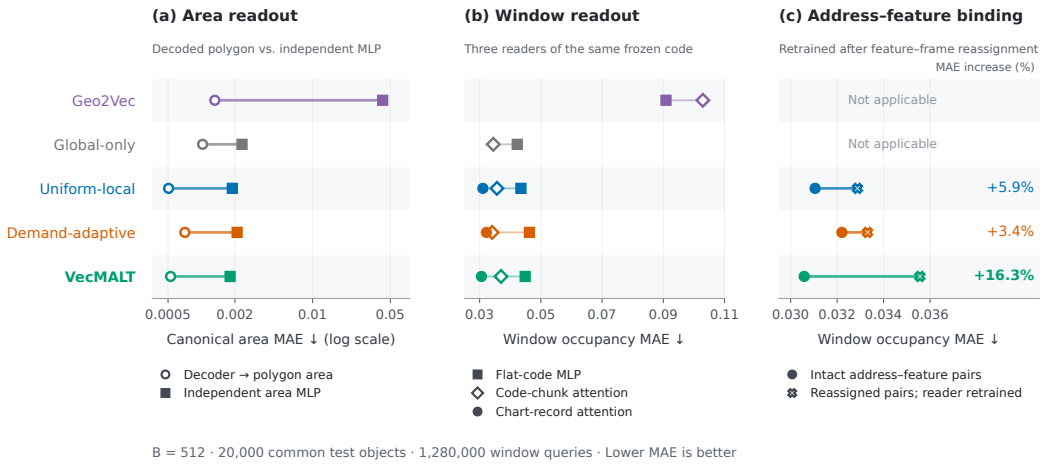


Figure 5: Frozen-code use on 20,000 common polygons at $B = 512$. (a) Reconstruction-derived versus direct area MAE. (b) Three window readers on 1.28 million queries. (c) Intact versus reassigned feature–frame pairs, with reader retraining after reassignment in every split; percentages are relative MAE increases. Protocols: Appendices H and J.

5.2 Independent Readout and Pairing

VecMALT gives the lowest direct area MAE, 0.001809 (Figure 5a). All methods estimate area more accurately by reconstructing first, but direct reading avoids dense field evaluation and extraction. For adapted Geo2Vec, area-MLP MAE is 32.3 times reconstruction-derived MAE despite 98.54% IoU. Thus accurate reconstruction need not make attributes easy for a new reader to extract.

Chart-record reading lowers VecMALT's window MAE to 0.030585: 31.8% below its flat MLP and 17.3% below code-chunk attention. Local codes do not outperform Global-only with flat window readers. Under the same record reader, VecMALT's advantages over Uniform-local and Demand-adaptive are smaller, 1.5% and 5.0% (Table H2). Exposing each feature with its frame gives the query a spatially organized input, although sharing and normalization also change.

Reassignment tests the correspondence separately: VecMALT's error rises by 16.3% after retraining, despite preserving the global code and both feature and frame sets (Figure 5c). This supports their original pairing rather than merely storing the two sets.

5.3 Allocation and Spatial Support

At $B = 512, K = 24$, Uniform-local and VecMALT leave no uncovered samples in the 54,379-object diagnostic; Demand-adaptive leaves 36.97% on average. VecMALT's individual supports are 2.3% smaller on average than Uniform-local's (Table H3). This is a configuration comparison, not a uniform-floor intervention.

Joint assignment matters even with near-uniform target mass. Independent transport raises the internal boundary penalty from 1.9×10^{-5} to 3.174×10^{-2} and causes the largest external polygon-IoU loss among interventions, 0.0727 points (Tables D1 and 2). This loss diagnostic is not an external uncovered fraction. Removing the floor lowers the mean per-object minimum participation ratio from 0.9406 to 0.6144, but improves polygon IoU. Isotropic charts also improve IoU while worsening curve HD95 from 0.9684 to 1.3080 m. Removing refinement or boundary supervision lowers polygon IoU. All six interventions improve buffered-curve IoU: allocation controls a trade-off, not one universally best operating point.

Coverage and fitting remain distinct. Uniform-local and VecMALT reach all sampled boundaries in the two-hole example but differ in contour accuracy (Figures 6 and A3). Independent attention does not enforce the decoder's support gate, so pairing tests a different property.

Table 2: Six interventions at $B = 512$. Δ is intervention minus VecMALT in polygon-IoU percentage points, computed before rounding. IoU includes all 39,473 polygons; HD95 uses 39,472 finite predictions for Independent transport and No refinement and 39,473 otherwise. Buffered-curve IoU compares the fixed distance-buffer masks. Appendix C gives internal reconstruction; Table D1 gives internal allocation diagnostics and Table H3 external support measurements for the main local configurations. Bold denotes the best unrounded value in each metric column, including the reference.

Configuration	**Polygon IoU (%) ↑**	**Δ (pp)**	**Polygon HD95 mean (m) ↓**	**Buffered-curve IoU (%) ↑**	**Curve HD95 mean (m) ↓**
VecMALT (reference)	99.3772	—	0.598	73.408	0.9684
No learned importance	99.3726	-0.0046	0.412	73.537	1.0627
Independent transport	99.3045	-0.0727	0.467	73.724	0.9755
Isotropic charts	**99.4291**	**+0.0519**	0.554	**74.290**	1.3080
No refinement	99.3323	-0.0450	0.617	73.693	**0.8834**
No explicit uniform floor	99.3881	+0.0109	**0.324**	73.706	1.0363
No reference-boundary coverage	99.3695	-0.0077	0.602	73.673	0.9139

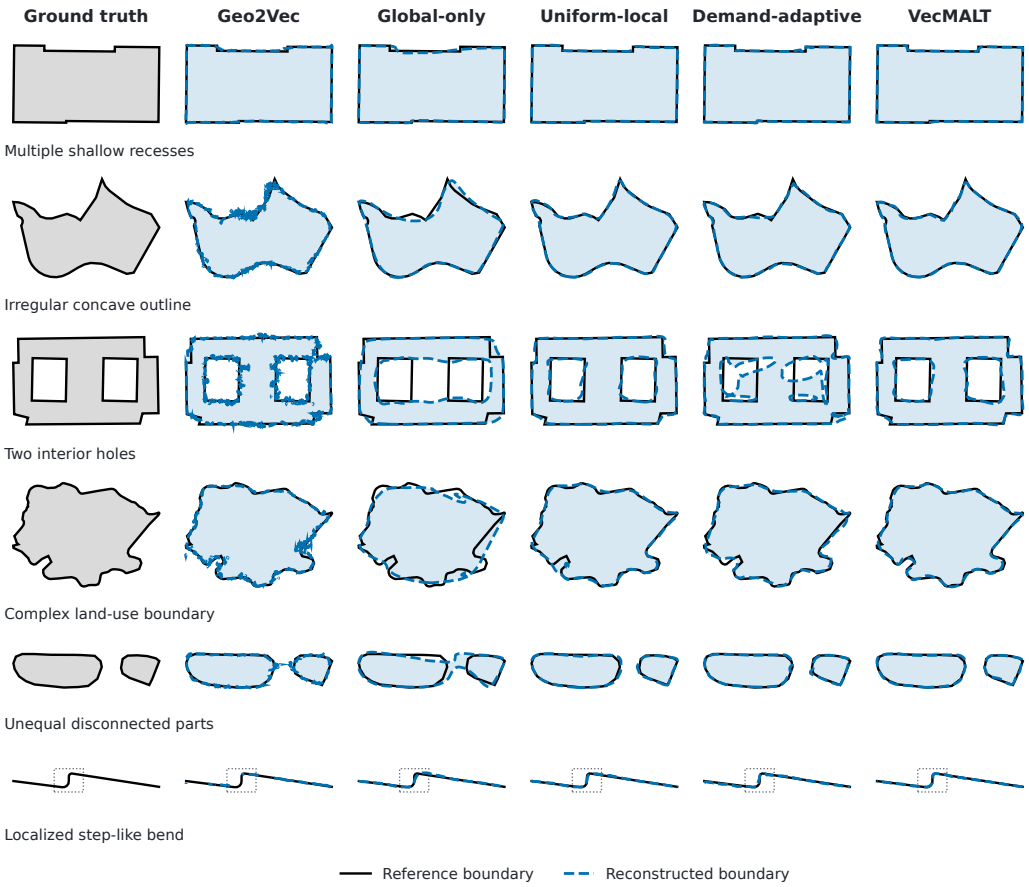


Figure 6: Matched reconstructions at $B = 512$ ($K = 24$ for local methods). Selected objects illustrate recesses, irregular contours, holes, disconnected parts, and a local curve bend. Black solid boundaries are references; blue dashed boundaries are predictions. Views are shared within rows, retaining every component. Rectangles locate the close-up in Figure A8; selection and extraction follow Appendix F.1.

5.4 Public Tasks and Reuse Costs

Table 3: Public-task readout from frozen 512-dimensional codes, with our configurations and the Geo2Vec baseline as in Table 1. A: common flat MLPs. B: task-adapted local readers; Geo2Vec and Global-only reuse A. Shape uses revised classifiers (Appendix K.4); Edge uses object-query readers, with MAE in segments. All geometries enter representation learning without task labels. Bold marks the best unrounded value per column and panel.

Method	Shape classification accuracy (%) ↑	Boundary-segment count MAE ↓		
		Shape	Singapore	NYC polygons
A. Common flat-code MLP				
Geo2Vec (adapted)	95.07	2.6452	2.3416	1.0447
Global-only	96.00	**0.8993**	**0.6724**	0.4227
Uniform-local	97.07	1.2041	0.8364	0.4369
Demand-adaptive	**97.60**	1.0221	0.9292	**0.3261**
VecMALT	95.60	1.2167	0.7647	0.4062
B. Task-adapted record-aware reading				
Geo2Vec (adapted)	95.07	2.6452	2.3416	1.0447
Global-only	96.00	0.8993	0.6724	0.4227
Uniform-local	97.20	1.0472	0.6799	0.2914
Demand-adaptive	**97.87**	**0.8142**	0.6620	**0.2314**
VecMALT	96.67	0.9883	**0.5581**	0.2978

Record-aware readers reduce Edge MAE by 13.0–33.3% in all nine same-code comparisons, with fewer parameters than flat MLPs (Table 3). VecMALT leads Singapore at 0.5581 MAE; Demand-adaptive leads Shape and NYC. Uniform-local's NYC RMSE nevertheless increases (Table K6). Shape is reader-sensitive: the exploratory global-conditioned revision restores performance close to flat MLPs, while the original reader is weaker (Table K5).

Caching separates acquisition from reuse. On 32 common objects, mean acquisition from canonical geometry is 2.35 s for VecMALT versus 3.18 s for Geo2Vec, but VecMALT has higher P95 (9.40 versus 3.32 s). Prepared-evidence encoding alone is 18.73 ms. Cached bulk reading processes 20,000 areas in 0.0207 s or 1.28 million windows in 0.0659 s, including the specified transfers. Each 512-dimensional FP32 code occupies 2 KiB. Appendices G and I specify workload boundaries, additional storage, probes, and retrieval.

6 Discussion and Conclusion

VecMALT organizes object-level context and spatially addressed detail within a fixed code budget. Three findings emerge. Explicit local capacity improves complex-polygon reconstruction across allocation rules. At fixed chart count, near-uniform token participation alone does not ensure spatial reach. Reconstruction fidelity likewise does not guarantee independent access: information recoverable by the original decoder can remain difficult for a new model to read, whereas exposing local features together with their learned spatial frames improves spatial readout. Readout controls do not isolate individual allocation losses. The external corpus has limited topological diversity; public tasks use known geometries with held-out labels.

The summary–record layout parallels class and patch tokens in vision transformers (Dosovitskiy et al., 2021), but its local records carry explicit spatial frames placed by allocation rather than occupying fixed grid positions. This structure provides a concrete starting point for future multimodal interfaces that combine whole-object context with spatially addressed local geometry without returning to the original coordinate sequence.

GENERATIVE AI ASSISTANCE

Generative AI tools were used as an auxiliary aid for research ideation or execution and for drafting and polishing portions of the manuscript. The authors reviewed and revised all AI-assisted material and independently verified the final results and scientific claims. The authors take full responsibility for the final content of this work.

ETHICS AND DATA ATTRIBUTION

The study uses Overture-derived vector geometry and public polygon datasets, with no human-subject experiments. Source errors and uneven geographic coverage remain relevant even when reconstruction is accurate. Overture data carry source-specific attribution and licensing requirements (Overture Maps Foundation, n.d.). We identify the corpus through source manifests and frozen preprocessing records; the available metadata do not contain an upstream release identifier. The public cohorts use data distributed with Geo2Vec and retain their recorded source attribution and coordinate systems. Evaluation use does not establish permission for redistribution.

REPRODUCIBILITY

Appendices A–G specify the representation, the 20-model capacity study, six interventions, qualitative examples, probes, retrieval, and profiling. Appendices H–J document independent area/window prediction, binding controls, Geo2Vec fitting, and common-object reconstruction. Appendix K records all fifteen public-cohort models, task readers, and the Shape-head revision. Appendix A.8 expands the encoder and coordinate formulas, Appendix H.2 gives complete reader equations, and Appendix L gives extended related work. The accompanying figure/table assets contain local numerical inputs and plotting scripts for Figures 4–6, A1–A12, and the tables. They reproduce the reported displays without model inference; reproducing training and evaluation is a separate task.

REFERENCES

Hmrishav Bandyopadhyay, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Tao Xiang, Timothy Hospedales, and Yi-Zhe Song. Sketchinr: A first look into sketches as implicit neural representations. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12565–12574, 2024. doi: 10.1109/CVPR52733.2024.01194. URL https://doi.org/10.1109/CVPR52733.2024.01194.

Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16351–16361. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/bcf9d6bd14a2095866ce8c950b702341-Paper.pdf.

Rohan Chabra, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Computer Vision - ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX*, pp. 608–625, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58525-9. doi: 10.1007/978-3-030-58526-6_36. URL https://doi.org/10.1007/978-3-030-58526-6_36.

Yile Chen, Weiming Huang, Kaiqi Zhao, Yue Jiang, and Gao Cong. Self-supervised representation learning for geospatial objects: A survey. *Information Fusion*, 123:103265, 2025. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2025.103265. URL https://www.sciencedirect.com/science/article/pii/S1566253525003380.

Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6968–6979, 2020. doi: 10.1109/CVPR42600.2020.00700. URL https://doi.org/10.1109/CVPR42600.2020.00700.

Chen Chu and Cyrus Shahabi. Geo2Vec: Shape- and distance-aware neural representation of geospatial entities. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(23):18985–18993, 2026. doi: 10.1609/aaai.v40i23.38970. URL https://ojs.aaai.org/index.php/AAAI/article/view/38970.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7153–7163, 2019. doi: 10.1109/ICCV.2019.00725. URL https://doi.org/10.1109/ICCV.2019.00725.

Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4856–4865, 2020. doi: 10.1109/CVPR42600.2020.00491. URL https://doi.org/10.1109/CVPR42600.2020.00491.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning*

Research, pp. 3789–3799. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/gropp20a.html.

Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6000–6009, 2020. doi: 10.1109/CVPR42600.2020.00604. URL https://doi.org/10.1109/CVPR42600.2020.00604.

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 Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Tianyang Li, Xin Wen, Yu-Shen Liu, Hua Su, and Zhizhong Han. Learning deep implicit functions for 3d shapes with dynamic code clouds. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12830–12840, 2022. doi: 10.1109/CVPR52688.2022.01250. URL https://doi.org/10.1109/CVPR52688.2022.01250.

Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells. In *International Conference on Learning Representations*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rJljdh4KDH.

Gengchen Mai, Chiyu Jiang, Weiwei Sun, Rui Zhu, Yao Xuan, Ling Cai, Krzysztof Janowicz, Stefano Ermon, and Ni Lao. Towards general-purpose representation learning of polygonal geometries. *GeoInformatica*, 27(2):289–340, 2023. doi: 10.1007/s10707-022-00481-2. URL https://doi.org/10.1007/s10707-022-00481-2.

Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: adaptive coordinate networks for neural scene representation. *ACM Trans. Graph.*, 40(4), 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459785. URL https://doi.org/10.1145/3450626.3459785.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4455–4465, Los Alamitos, CA, USA, June 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00459. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00459.

Overture Maps Foundation. Overture schema reference and attribution and licensing. Online documentation, n.d. URL https://docs.overturemaps.org/schema/. Also https://docs.overturemaps.org/attribution/. Documentation accessed 24 September 2026; this is not the corpus acquisition date or release version.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174, 2019. doi: 10.1109/CVPR.2019.00025. URL https://doi.org/10.1109/CVPR.2019.00025.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision - ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, pp. 523–540, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58579-2. doi: 10.1007/978-3-030-58580-8_31. URL https://doi.org/10.1007/978-3-030-58580-8_31.

Maria Despoina Siampou, Jialiang Li, John Krumm, Cyrus Shahabi, and Hua Lu. Poly2Vec: Polymorphic Fourier-based encoding of geospatial objects for GeoAI applications. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine*

Learning, volume 267 of *Proceedings of Machine Learning Research*, pp. 55511–55532. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/siampou25a.html.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf.

Jia-Heng Tang, Weikai Chen, jie Yang, Bo Wang, Songrun Liu, Bo Yang, and Lin Gao. Octfield: Hierarchical implicit functions for 3d modeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12648–12660. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/698d51a19d8a121ce581499d7b701668-Paper.pdf.

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3d shape representations. In *Computer Vision - ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVI*, pp. 293–309, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58516-7. doi: 10.1007/978-3-030-58517-4_18. URL https://doi.org/10.1007/978-3-030-58517-4_18.

Meng Wang, Yu-Shen Liu, Yue Gao, Kanle Shi, Yi Fang, and Zhizhong Han. Lp-dif: Learning local pattern-specific deep implicit function for 3d objects and scenes. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21856–21865, 2023a. doi: 10.1109/CVPR52729.2023.02093. URL https://doi.org/10.1109/CVPR52729.2023.02093.

Zhen Wang, Shijie Zhou, Jeong Joon Park, Despoina Paschalidou, Suya You, Gordon Wetzstein, Leonidas Guibas, and Achuta Kadambi. Alto: Alternating latent topologies for implicit 3d reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 259–270, 2023b. doi: 10.1109/CVPR52729.2023.00033. URL https://doi.org/10.1109/CVPR52729.2023.00033.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

A Representation and Implementation Details

A.1 Canonical Coordinates and Stored Metadata

The Overture-derived study validates or repairs input geometry, maps geographic coordinates into an object-local metric frame, and applies deterministic orientation normalization, translation, and isotropic scaling. The public polygon study preserves source orientation and uses translation and isotropic scaling without repair or simplification. Each nondegenerate object is scaled by its bounding-box longest side to fit a unit box; the field-query domain is the larger $[-0.75, 0.75]^2$. The pose record P_i stores the map T_{P_i} for coordinate normalization and its inverse for restoring the preprocessed object's placement. Its scale also converts canonical boundary distances to meters.

In the reported curve and polygon experiments, shape predictions are conditioned on z_i; query coordinates and field type specify the field being evaluated. The stored structural header H_i^{str} records geometry types, component and hole counts, open/closed status, and container metadata, but is not used to determine the predicted canonical geometry. Appendix G.3 includes these metadata in the measured serialization cost.

The general implementation also supports point geometry. An isolated point has canonical geometry $\bar{G}_i = \{0\}$ and analytic UDF $\|x\|_2$; its location comes from the pose. It uses a shared fixed code and is excluded from learned training and validation. Multipoint objects use the learned point-set pathway. These point pathways are outside the reported curve/polygon reconstruction experiments.

Field samples and topology evidence are encoder inputs. Query distances, reliable derivative targets, singular probes, reference-boundary samples, area sentinels, and point/endpoint targets supervise the predicted fields and allocation. All may be derived from the same reference geometry, but the supervision arrays are not retained as decoder inputs. The distinction is their role in the computation: some normal and probe descriptors also belong to the topology input. Evidence construction is deterministic under the recorded seed and settings; the field and boundary feature encoders are learned.

The original capacity study caps cached support and query pools at 8,192 samples per object and topology evidence at 4,096 entries. Support and query policies use boundary fractions 0.90 and 0.94 and normalized distance bands 0.050 and 0.035. These settings guide adaptive selection; actual counts and sampling proportions vary with geometry. Training and internal validation use at most 7,168 and 6,144 query samples. A separate 16×16 area-sentinel grid supervises ambient occupancy. Each support sample has seven channels, and each topology descriptor has 24, accompanied by neighborhood, ring, part, and validity indices.

A.2 Code Layout and Encoder Inference

Table A1: Local-code layouts and FP32 storage. Global-only assigns all B coordinates to its global embedding. Pose, structural headers, and shared networks are additional.

Code width B	**Global width** d_g	**Chart count** K	**Scalars per chart record**	**FP32 code (KiB)**
128	32	6	2 + 3 + 11 = 16	0.5
256	64	12	2 + 3 + 11 = 16	1.0
512	128	24	2 + 3 + 11 = 16	2.0
768	192	36	2 + 3 + 11 = 16	3.0
1024	256	48	2 + 3 + 11 = 16	4.0

Table A1 counts all stored floating-point values before quantization or entropy coding. Each local record has two anchor coordinates, three metric coordinates, and eleven feature coordinates. The isotropic intervention keeps the same layout but fixes two anisotropy coordinates to zero, so stored width and active geometric degrees of freedom differ. Pose and structural-header storage are reported separately in Appendix G.3.

The shared encoder predicts a code for each object without an independently optimized object-parameter table. In contrast, an auto-decoder jointly optimizes its shared decoder and an $N \times d$ table for N training objects. This distinction concerns training parameters and code acquisition. Saving N output codes costs NB scalars for either approach, in addition to side information and shared networks.

A.3 Geometric Prior and Chart Construction

The topology prior assigns salience to turns, endpoints, junctions, and holes. For turning angle ϑ_n, with $I_{\text{endpoint}}, I_{\text{junction}}, I_{\text{hole}}$ denoting the corresponding binary indicators, the positive score is $1 + 2\sqrt{\sin^2 \vartheta_n + (1 - \cos \vartheta_n)^2} + 1.5 I_{\text{endpoint}} + 1.5 I_{\text{junction}} + I_{\text{hole}}$. Scores are normalized within each ring or part. Group mass is an equal-weight mixture of equal group shares and normalized square-root group lengths. Ring and part distributions are then combined with weights 0.70 and 0.30. The field prior uses normalized $\log(1 + \text{mass})$ of occupied support cells. When both streams are present, topology and field receive total masses 0.75 and 0.25. Missing streams and masked entries are handled before final valid-token normalization.

The boundary encoder has four layers of width 192, and the field encoder uses a 32×32 support grid. Both produce 192-dimensional tokens. Pooling boundary, ring, part, and field features yields the 768-dimensional summary e_i. The main decoder hidden width is 256. Chart anchors start on a deterministic lattice with isotropic radius 0.22. Three localization iterations alternate assignments and moments; each assignment uses six log-domain Sinkhorn iterations with temperature 0.12. Assignments are recomputed after the final moment update. Learnable chart identities and e_i condition the local encoder, rather than replacing the spatial seeds.

Principal radii are 2.5 times the stabilized square roots of moment eigenvalues, bounded to 0.035–0.35 with maximum axis ratio 6. The base radius is $s_0 = 0.10$. For principal direction θ, let $R(\theta)$ be the planar rotation matrix. The chart transform is $A(q) = R(\theta)\,\mathrm{diag}(1/s^{\text{maj}}, 1/s^{\text{min}})R(\theta)^\top$, with $Q = A^\top A$. A stabilized double-angle formulation avoids choosing a sign for the principal axis.

The reference uses residual limit $L = 0.35$, prior strength $\alpha = 0.50$, clipping limit $c = \log 4$, and no content-affinity term in the transport cost. Anchor corrections are bounded with coefficient 0.20 in chart-local coordinates before mapping back to the canonical frame; metric corrections use coefficient 0.15. The final correction layer initially outputs zero. After this correction, the system recomputes assignments and local states, then projects the final state to ℓ_{ik}.

Numerical safeguards stabilize the prior logarithm, metric construction, coverage-distance square root, and decoder normalization. The decoder denominator is $\max(Z_i(x), 10^{-8})$. At exact transport feasibility, $\omega_{ikn} = K\Pi^*_{ikn}$ and $K^{-1}\sum_k \omega_{ikn} = \mu_{in}$; finite Sinkhorn iterations followed by row normalization need not preserve the second identity exactly. The participation bounds apply to the target marginal μ_i. Moment formulas describe localization updates, while the final stored frame follows refinement and reassignment.

The global feature pyramid has channels 128/112/96/64/32, each projected to width 32. The coarse predictor uses the first two, coarser levels and low-frequency coordinates; the detail predictor uses all levels and higher-frequency coordinates. The pyramid is generated from g_i during decoding.

A.4 Optimization and Training Objectives

The 20 capacity-sweep runs and six interventions use AdamW with learning rate 10^{-4}, weight decay 10^{-6}, 5,000 warm-up steps, and a cosine schedule over 100,000 steps with minimum learning-rate ratio 0.05. Training uses eight ranks and a cap of 16 objects per rank under cost-based batching, so the realized object count can be smaller. Internal validation runs every 1,000 steps; remaining batches at the end of a data pass are retained. Early-stopping patience is 10,000 steps. All 26 recorded runs reach the step cap, with selected checkpoints between 97,000 and 100,000 steps.

Area-field loss has dimension weight 1.5 and area-family sampling weight 1.25. The *Area-priority* selection criterion combines internally evaluated area-family tier field errors and area-only geometry terms, using the configured tier weights. External reconstruction scores are not used for selection. VecMALT's adaptive fraction follows a cosine ramp from zero at step 5,000 to $\lambda_{\max}$ at step 20,000; checkpoints become eligible after the ramp. Demand-adaptive is eligible from its first vali-

dation. At $B = 512$, the selected steps are 99,000/100,000/98,000/98,000 for Global-only/Uniform-local/Demand-adaptive/VecMALT.

We group the training objective by function. Each group includes its configured coefficients; the reference-boundary coefficient is shown separately:

$$\mathcal{L} = \mathcal{L}_{\text{field}} + \mathcal{L}_{\text{geometry}} + \mathcal{L}_{\text{regularity}} + \mathcal{L}_{\text{chart}} + \lambda_{\text{bcov}}\mathcal{L}_{\text{bcov}}.$$

Field losses fit typed distances and the coarse/detail decomposition, with stronger boundary weighting. Geometry losses constrain reference and predicted boundaries, area signs, off-boundary behavior, sentinel samples, and applicable point-set or endpoint targets. Regularity losses supervise reliable gradients, tangents, and local field behavior, omitting singular or ambiguous derivative targets. Chart losses regulate assignment compactness, metric consistency, importance residuals, and refinement. The reference uses $\lambda_{\text{bcov}} = 0.15$.

Fourier coordinate features support fitting high-frequency variation with coordinate MLPs (Tancik et al., 2020), and unit-gradient regularization follows the geometric motivation of Gropp et al. (2020). The reference-boundary objective below acts on chart support rather than field derivatives.

A.5 Sampled Reference-Boundary Coverage Objective

For one object, write $d_j = d_{ij}^{\text{cov}}$. Let $\mathcal{G}_i$ contain its nonempty reference groups, let J_g index samples in group g, and let φ_{ij} identify the group of sample b_{ij}. The group-balanced weight is $w_j = 1/(|\mathcal{G}_i||J_{\varphi_{ij}}|)$. Define $h_{\rho_{\text{cov}}}(d) = \max(d - \rho_{\text{cov}}, 0)^2$, with $\rho_{\text{cov}} = 0.85$, and let J_{tail} contain the largest 10% of valid distances, rounded up to at least one. The object-level loss is

$$\begin{aligned}\mathcal{L}_{\text{bcov},i} = &\sum_j w_j h_{\rho_{\text{cov}}}(d_j) + \frac{1}{|J_{\text{tail}}|}\sum_{j\in J_{\text{tail}}} h_{\rho_{\text{cov}}}(d_j) \\ &+ 2\sum_j w_j h_1(d_j) + \frac{1}{|\mathcal{G}_i|}\sum_{g\in\mathcal{G}_i} h_{\rho_{\text{cov}}}\left(\max_{j\in J_g} d_j\right).\end{aligned}$$

The first term weights groups equally, the second emphasizes poorly served samples, the third penalizes samples outside every unit support, and the fourth gives each group's worst-covered location explicit weight. These are soft penalties on chart-normalized distances at cached reference samples, not a hard constraint on the continuous boundary. Training reduces these losses using the configured object weights. This objective evaluates the valid cached reference-boundary set, separately from hard-example selection for other boundary losses and from assignment compactness.

A.6 Token-Marginal Bounds and the Uniform-Floor Ablation

For valid tokens, nonnegativity of $\widetilde{\mu}_i$ gives $\mu_{in} \geq (1-\lambda_{\max})u_{in}$. Since both u_i and $\widetilde{\mu}_i$ are probability distributions,

$$\tfrac{1}{2}\|\mu_i - u_i\|_1 = \lambda\, \tfrac{1}{2}\|\widetilde{\mu}_i - u_i\|_1 \leq \lambda.$$

The demand proposal remains bounded even without the explicit mixture floor. Writing $A_0 = \alpha c + L$, the bounded logits give $\widetilde{\mu}_{in}/u_{in} \geq \exp(-2A_0)$ in real arithmetic. Setting $\lambda_{\max} = 1$ therefore removes the mixture's positive uniform term only at full adaptation; it retains clipping, bounded residuals, and the training ramp. This defines the *No explicit uniform floor* intervention. It retains a bounded proposal and differs from Demand-adaptive, whose marginal, transport cost, refinement, and regularization settings are listed below. Table D1 reports the resulting target-mass diagnostics; Section 5.3 relates them to reconstruction.

A.7 Representation Configurations

Table A2 summarizes the four configurations developed for this study. Uniform-local and Demand-adaptive are comparison configurations of our global–local design, not methods imported from prior work. VecMALT names the coverage-constrained configuration. In addition to its different

marginal, Demand-adaptive uses learned content affinity from unnormalized projected dot products, residual bound 0.50, anchor-correction coefficient 0.35, and assignment-compactness weight 0.01. The corresponding VecMALT values are zero content affinity, 0.35, 0.20, and 0.002; both use metric-correction coefficient 0.15. Demand-adaptive weights the importance penalty by p_i, enables demand from the start, and has no reference-boundary loss. VecMALT weights that penalty by u_i, uses the adaptation ramp, and sets reference-boundary loss weight to 0.15. These are configuration comparisons, whereas the six interventions each change one setting of the VecMALT reference.

Table A2: Four representation configurations developed in this work, at matched object-code width. Local configurations share a layout but differ in allocation and localization; the six interventions instead modify individual settings of the VecMALT reference.

Configuration	**Stored representation**	**Allocation and localization**
Global-only	$z_i = g_i$, no local chart records	Shared geometry encoder; global coarse/detail decoding
Uniform-local	Global embedding + K local chart records	$\mu_i = u_i$; geometry-conditioned moments, anisotropic metrics, and bounded refinement
Demand-adaptive	Same local layout at each B	$\mu_{in} \propto p_{in} \exp(\delta_{in})$; learned content affinity and its original refinement/regularization settings
VecMALT	Same local layout at each B	Bounded demand proposal, uniform mixture, balanced spatial transport, and reference-boundary supervision

A.8 Evidence Descriptors and Complete Coordinate Formulas

The following definitions expand the compact main-text construction. Field input is $S_i = \{(x_{ij}, d_{ij}, \beta_{ij}, m_{ij})\}_{j=1}^{N_{s,i}}$: a canonical position, a distance observation, deterministic salience, and a type indicator for point UDF, curve UDF, or polygon SDF. Encoding the three types as one-hot values gives seven input channels. The learned support-grid encoder aggregates these samples into occupied cells of a 32×32 grid; token positions are cell centers. The four-layer cyclic boundary encoder processes 24-dimensional descriptors of coordinates, directions, segment lengths, turning, endpoint, junction, and hole cues, with predecessor, successor, ring, and part indices. Junctions describe within-object connectivity. Both streams produce width-192 tokens, and separate boundary, ring, part, and field pooling yields $e_i \in \mathbb{R}^{768}$. Sampling retains ambient observations while emphasizing boundaries.

For stabilized major and minor radii, let $s_{ik} = \sqrt{s_{ik}^{\mathrm{maj}} s_{ik}^{\mathrm{min}}}$ and $\gamma_{ik} = \frac{1}{2}\log(s_{ik}^{\mathrm{maj}}/s_{ik}^{\mathrm{min}})$. The three stored metric coordinates are

$$q_{ik} = [\log(s_{ik}/s_0),\ \gamma_{ik}\cos(2\theta_{ik}),\ \gamma_{ik}\sin(2\theta_{ik})].$$

The double angle represents an unoriented principal axis; isotropy corresponds to a zero anisotropy vector. The transform in Appendix A.3 gives $\xi_{ikn} = A(q_{ik})(x_{in} - a_{ik})$ for local encoding. The refiner's last correction layer initially outputs zero. After the final moment update, reassignment precedes initial encoding; one frame correction is followed by another reassignment, re-encoding, and projection. The unindexed final frame is therefore not constrained to equal the centroid of the final assignment.

The compact-support kernel used in decoding is

$$\kappa(r) = \begin{cases} (1-r)^2(1+2r), & 0 \le r < 1, \\ 0, & r \ge 1. \end{cases}$$

A chart's support is $\Omega_{ik} = \{x : r_{ik}(x) < 1\}$. The ideal blending weights normalize kernels where their sum is positive; the implementation uses the denominator safeguard stated in Appendix A.3. These query-dependent weights differ from the evidence-assignment weights. Both global detail and local residuals are gated, as in Eq. (6). Coarse and detail fields use the feature pyramid generated from g_i, rather than separately stored feature planes. Coordinate restoration follows canonical polygon or curve extraction; shape conditioning remains the stored code, with query position and field type specifying its evaluation.

B Evaluation Details

B.1 Sampled Validation and Dense Reconstruction

The four encoder–decoder configurations use the same extraction resolution, zero contour-refinement steps, object lists, and family-specific settings. Boundary-distance evaluation uses 2,048 samples for polygons and 1,024 for curves; curve extraction uses a two-pixel distance band. Appendix J specifies the common five-method comparison and the Geo2Vec extraction differences.

Vector polygon IoU is $|\widehat{G}_i \cap G_i|/|\widehat{G}_i \cup G_i|$, computed from reference and extracted polygons in the same frame. Occupancy-grid IoU is a separate evaluator diagnostic. Although polygon intersection is evaluated in vector form, the extracted contours retain the discretization error of the field-query grid.

B.2 Distance Metrics and Aggregation

For sampled reference and predicted boundaries, let $\mathcal{D}_{\mathrm{ref}\to\mathrm{pred}}$ and $\mathcal{D}_{\mathrm{pred}\to\mathrm{ref}}$ be the two lists of nearest-neighbor distances. The evaluator uses unsquared symmetric Chamfer distance and a pooled bidirectional 95th percentile:

$$\mathrm{CD}_i = \tfrac{1}{2}\left(\operatorname{mean}\mathcal{D}_{\mathrm{ref}\to\mathrm{pred}} + \operatorname{mean}\mathcal{D}_{\mathrm{pred}\to\mathrm{ref}}\right),$$
$$\mathrm{HD95}_i = Q_{0.95}(\operatorname{concat}[\mathcal{D}_{\mathrm{ref}\to\mathrm{pred}}, \mathcal{D}_{\mathrm{pred}\to\mathrm{ref}}]).$$

Mean and median HD95 aggregate the per-object statistic above. Multiplying canonical distances by each object's metric scale gives distances in meters; large objects can consequently dominate the mean. Reference-complexity tiers use the calibration fixed on training data. Exact fingerprint separation detects identical canonical geometries, but does not establish independence of near duplicates or nearby objects.

All eligible objects contribute to polygon IoU, including empty predictions with IoU zero. An empty boundary has no finite boundary distance. At $B = 512$, Uniform-local and Demand-adaptive each have one such external prediction, as do Independent transport and No refinement among the interventions. Their polygon-distance means use 39,472 objects; the other configurations use 39,473. Paired distance analyses use the finite intersection, and Table C2 marks exceptions across the full width sweep.

Figure 4 intervals use 2,000 bootstrap draws with seed 42. Centroid-based 2-km spatial blocks are resampled within each source file, preserving observed source weights and object pairing. These exploratory intervals condition on the fitted models and observed domains; they neither estimate training-seed variability nor adjust for multiple comparisons.

B.3 Dataset and Source Composition

Building polygons, land-use polygons, and transportation curves provide complementary geometry families. Deterministic object hashing partitions the selected 395,356 objects into training and internal validation as listed in Table B1. No exact canonical-geometry fingerprint overlaps the training and external sets. The external polygons comprise 32,373 Delhi buildings and 7,100 São Paulo land-use objects; the curves comprise 14,903 Cairo segments and three São Paulo objects. Only 17 external polygons have holes (21 holes total), and three are multipart. This composition motivates separate complexity-stratified reconstruction results, while the small hole and multipart counts limit population-level conclusions about complex topology. The area/window subset is described separately in Appendix H.1.

Table B1: Objects used for representation training, internal validation, and external evaluation. Training and internal validation partition the same selected corpus.

Partition	Polygons	Curves	Total objects
Training	250,280	105,540	355,820
Internal validation	28,727	10,809	39,536
External evaluation	39,473	14,906	54,379

Table B2: Geographic sources and feature types. Geographic region, source theme, and polygon/curve geometry family are distinct attributes; their combinations are not independently crossed in this corpus.

Partition	Geographic source	Feature type	External objects by geometry family
Training / internal	Beijing; Tokyo	building	—
Training / internal	London; New York; Tokyo	land_use	—
Training / internal	Singapore; Sydney	segment	—
External	Delhi	building	32,373 polygons
External	São Paulo	land_use	7,100 polygons; 3 curves
External	Cairo	segment	14,903 curves

C Complete Reconstruction Results

Tables C1–C3 report the common complexity breakdown, all 20 capacity configurations, and all six interventions. The five-method $B = 512$ breakdown uses one reference-complexity registry for every method. Table C1 isolates the complexity strata behind the aggregate gains, Table C2 retains the complete width comparison, and Table C3 pairs internal and external intervention results.

Table C1: Polygon IoU (%) at $B = 512$ within common reference-complexity strata. Strata use the training-frozen registry for all five methods, including Geo2Vec, rather than its separately prepared corpus labels. Bold denotes the highest IoU within each complexity stratum.

Method	**Simple (n = 14,795)**	**Moderate (n = 14,201)**	**Complex (n = 7,100)**	**Extreme (n = 3,377)**
Geo2Vec (adapted)	99.639	99.144	97.654	92.913
Global-only	99.891	99.692	98.281	91.233
Uniform-local	**99.903**	**99.735**	99.080	**96.204**
Demand-adaptive	99.855	99.719	**99.104**	95.766
VecMALT	99.901	99.733	99.089	96.192

Table C2: All 20 capacity configurations. IoU is percent, distance columns are meters, and endpoint F1 is a fraction. Mean and median HD95 aggregate the same per-object statistic. † One missing external polygon boundary: 39,472 finite distances, versus 39,473 in other rows. IoU retains all polygons. These are the original capacity-study exports. Bold denotes the best unrounded value per metric at each code width.

B	**Method**	**Internal polygon IoU (%) ↑**	**External polygon IoU (%) ↑**	**External polygon HD95 mean (m) ↓**	**External polygon HD95 median (m) ↓**	**External curve CD (m) ↓**	**External endpoint F1 ↑**
128	Global-only	98.0134	98.7880	0.7880	0.0090	**0.3618**	**0.9920**
128	Uniform-local	98.3116	98.9848	0.7272	0.0091	0.4025	0.9899
128	Demand-adaptive	94.9046	96.5422	1.6859	0.0107	0.7342	0.9396
128	VecMALT	**98.4029**	**99.0493**	**0.6578**†	**0.0090**	0.3749	0.9901
256	Global-only	97.9588	98.7557	0.8328†	0.0093	**0.3712**	**0.9917**
256	Uniform-local	98.6888	99.2015	0.6320†	**0.0090**	0.3849	0.9892
256	Demand-adaptive	98.4352	99.0688	**0.5312**†	0.0092	0.4311	0.9864
256	VecMALT	**98.7237**	**99.2228**	0.6729	0.0091	0.3908	0.9878
512	Global-only	98.0143	98.7891	0.7714	0.0090	**0.3622**	**0.9928**
512	Uniform-local	98.9936	**99.3783**	0.6517†	**0.0090**	0.4109	0.9885
512	Demand-adaptive	98.8761	99.3210	**0.3683**†	0.0094	0.4988	0.9830
512	VecMALT	**98.9945**	99.3772	0.5980	0.0090	0.4060	0.9885
768	Global-only	97.9639	98.7584	0.7970	0.0092	**0.3554**	**0.9930**
768	Uniform-local	99.1056	99.4403	0.4286†	0.0092	0.3899	0.9871
768	Demand-adaptive	99.0419	99.4216	**0.3003**	0.0093	0.8937	0.9858
768	VecMALT	**99.1479**	**99.4688**	0.3726	**0.0090**	0.4691	0.9886
1024	Global-only	97.9716	98.7785	0.8552	0.0090	0.3462	**0.9928**
1024	Uniform-local	**99.2262**	**99.5141**	0.5416	**0.0090**	**0.3322**	0.9874
1024	Demand-adaptive	99.0934	99.4448	**0.2760**	0.0093	0.4424	0.9885
1024	VecMALT	99.1852	99.4856	0.3204†	0.0091	0.3811	0.9881

Table C3: Internal and external reconstruction for the reference and six interventions. IoU includes all 28,727 internal or 39,473 external polygons. Internal polygon HD95 uses 28,726 finite distances for Isotropic charts and No refinement and 28,727 otherwise; external HD95 uses 39,472 for Independent transport and No refinement and 39,473 otherwise. Buffered-curve IoU follows the fixed distance-buffer protocol, separately from boundary distance. Bold denotes the best unrounded value per metric within each split.

Configuration	Split	Polygon IoU (%) ↑	Polygon HD95 mean (m) ↓	Buffered-curve IoU (%) ↑	Curve HD95 mean (m) ↓
VecMALT (reference)	internal	98.9945	0.5453	68.273	2.8485
VecMALT (reference)	external	99.3772	0.5980	73.408	0.9684
No learned importance	internal	98.9895	0.5778	68.625	3.0766
No learned importance	external	99.3726	0.4119	73.537	1.0627
Independent transport	internal	98.8756	0.6039	69.027	2.7915
Independent transport	external	99.3045	0.4674	73.724	0.9755
Isotropic charts	internal	**99.0755**	**0.5314**	**70.804**	2.6471
Isotropic charts	external	**99.4291**	0.5542	**74.290**	1.3080
No refinement	internal	98.9161	0.5842	68.858	**2.5122**
No refinement	external	99.3323	0.6171	73.693	**0.8834**
No explicit uniform floor	internal	99.0170	0.5747	68.988	2.7375
No explicit uniform floor	external	99.3881	**0.3241**	73.706	1.0363
No reference-boundary coverage	internal	98.9874	0.5355	68.997	2.7941
No reference-boundary coverage	external	99.3695	0.6016	73.673	0.9139

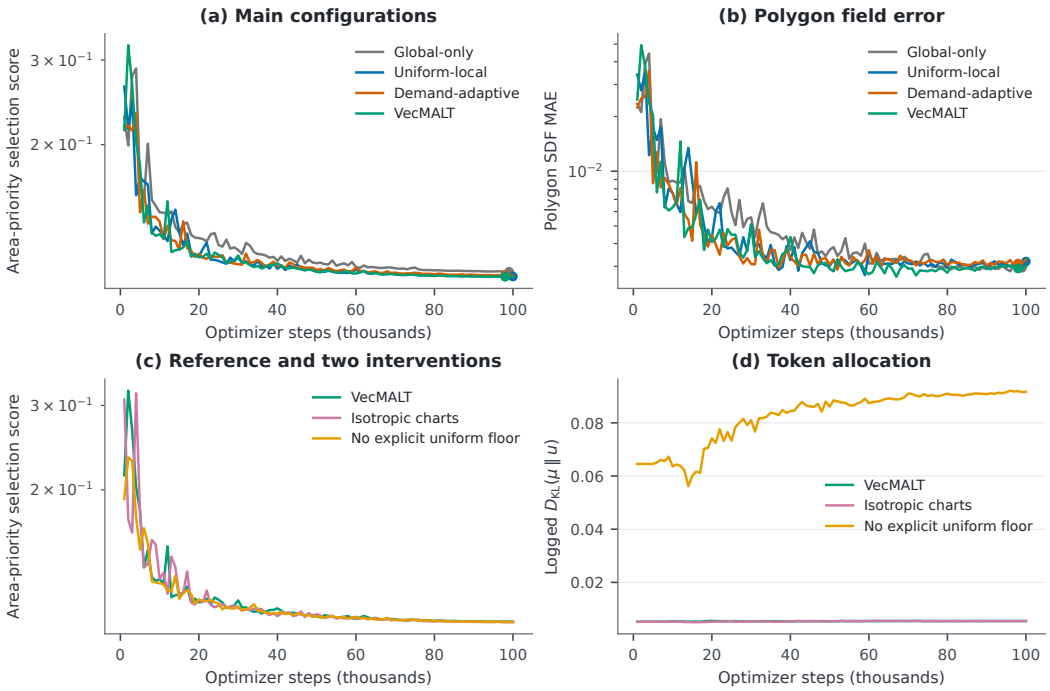


Figure A1: Optimization and allocation diagnostics at $B = 512$. (a) Internal Area-priority selection score for the four configurations; (b) internal polygon SDF error; (c) selection score for the reference and two interventions; (d) token-marginal divergence from uniformity. Dots in (a) and (b) mark selected checkpoints. These sampled-query validation histories are separate from dense reconstruction evaluation.

D Training and Allocation Diagnostics

Table D1: Internal-validation allocation diagnostics at selected checkpoints. Ratios and penalties are aggregated over validation objects, rather than hard corpus-wide minima or external reconstruction errors. "Not measured" denotes a diagnostic skipped when its loss is disabled.

Configuration	**Selected step**	λ	**Mean** $D_{\mathrm{KL}}(\mu\|u)$	**Mean of per-object** $\min_n \mu_n/u_n$	**Reference-boundary coverage penalty**
VecMALT (reference)	98,000	0.25	0.005463	0.9406	0.000019
No learned importance	100,000	0.25	0.005196	0.9542	0.000018
Independent transport	99,000	0.25	0.004031	0.9064	0.031740
Isotropic charts	99,000	0.25	0.005454	0.9389	0.000505
No refinement	99,000	0.25	0.005520	0.9407	0.000021
No explicit uniform floor	99,000	1.00	0.091569	0.6144	0.000041
No reference-boundary coverage	99,000	0.25	0.005442	0.9406	Not measured

Table D1 records target-mass divergence, per-object minimum participation ratios, and the reference-boundary penalty at each selected checkpoint. A mean of per-object minima is not a minimum over the whole corpus. The mass ratios describe the target μ_i; finite transport and row normalization can change realized aggregate assignment mass (Appendix A.3). These diagnostics therefore address allocation behavior, not reconstruction accuracy by themselves.

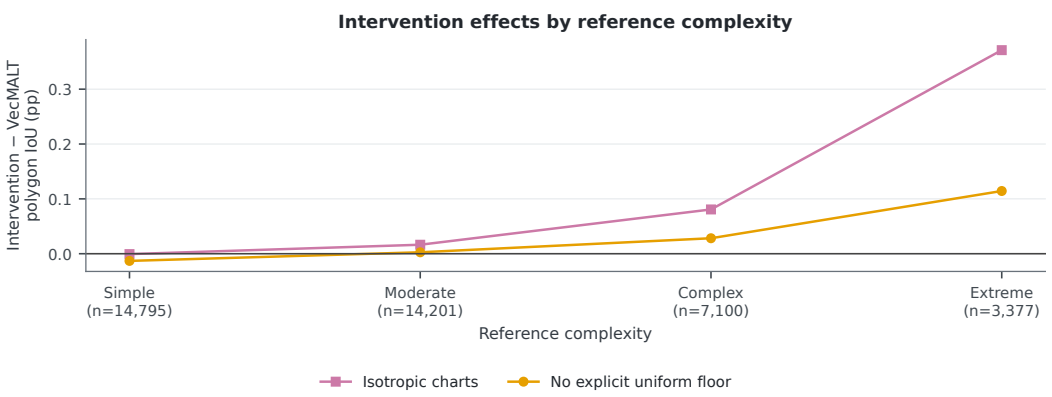


Figure A2: Changes in polygon IoU from two interventions at $B = 512$, stratified by reference complexity. Points show paired mean differences in percentage points, computed as intervention minus VecMALT. Positive values favor the intervention, matching the convention in Table 2. All four training-calibrated strata and their object counts are shown.

The reference remains closer to uniform than the floor requires, while the no-floor intervention permits stronger concentration within its bounded proposal. Independent assignment produces a much larger boundary penalty even with a near-uniform target. Section 5.3 relates those observations to the external reconstruction changes. The penalty uses internal cached reference samples and chart-normalized distances; it is neither the arclength-sampled external gap fraction in Table H3 nor an estimate of that fraction for an unmeasured intervention. Disabling the loss skips this diagnostic, hence "Not measured" in the final row.

Figure A2 reports the complexity-stratified changes for isotropic charts and removal of the explicit uniform floor. Positive values denote higher polygon IoU than the VecMALT reference, following Table 2. Both interventions have their largest gains in the extreme tier, while simple objects show smaller changes that favor the reference.

E TERMINOLOGY AND NOTATION

Table E1: Terminology by computational role: input geometry and evidence, encoded features, allocation, stored records, and downstream reading. Additional allocation symbols are defined below the table.

Term	Symbol	Definition
Vector geometry	G_i	Original input object, before canonicalization
Canonical geometry	$\bar{G}_i$	Geometry in a normalized object-local frame after protocol-specific preprocessing
Geometric evidence	S_i, T_i	Field samples and boundary geometry/connectivity descriptors supplied to the encoder
Field/boundary features	$H_{\text{field}}, H_{\text{boundary}}$	Encoded features from the two evidence streams; combined with positions to form $\mathcal{C}_i$
Evidence token	(x_{in}, h_{in})	A spatial position with an encoded feature; $\mathcal{C}_i$ collects valid tokens
Valid evidence token	$n = 1, \ldots, M_i$	A token retained by the validity mask, excluding padding
Global evidence summary	e_i	Pooled boundary, ring, part, and field features before code projection
Token marginal	μ_i	Target participation mass over valid evidence tokens
Uniform token marginal	$u_{in} = 1/M_i$	Equal target participation per valid token
Bounded demand proposal	$\widetilde{\mu}_i$	Normalized proposal from clipped prior contrast and bounded learned residual
Transport plan	Π_i^*	Ideal chart–token allocation; $\widehat{\Pi}_i$ denotes the finite-Sinkhorn approximation
Conditional assignment weights	ω_{ikn}	Row-normalized finite transport weights used for moments and local encoding
Global embedding	g_i	Stored object-level latent feature
Chart anchor	a_{ik}	Location of the local coordinate frame
Chart metric parameters	q_{ik}	Three stored parameters for scale, orientation, and anisotropy; $Q(q) = A(q)^\top A(q)$ is the metric matrix
Local state / local embedding	v_{ik} / ℓ_{ik}	Intermediate encoder state / its final projection stored in the code
Local chart record	$(a_{ik}, q_{ik}, \ell_{ik})$	Stored geometry and learned local feature
Chart support region	$\Omega_{ik} = \{x : r_{ik}(x) < 1\}$	Spatial region in which its compact-support decoder contribution can be nonzero
Decoder blending weights	$\bar{\omega}_{ik}(x)$	Normalized compact-support weights at query x, distinct from evidence assignments
Structured code	$z_i \in \mathbb{R}^B$	Global embedding plus local chart records; Global-only uses $z_i = g_i$
Pose record	P_i	World placement and coordinate-transform metadata; T_{P_i} denotes the associated map
Structural header	H_i^{str}	Geometry family, cardinality, and assembly metadata
Code width	B	Number of stored scalar coordinates in z_i, excluding separately reported side information
Chart count	K	Fixed number of local records within a specified local operating point

Continued on next page

Table E1 (continued)

Term	Symbol	Definition
Decoder coverage gate	$c_i(x)$	Capped sum of compact-support kernels for a local configuration
Canonical polygon area	A_i	Area of the reference canonical polygon; not reconstruction IoU.
Query window	$W = (c_x, c_y, w, h)$	Canonical rectangle supplied to the reader.
Window occupancy	$\rho_i(W)$	Intersection area divided by rectangle area.
Independent readout head	h_A, h_W	Small predictor trained after code freezing; not the original field decoder.
Flat-code MLP	—	Independent reader that treats the complete code as one vector
Code-chunk attention	—	Attention over nonspatial chunks of the same frozen vector; not encoder evidence tokens
Chart-record attention	—	Window/object-query attention over the global token and complete local records; the revised Shape classifier instead queries local records from the global feature
Address–feature binding	$(a_{ik}, q_{ik}) \leftrightarrow \ell_{ik}$	Feature–frame pairing: a local embedding associated with its stored anchor and metric
Public Edge target	—	Number of stored exterior/interior boundary segments; excludes repeated closing vertices and retains collinear vertices

Allocation symbols. p_i is the normalized geometric prior, η_i its clipped log contrast to uniform mass, and δ_i the learned bounded residual. Their controls are prior strength α, clipping limit c, residual limit L, and mixing coefficient $\lambda(t)$. Transport temperature is τ; Π_i^* is the ideal solution and $\widehat{\Pi}_i$ its finite-iteration approximation. $\Sigma_{ik}^{(r)}$ is a localization covariance, $r_{ik}(x)$ a chart-normalized radius, and sg the stop-gradient operator. The boundary-loss margin ρ_{cov} differs from window occupancy $\rho_i(W)$. The iteration index r is used only on localization states; unindexed (a_{ik}, q_{ik}) denotes the final stored frame. Throughout the paper, *participation* refers to target token mass, *support* to the region of a chart's decoder contribution, and *readout* to prediction from a frozen code. These terms describe different stages rather than interchangeable notions of coverage.

Code and operator conventions. The global and local feature widths are d_g and d_ℓ; $\mathrm{vec}_{k=1}^{K}$ concatenates K chart records. f_{imp} predicts a scalar from a token feature, and t indexes training steps. The plans $\Pi_i^*, \widehat{\Pi}_i$ and cost D_i have shape $K \times M_i$, $\langle \Pi_i, D_i \rangle = \sum_{k,n} \Pi_{ikn} D_{ikn}$, and $\mathbf{1}_d$ is the all-ones vector of length d. $A(q)$ maps offsets into chart-relative coordinates; $Q(q) = A(q)^\top A(q)$ is a metric, not the covariance Σ. The clipping constant c differs from the query-dependent gate $c_i(x)$.

Reader conventions. In $W = (c_x, c_y, w, h)$, (c_x, c_y) is the center and $w, h > 0$ are width and height. A_i is polygon area, distinct from the frame transform $A(q)$. The permutation π_i reassigns local features to fixed chart frames. In the name VecMALT, "local tokens" means stored chart records presented to readers, not the encoder evidence tokens (x_{in}, h_{in}).

F Qualitative Reconstruction and Chart Coverage

F.1 Object Selection and Visualization

Figure 6 and Figures A4–A8 use the fixed 20-object display set and its original six-object main subset, chosen before adding Geo2Vec. All outputs come from the checkpoints and test codes used in the common evaluation. Stable object identities and canonical references align the methods. The saved vector predictions are displayed without smoothing, component filtering, selected-object refitting, or new field queries. Curve extraction follows each method's evaluation pipeline (Appendix J.1).

The display set contains five near-exact, five high-overlap, and five more difficult VecMALT polygon reconstructions, plus five high-overlap curves. Selection uses VecMALT scores and geometric features to illustrate recesses, concavities, holes, unequal parts, and localized bends. These display groups differ from the reference-complexity tiers in Table C1 and are not a random sample for estimating population performance.

All four encoder–decoder models use $B = 512$, a 512×512 field grid, zero contour refinement, and the same family-specific extraction settings. Each local configuration has 24 records. Whole-object panels include all predicted components; close-ups use the same subregion for every method. Black solid lines denote reference boundaries, blue dashed lines denote reconstructions, and pale fills aid polygon visualization.

The six rows of Figure 6 correspond to object IDs 43097, 50700, 24158, 49924, 50622, and 4132. Table F1 reports IoU recomputed from the five displayed polygon predictions. The curve has no polygon-IoU score. Full-cohort results remain those of the complete evaluation runs.

Table F1: IoU (%) of the five polygon examples in Figure 6, recomputed from the displayed canonical predictions. IDs link the table to the fixed display set; they are not model inputs. Bold denotes the highest IoU for each example.

Example (object ID)	**Geo2Vec**	**Global-only**	**Uniform-local**	**Demand-adaptive**	**VecMALT**
Multiple shallow recesses (43097)	98.82	97.78	99.25	99.32	**99.42**
Irregular concave outline (50700)	94.33	93.00	97.64	97.22	**98.02**
Two interior holes (24158)	90.51	72.30	**93.57**	80.26	93.23
Complex land-use boundary (49924)	95.54	86.60	94.77	94.07	**96.14**
Unequal disconnected parts (50622)	93.58	81.17	94.24	**96.23**	96.01

F.2 Sampled Chart Coverage

For each reference exterior ring, hole, or line component, we place 1,024 samples at equal arclength intervals. Rings exclude the duplicate endpoint; open lines include both endpoints. The samples are constructed from the canonical reference geometry, rather than reused from encoder tokens or weighted training targets. With J_{ig} indexing samples b_{ij} in group g, the reported uncovered fraction is

$$\widehat{U}_{ig} = \frac{1}{|J_{ig}|} \sum_{j \in J_{ig}} \mathbf{1}\left[\min_{1 \le k \le K} r_{ik}(b_{ij}) \ge 1 \right].$$

The chart-normalized radius r_{ik} defines the decoder support and is dimensionless. This diagnostic measures spatial reach, not boundary-reconstruction distance. Zero sampled gaps leave coverage between samples untested, and points just inside a support can still have small kernel weights.

Table F2: Uncovered reference samples (%) in the two-hole example of Figure A3. Each group has 1,024 samples. H1 is the right-hand hole and H2 the left-hand hole in the displayed frame. Bold denotes the lowest uncovered fraction in each row; exact ties are all bold.

Reference group	Uniform-local	Demand-adaptive	VecMALT
Outer ring	**0.00**	**0.00**	**0.00**
Hole H1	**0.00**	6.15	**0.00**
Hole H2	**0.00**	6.84	**0.00**

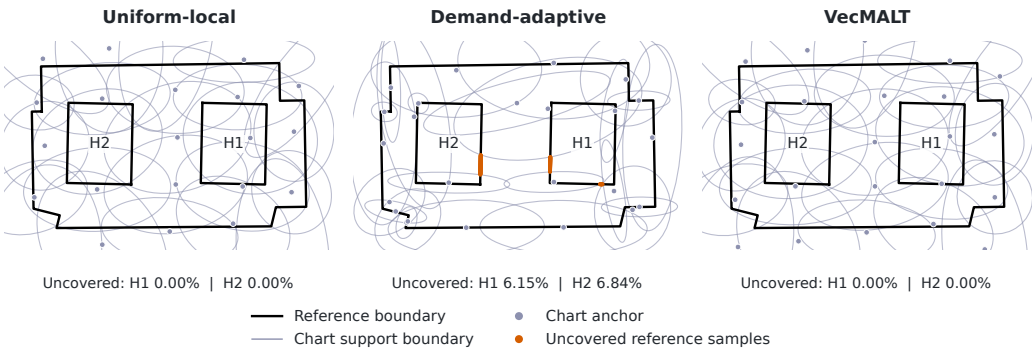


Figure A3: Final chart supports for the two-hole example in Figure 6, with 24 records per local model. Black curves show reference boundaries; pale ellipses and dots show supports and anchors. Orange-red points lie outside every support. Each ring has 1,024 arclength samples. H1 is the right-hand hole and H2 the left-hand hole; Table F2 reports the uncovered fractions.

Demand-adaptive leaves 63 of 1,024 samples uncovered on H1 and 70 on H2. Uniform-local and VecMALT cover all sampled points on both holes and the outer ring, yet differ in polygon IoU. The two-hole case thus compares fitting quality after both configurations reach the sampled boundary; it is not evidence that support alone fixes the contour. Figure A3 uses the stored final anchors and metrics to draw supports and highlights only samples meeting the uncovered rule.

F.3 Twenty-Object Reconstruction Gallery

Figures A4–A7 show the four fixed display groups described in Appendix F.1: near-exact polygons, high-overlap polygons, difficult polygons, and curves. Each row uses the same object and view across methods. Figure A8 enlarges the curve region marked in the last row of Figure 6.

F.4 Curve Reconstruction Close-Up

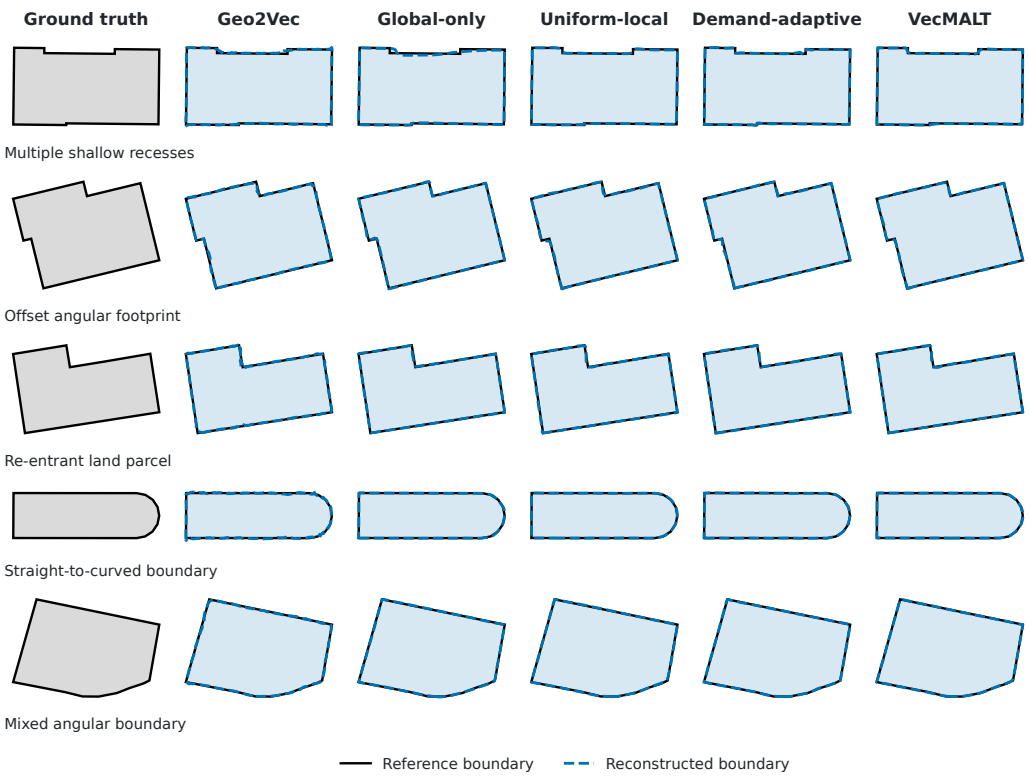


Figure A4: Five near-exact VecMALT polygon reconstructions, with the same objects shown for all methods. Rows are IDs 43097, 44741, 52746, 51439, and 47754, illustrating recesses, stepped outlines, and corners. Column order and line conventions follow Figure 6; Appendix F.1 describes the selection.

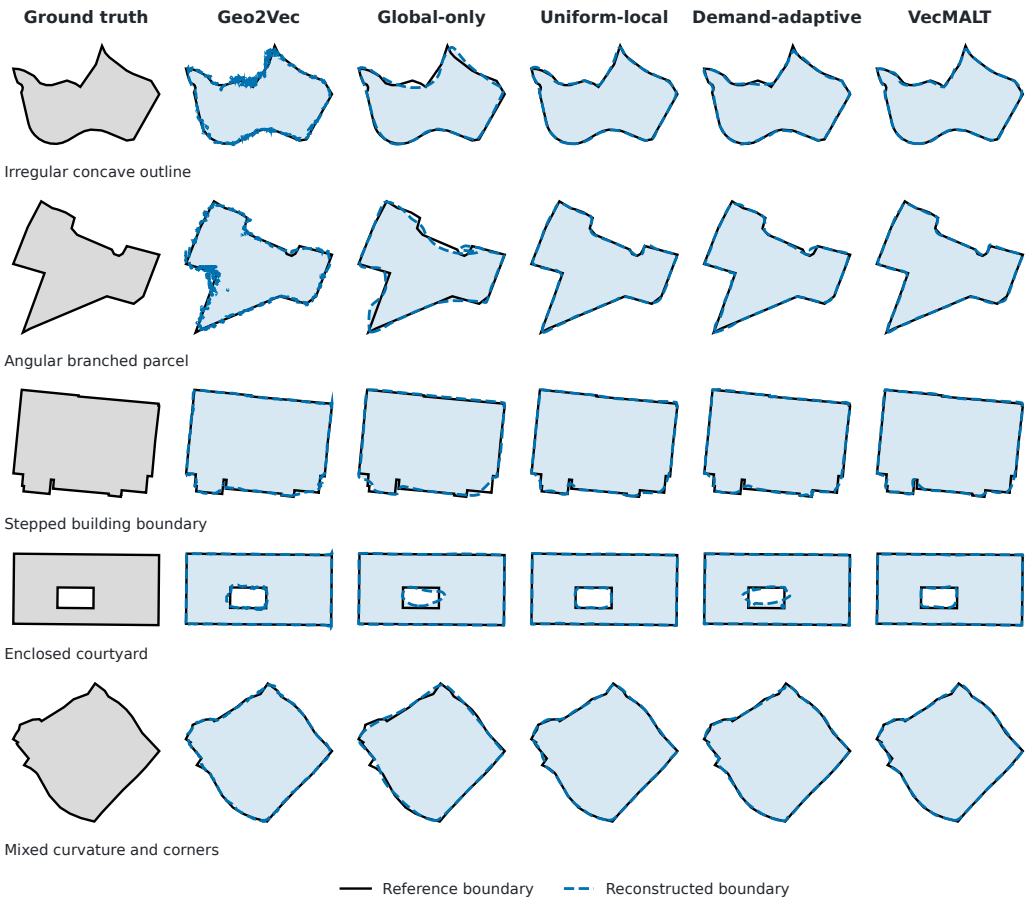


Figure A5: High-overlap polygon examples, with IDs 50700, 49302, 19919, 21937, and 54272. Irregular outlines, branches, steps, a courtyard, and mixed curvature reveal differences in boundary position and interior structure. Column order and line conventions follow Figure 6.

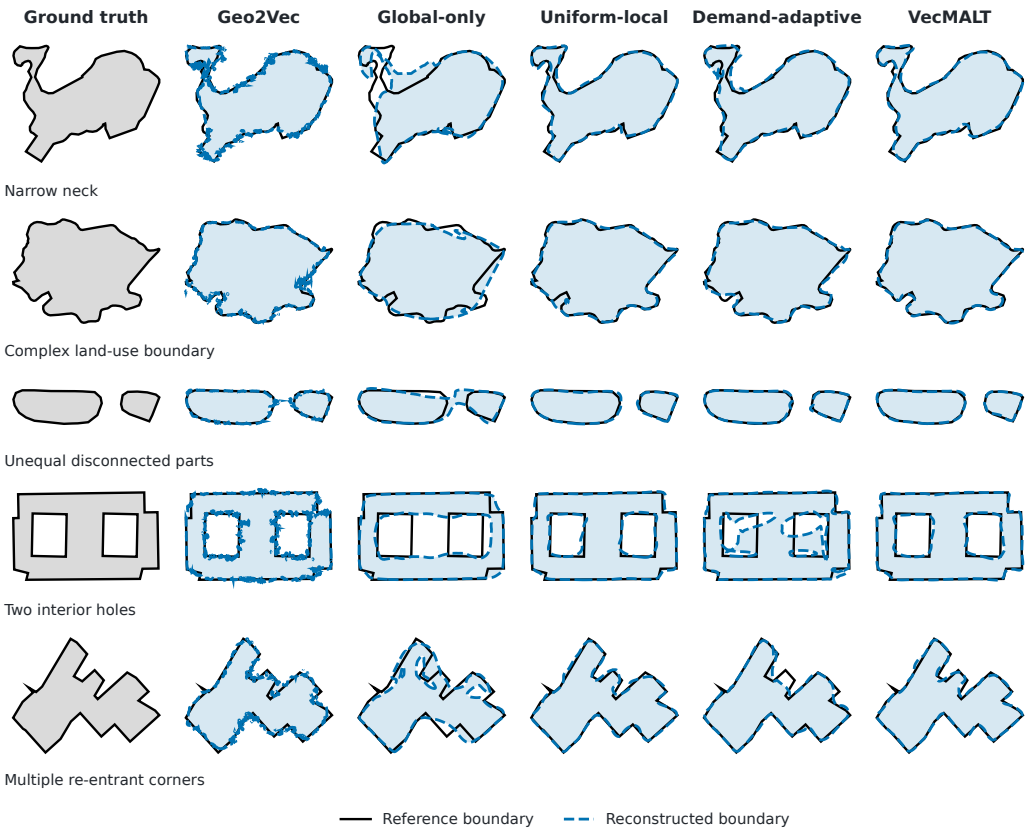


Figure A6: Difficult polygon examples, with IDs 47530, 49924, 50622, 24158, and 45578. The shared views reveal contour errors, changes in holes, and incorrect connections between parts. Column order and line conventions follow Figure 6; all saved components are retained.

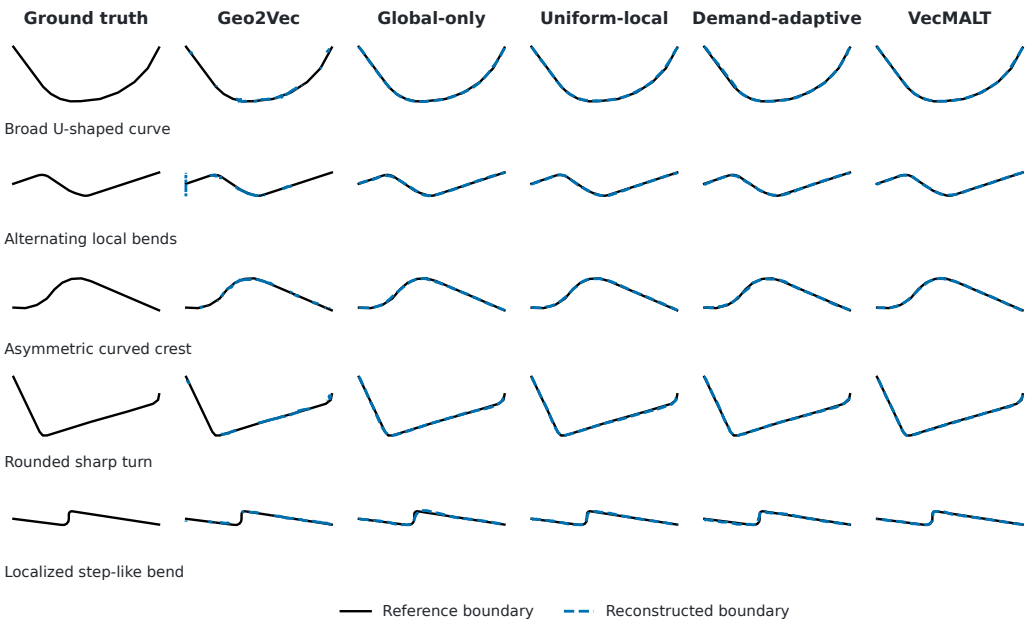


Figure A7: Curve examples selected by high VecMALT buffered-curve IoU, with IDs 2342, 14225, 8773, 2817, and 4132. References and predictions often overlap at this scale; Figure A8 enlarges the final example. Column order and line conventions follow Figure 6.

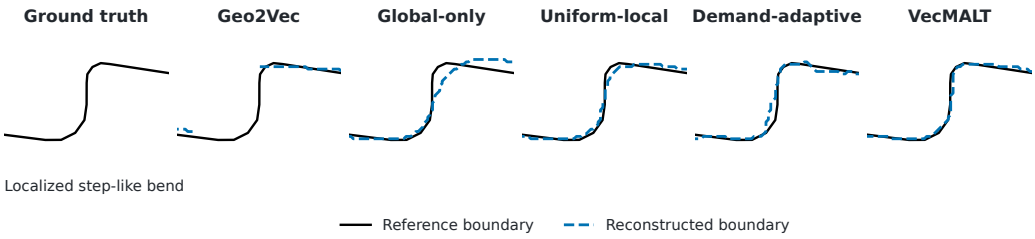


Figure A8: Shared close-up of object 4132 from the final row of Figure 6, using $x \in [-0.22, -0.02]$, $y \in [-0.08, 0.08]$ for every method. The same saved predictions are shown. Geo2Vec has a gap across the turn where the black reference remains visible. Column order and line conventions follow Figure 6.

G Frozen-Code Probes, Retrieval, and Efficiency

G.1 Geometric Probes

The property probes test what is accessible through a linear readout, complementing the nonlinear area/window readers in Appendix H. For VecMALT, adding complete local records to its 128-dimensional global embedding improves all six reported polygon-property R^2 values. This is a within-model comparison with different input widths; the full-code comparison across methods holds width at 512.

We freeze the four $B = 512$ checkpoints selected by internal Area-priority validation. Local codes contain a 128-dimensional global embedding and 24 complete 16-scalar records, whereas Global-only stores a 512-dimensional global embedding. Both complete-code and global-component results are reported with their actual readout widths.

All methods use the same stable-hash sample, stratified by geometry family, source, and reference-complexity tier. It contains 50,000 polygons and 50,000 curves for training, 10,000 of each for validation, and 20,000 polygons plus all 14,906 external curves for testing. These are subsets of the reconstruction splits. For each family and target, we fit feature standardization on training data and select ridge regularization from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$ on internal validation. Targets are computed from canonical reference geometry.

Table G1 includes a seven-input metadata reference: log-transformed vertex, unique-vertex, hole, and part counts, plus three training-calibrated dimension-specific complexity values. It measures information accessible from geometry metadata. Table G2 reports the global-component probes. Both tables retain all evaluated properties, including hole-area fraction, for which the external population contains few positive examples.

G.2 Object and Local-Chart Retrieval

Queries, galleries, patches, and geometric relevance are fixed before learned descriptors are evaluated. For each geometry family, object retrieval uses 128 queries and a gallery of 1,024 objects. Local retrieval selects 256 reference-motif queries and 2,048 gallery locations from a 1,024-object pool. The 512 local queries span 436 objects. Candidates must share the query's geometry family; same-object matches are excluded, while same-region matches are allowed. All queries have enough eligible candidates.

Object relevance uses symmetric Chamfer distance between 96 arclength samples of canonical reference geometry. Local relevance uses 96 samples from a reference-centered patch of radius 0.12, expressed in a fixed reference PCA frame. Reference motifs include concave and sharp convex corners, high-curvature polygon and curve locations, hole boundaries, and open-line endpoints. The ten nearest eligible candidates in this geometric distance define binary relevance for Recall@10, mAP@10, and nDCG@10. Distance Spearman compares geometric and embedding-distance rankings over the eligible gallery.

Complete-code retrieval uses cosine distance after coordinate-wise standardization fitted on at most 50,000 training objects per family. Global-component retrieval uses raw cosine distance. These are different readout protocols, not an isolated normalization comparison. At each reference location, local retrieval selects the nearest chart in elliptical distance and compares its stored 11-dimensional ℓ_{ik} by raw cosine distance. Uncovered locations remain in the evaluation. Global-only has no local record and is marked inapplicable.

Global components provide useful object descriptors, but the complete-code and local-feature rankings differ. Demand-adaptive gives the highest local top-ten scores, while VecMALT gives the highest local distance Spearman. Raw local nearest-neighbor retrieval therefore gives a different ranking from the window reader. A plausible interpretation is that a feature learned with its frame and object context can support reconstruction without becoming a strong stand-alone similarity descriptor. These models have no explicit cross-object metric-learning objective; the retrieval experiment tests that descriptor directly rather than the learned record-reading interface.

At the selected motifs, all 512 query locations lie inside a support for Uniform-local and VecMALT, compared with 511 for Demand-adaptive. This measures availability at the chosen locations, not

Table G1: Complete-code geometric probes, reported as external $R^2 \uparrow$ / MAE $\downarrow$. Polygon targets use 20,000 test objects; curve targets use 14,906. All learned inputs have width 512. The seven-input count/complexity reference is metadata, not a matched-width model. MAE remains in each target's own units and is not averaged across targets. Bold marks the highest R^2 and lowest MAE separately in each row, including the metadata reference.

Property	Global-only	Uniform-local	Demand-adaptive	VecMALT	Count / complexity (7)
Area compactness	0.9915 / 0.0095	**0.9933** / **0.0093**	0.9874 / 0.0118	0.9928 / 0.0097	0.7312 / 0.0585
Area convexity	**0.9465** / **0.0138**	0.9441 / 0.0165	0.9359 / 0.0163	0.9412 / 0.0167	0.4047 / 0.0512
Area rectangularity	**0.9733** / **0.0172**	0.9712 / 0.0200	0.9646 / 0.0208	0.9676 / 0.0210	0.4974 / 0.0902
Area log elongation	0.9804 / **0.0393**	0.9794 / 0.0452	0.9773 / 0.0424	**0.9808** / 0.0417	0.7315 / 0.1736
Area boundary turning entropy	0.6064 / 0.0521	0.5024 / 0.0612	**0.6452** / **0.0518**	0.5199 / 0.0603	0.0993 / 0.0994
Area hole fraction	**0.3691** / 0.0056	0.1596 / 0.0070	0.3459 / 0.0057	0.2269 / 0.0067	0.3482 / **0.0009**
Curve log sinuosity	**0.9288** / **0.0103**	0.8646 / 0.0264	0.9112 / 0.0139	0.8614 / 0.0273	-1.4998 / 0.1457
Curve maximum absolute turn	**0.8827** / **0.0327**	0.8578 / 0.0506	0.8715 / 0.0335	0.8595 / 0.0476	0.2917 / 0.1741
Curve mean absolute turn	**0.9358** / **0.0178**	0.9114 / 0.0281	0.9242 / 0.0217	0.9068 / 0.0290	0.5352 / 0.0748
Curve normalized total length	**0.9867** / **0.0044**	0.9677 / 0.0131	0.9800 / 0.0067	0.9678 / 0.0128	-0.8709 / 0.1263
Curve turning entropy	0.7526 / 0.0206	0.7307 / 0.0261	**0.7670** / **0.0197**	0.7290 / 0.0253	0.2139 / 0.0721

Table G2: Global-component probe R^2 on the external set. Global-only supplies 512 coordinates; local models supply 128. Table G1 provides the matched-width complete-code comparison. Bold denotes the highest unrounded value in each row.

Property	Global-only g (512-D)	Uniform-local g (128-D)	Demand-adaptive g (128-D)	VecMALT g (128-D)
Area compactness	**0.9915**	0.9776	0.9694	0.9781
Area convexity	**0.9465**	0.8802	0.8577	0.8882
Area rectangularity	**0.9733**	0.9493	0.9409	0.9454
Area log elongation	**0.9804**	0.9711	0.9689	0.9733
Area boundary turning entropy	**0.6064**	0.4380	0.5486	0.4688
Area hole fraction	**0.3691**	0.0360	0.0339	0.1031
Curve log sinuosity	**0.9288**	0.8534	0.8773	0.8801
Curve maximum absolute turn	**0.8827**	0.8609	0.8460	0.8601
Curve mean absolute turn	**0.9358**	0.9095	0.8786	0.9092
Curve normalized total length	**0.9867**	0.9722	0.9732	0.9731
Curve turning entropy	**0.7526**	0.7231	0.7202	0.7291

Table G3: Object retrieval on 256 common external queries; all metrics ↑. Complete-code rows use training-standardized cosine at width 512. Global-component rows use raw cosine at the stated widths. No retrieval adapter is trained. Bold denotes the best value per metric within each readout protocol (full or global).

Method	Readout and distance	Width	nDCG@10	Recall@10	mAP@10	Distance Spearman
Global-only	Full z, standardized cosine	512	**0.5027**	**0.4523**	**0.3629**	**0.7294**
Global-only	Global g, raw cosine	512	0.5297	0.4809	0.3915	**0.8059**
Uniform-local	Full z, standardized cosine	512	0.3360	0.3023	0.2023	0.6881
Uniform-local	Global g, raw cosine	128	**0.5368**	**0.4930**	**0.4027**	0.7611
Demand-adaptive	Full z, standardized cosine	512	0.3691	0.3305	0.2400	0.5634
Demand-adaptive	Global g, raw cosine	128	0.4848	0.4402	0.3516	0.7412
VecMALT	Full z, standardized cosine	512	0.3445	0.3141	0.2087	0.6806
VecMALT	Global g, raw cosine	128	0.5274	0.4848	0.3915	0.7750

Table G4: Raw local-feature retrieval on 512 common reference queries, using 11-dimensional local embeddings; all metrics ↑. Global-only has no chart record. These scores measure nearest-neighbor similarity, not reconstructed patch accuracy or motif classification. Bold denotes the best value in each metric column.

Method	nDCG@10	Recall@10	mAP@10	Distance Spearman
Global-only	—	—	—	—
Uniform-local	0.0404	0.0340	0.0150	0.1116
Demand-adaptive	**0.0569**	**0.0480**	**0.0232**	0.0665
VecMALT	0.0346	0.0318	0.0125	**0.1250**

population-level boundary coverage or retrieval discrimination. Top-ten relevance can also distinguish nearly equivalent shapes. Table G5 reports the complete motif breakdown, including the small hole-boundary subset.

Table G5: Local retrieval by reference motif, measured by nDCG@10 ↑. Query locations are shared across methods; the 512 locations need not identify distinct objects or chart records. Bold denotes the best score for each motif.

Reference motif	Queries	Uniform-local	Demand-adaptive	VecMALT
Concave corner	69	0.0278	**0.0434**	0.0222
Sharp convex corner	88	0.0282	**0.0535**	0.0235
High-curvature area boundary	88	0.0187	**0.0362**	0.0172
High-curvature curve	92	0.1002	**0.1218**	0.0856
Hole boundary	11	0.0000	**0.0313**	0.0186
Open-line endpoint	164	0.0331	**0.0408**	0.0275

G.3 Runtime, Memory, and Storage

Profiling uses an NVIDIA A100-SXM4-80GB, PyTorch 2.5.1/CUDA 12.1, FP32, and one CPU thread. At each encoding batch size, ten preloaded batches receive three warm-up calls and five timed repetitions. A scalar-field microbenchmark evaluates 65,536 queries on one representative polygon with decoder memory prepared. Vector reconstruction uses 24 common objects, with one warm-up and one timed call per object and stage. The field grid is 512×512 over $[-0.75, 0.75]^2$; query chunks contain at most 65,536 points, curve extraction uses a two-pixel band, and polygon contour refinement is disabled.

Encoding starts with device-resident evidence and ends with the code. Code-to-vector reconstruction includes decoder-memory and landmark preparation, field queries, extraction, and coordinate restoration; evidence-to-vector also includes encoding. Raw preprocessing, disk I/O, collation, and the initial evidence transfer are excluded. These are separately timed microbenchmarks, not additive stage totals. GPU memory denotes peak allocated tensor memory for each workload.

Table G6: Mean execution time (ms) for separate microbenchmarks. Encoding is per batch; the prepared field benchmark evaluates 65,536 scalar queries. Reconstruction uses 512×512 grids on 24 shared objects. Timings have different start/end points and are not additive. Bold denotes the lowest time in each workload column.

Method	Encode batch 1	Encode batch 8	Encode batch 16	Prepared field query	Code → vector	Prepared evidence → vector
Global-only	**3.05**	**4.05**	**5.47**	**14.13**	**98.41**	**103.07**
Uniform-local	17.38	22.68	30.47	58.93	272.04	289.92
Demand-adaptive	18.79	22.55	28.96	58.46	271.36	290.34
VecMALT	18.17	24.62	33.80	58.89	273.68	291.08

Each complete FP32 code contains 2,048 bytes. For a local code, 512 bytes store the global embedding, 480 store chart geometry, and 1,056 store local features. In the measured 54,379-object serialization, indices and IDs add 16 bytes per object, while pose and structural header add 832.3 bytes on average. The mean payload is therefore 2,896.3 bytes. Table G7 amortizes the shared reconstruction decoder over this collection and reports the full encoder–decoder parameter count separately. The retained reconstruction artifact excludes the encoder, optional reference geometry, and evaluation metadata.

These profiles apply to $B = 512$. Figure 4 measures storage–reconstruction trade-offs across widths, but computation was not profiled at every width. At the measured operating point, local fields

Table G7: Network size and measured storage. Parameters count the full encoder–decoder; shared storage is the reconstruction decoder retained once per collection. Payload includes code, IDs, indices, pose, and header. Amortization uses 54,379 objects and excludes optional references and optimizer state. KiB = 2^{10} bytes; MiB = 2^{20} bytes. Bold denotes the smallest recorded resource requirement in each column, not the best reconstruction quality.

Method	**Parameters (10^6)**	**Code / object (KiB)**	**Payload / object (KiB)**	**Shared decoder (MiB)**	**Amortized / object (KiB)**
Global-only	8.040	**2.000**	**2.828**	15.615	3.122
Uniform-local	**7.608**	**2.000**	**2.828**	**12.660**	**3.067**
Demand-adaptive	7.721	**2.000**	**2.828**	**12.660**	**3.067**
VecMALT	7.646	**2.000**	**2.828**	**12.660**	**3.067**

account for most of the additional reconstruction time relative to Global-only; switching among local allocation policies has a smaller effect.

H AREA AND QUERY-WINDOW READOUT

H.1 TARGETS, DATA SPLITS, AND QUERY SAMPLING

Area/window tasks use a fixed-hash subset of the reconstruction splits, separate from the probe subset in Appendix G.1: 50,000 training, 10,000 validation, and 20,000 external-test polygons. Each method uses one frozen checkpoint across its three splits. Stable object keys and canonical-geometry fingerprints align codes and labels. Inputs follow Section 3.4: the selected code view and, for windows, the four rectangle coordinates.

The area target is the area of the repaired canonical polygon, including holes and parts. Window occupancy is the exact intersection area divided by rectangle area, evaluated at the stored float32 window coordinates. Width and height are sampled independently from 0.0625, 0.125, 0.25, and 0.5. Half the centers are uniform among positions that keep the rectangle in the unit box; the other half are sampled by reference-boundary arclength. Boundary-centered rectangles can extend outside the box and are not clipped. The seed and object key fix 16 windows per training object and 64 per validation/test object. All methods share the resulting 800,000/640,000/1,280,000 queries.

The test subset contains 16,403 Delhi building polygons and 3,597 São Paulo land-use polygons. Eight objects contain holes and three are multipart; 14,606 have at most five vertices and 4,523 have 6–16. Most examples therefore have simple topology. Of the test windows, 72.61% are partially occupied, 7.37% empty, and 20.01% full. A training-median constant gives area MAE 0.18357, and a trained window-only MLP gives occupancy MAE 0.18422.

H.2 READER ARCHITECTURES, TRAINING, AND CONTROLS

The area MLP is 512–128–64–1 with SiLU hidden activations and sigmoid output. The window MLP adds four rectangle coordinates, giving input width 516. Chart-record attention projects the 128-dimensional global embedding and each of the 24 complete 16-scalar records to width 64. Local records share one affine map; the window query has a separate map. Four-head attention yields a context, which is concatenated with the query and read by a 128–64–1 SiLU/sigmoid MLP. The stored metric coordinates enter as features, without conversion to the decoder's compact-support mask.

In the external window reader, the affine projections and attention are

$$t_{i0} = U_g g_i + b_g, \qquad t_{ik} = U_\ell[a_{ik}; q_{ik}; \ell_{ik}] + b_\ell,$$
$$v_W = U_W W + b_W, \qquad c_{iW} = \mathrm{MHA}(v_W, \{t_{ik}\}_{k=0}^{K}, \{t_{ik}\}_{k=0}^{K}).$$

The output MLP receives $[c_{iW}; v_W]$. Area and occupancy satisfy

$$A_i = \mathrm{Area}(\bar{G}_i), \qquad \rho_i(W) = \frac{\mathrm{Area}(\bar{G}_i \cap W)}{\mathrm{Area}(W)}, \qquad \widehat{A}_i = h_A(z_i), \quad \widehat{\rho}_i(W) = h_W(z_i, W).$$

The canonical polygon fits a unit box, so $A_i \in (0, 1]$ and $\rho_i(W) \in [0, 1]$. Occupancy accounts for holes and components and is not intersection-over-union.

Code-chunk attention partitions the 512 scalars into 25 chunks of length 21, including 13 padding zeros, then adds trainable slot identifiers to their projections. Padding is internal to the reader. Its attention and output widths match chart-record attention. Standardization uses task-training codes: record readers share statistics for each local coordinate across chart slots, while other views use per-code-coordinate statistics. The interface comparison therefore changes grouping, parameter sharing, and normalization together.

Neural readers use AdamW with learning rate 0.001, weight decay 0.0001, object batches of 256, smooth-L1 loss with transition 0.05, seed 42, and at most 50 epochs with patience eight. Each window batch includes all 16 training windows for its objects. The validation-MAE minimum selects the checkpoint. Ridge strength is selected from $10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100$ by validation MAE. Uniform-local and Demand-adaptive chart readers select epoch 50, and VecMALT selects epoch 47. Results reflect this finite training budget, with the representation and original decoder frozen throughout.

Table H1: Registered reader parameters, shared across objects and separate from the fixed code budget. Counts include parameters registered by the common reader implementation, including any task-unused query parameter.

Reader	Input dimensions	Registered parameters
Area ridge	512	513
Area MLP	512	73,985
Window flat-code MLP	512 + 4 query	74,497
Window code-chunk attention	512 + 4 query	28,353
Window chart-record attention	512 + 4 query	34,689
Window-only MLP	4 query	8,961

The binding control permutes each object's 24 local features relative to its frames using an object-specific deterministic seed. It preserves the global embedding and both sets of frames and features. Every split receives the reassignment before a fresh reader is trained with the same architecture and budget. The permutation need not move every feature. The window-only control receives no code; component views g_i, $[g_i; a_i; q_i]$, and $[g_i; \ell_i]$ use fewer input coordinates and correspondingly smaller MLPs.

H.3 Complete Readout Results and Code Components

Table H2: All 36 external area/window configurations. MAE/RMSE ↓; R^2 ↑. Code input width excludes window coordinates. Component views change input width and head size; complete-code comparisons use 512 coordinates. "Shuffled" denotes feature–frame reassignment in every split followed by reader retraining. Bold compares representations within the same task, head, and component view; ranks use unrounded results. The window-only control has no matched competitor.

Representation	Task / reader / code view	Code input	MAE ↓	RMSE ↓	R^2 ↑	Reader parameters
Geo2Vec (adapted)	Area / MLP / full	512	0.04268	0.05979	0.92433	73,985
Geo2Vec (adapted)	Area / ridge / full	512	0.08824	0.11325	0.72853	513
Geo2Vec (adapted)	Window / flat-code MLP	512	0.09087	0.13958	0.78100	74,497
Geo2Vec (adapted)	Window / code-chunk attention	512	0.10292	0.15099	0.74374	28,353
Global-only	Area / MLP / full	512	0.00232	0.00445	0.99958	73,985
Global-only	Area / ridge / full	512	0.00467	0.00898	0.99829	513
Global-only	Window / flat-code MLP	512	**0.04229**	**0.06741**	**0.94892**	74,497
Global-only	Window / window-only MLP	0	0.18422	0.24571	0.32135	8,961
Global-only	Window / code-chunk attention	512	0.03446	**0.06331**	**0.95495**	28,353
Uniform-local	Area / MLP / full	512	0.00190	0.00361	0.99972	73,985
Uniform-local	Area / MLP / global	128	**0.00225**	**0.00560**	**0.99934**	24,833
Uniform-local	Area / MLP / global + features	392	**0.00205**	**0.00400**	**0.99966**	58,625
Uniform-local	Area / MLP / global + geometry	248	0.00253	0.00526	0.99941	40,193

Continued on next page

Table H2 (continued)

Representation	Task / reader / code view	Code input	MAE ↓	RMSE ↓	R^2 ↑	Reader parameters
Uniform-local	Area / ridge / full	512	0.00450	0.00699	0.99896	513
Uniform-local	Window / flat-code MLP	512	0.04348	0.06855	0.94718	74,497
Uniform-local	Window / chart-record attention	512	0.03105	0.05712	0.96332	34,689
Uniform-local	Window / chart-record attention, shuffled	512	**0.03288**	**0.05945**	**0.96027**	34,689
Uniform-local	Window / code-chunk attention	512	0.03565	0.06340	0.95482	28,353
Demand-adaptive	Area / MLP / full	512	0.00209	0.00476	0.99952	73,985
Demand-adaptive	Area / MLP / global	128	0.00329	0.00715	0.99892	24,833
Demand-adaptive	Area / MLP / global + features	392	0.00246	0.00552	0.99935	58,625
Demand-adaptive	Area / MLP / global + geometry	248	**0.00232**	0.00580	0.99929	40,193
Demand-adaptive	Area / ridge / full	512	0.00550	0.00988	0.99793	513
Demand-adaptive	Window / flat-code MLP	512	0.04627	0.07364	0.93905	74,497
Demand-adaptive	Window / chart-record attention	512	0.03221	0.05965	0.96001	34,689
Demand-adaptive	Window / chart-record attention, shuffled	512	0.03331	0.06094	0.95825	34,689
Demand-adaptive	Window / code-chunk attention	512	**0.03408**	0.06495	0.95257	28,353
VecMALT	Area / MLP / full	512	**0.00181**	**0.00349**	**0.99974**	73,985
VecMALT	Area / MLP / global	128	0.00241	0.00570	0.99931	24,833
VecMALT	Area / MLP / global + features	392	0.00217	0.00431	0.99961	58,625
VecMALT	Area / MLP / global + geometry	248	0.00251	**0.00508**	**0.99945**	40,193
VecMALT	Area / ridge / full	512	**0.00442**	**0.00695**	**0.99898**	513
VecMALT	Window / flat-code MLP	512	0.04487	0.06992	0.94504	74,497
VecMALT	Window / chart-record attention	512	**0.03059**	**0.05615**	**0.96456**	34,689
VecMALT	Window / chart-record attention, shuffled	512	0.03557	0.06163	0.95731	34,689
VecMALT	Window / code-chunk attention	512	0.03698	0.06493	0.95261	28,353

The component views in Table H2 test how the global embedding, frame geometry, and local features contribute to area prediction. VecMALT's complete code gives MAE 0.00181, compared with 0.00241 from the global component, 0.00251 from global plus frame geometry, and 0.00217 from global plus local features. The complete record is most useful for these readers, although compo-

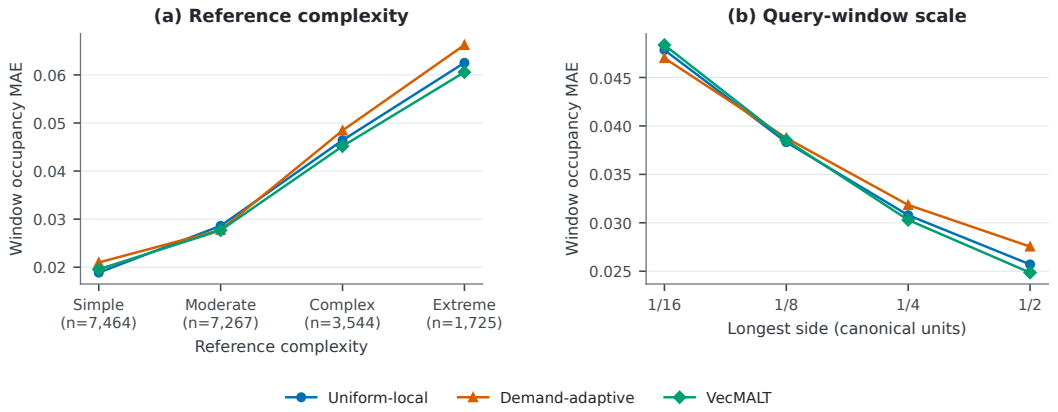


Figure A9: Chart-record window MAE by (a) training-calibrated reference complexity and (b) longest window side. Within each group, errors are averaged over an object's applicable windows, then over objects. The smallest-scale group excludes objects without a sampled window at that scale. Rankings vary with geometry and query size; these are occupancy-prediction errors, not boundary distances.

nent removal also changes input width and head size. The corresponding full-code comparisons hold width at 512 across methods.

The per-object comparisons show how broadly the aggregate gains are distributed. VecMALT's chart-record reader has lower window error than its flat and chunk readers on 94.51% and 87.91% of test objects. Against Uniform-local and Demand-adaptive chart readers, the corresponding win rates are 51.06% and 49.18%, despite lower mean MAE. The intact feature–frame pairing beats reassignment on 85.85% of objects. Thus the same-code interface benefit is more broadly distributed than the smaller differences between local configurations.

H.4 Window Strata and Spatial Support

Table H3: Sampled chart support on all 54,379 external objects at $B = 512, K = 24$. Uncovered fraction is averaged per object; "any gap" is the percentage of objects with an uncovered sample. Tail radius is the mean per-object 95th-percentile nearest-chart radius. Support area is the mean individual ellipse area in canonical coordinates, counting overlaps separately. These are support statistics, not reconstruction errors. Bold marks the lowest uncovered fraction and gap frequency. Tail radius and support area describe the supports and are not ranked.

Representation	Uncovered samples (%) ↓	Objects with any gap (%) ↓	Mean per-object P95 radius	Mean individual support area
Uniform-local	**0.00**	**0.00**	0.6137	0.08109
Demand-adaptive	36.97	75.57	1.5699	0.02345
VecMALT	**0.00**	**0.00**	0.6136	0.07923

Table H4 and Figure A9 show how the method ranking varies with geometry and query scale. VecMALT improves on Uniform-local in the higher-complexity tiers, while Uniform-local has lower error in the simplest tier. VecMALT also has lower error on contained random windows, but not on the boundary-centered aggregate or at the two smallest longest-side scales. These strata qualify the aggregate gain in spatially conditioned readout.

The dataset-level support diagnostic applies the rule in Appendix F.2 to each external object and averages object statistics. Each ring or line component receives 1,024 samples, giving equal sample counts per group rather than length weighting. Tail radius is the mean per-object 95th percentile of nearest-chart radii. Mean individual support area is $K^{-1}\sum_k \pi s_{ik}^{\mathrm{maj}} s_{ik}^{\mathrm{min}}$, counting overlapping

Table H4: Chart-record window MAE by complexity and query distribution. Each group averages applicable windows within objects, then averages objects. The smallest-scale group omits objects with no sampled window at that scale. Bold denotes the lowest unrounded occupancy MAE in each row.

Stratum	Objects	Uniform-local	Demand-adaptive	VecMALT
Simple	7464	**0.01887**	0.02098	0.01956
Moderate	7267	0.02861	0.02775	**0.02767**
Complex	3544	0.04641	0.04844	**0.04517**
Extreme	1725	0.06253	0.06623	**0.06057**
Contained random windows	20000	0.03022	0.03150	**0.02921**
Boundary-centered windows	20000	**0.03189**	0.03292	0.03196
Longest side 1/16	19660	0.04784	**0.04700**	0.04836
Longest side 1/8	20000	**0.03832**	0.03875	0.03852
Longest side 1/4	20000	0.03079	0.03184	**0.03029**
Longest side 1/2	20000	0.02571	0.02754	**0.02486**

ellipses separately. Uniform-local and VecMALT have no sampled gaps on the 54,379 objects. Demand-adaptive's mean uncovered fractions are 43.77% for polygons and 18.98% for curves.

Demand-adaptive's uncovered fraction has a pooled Spearman correlation of -0.542 with window MAE on the 20,000 readout polygons. Object geometry and support scale vary, and the reader does not enforce the decoder's compact support. This pooled association does not identify a causal effect of support gaps on readout. It reinforces the separation used in Section 5.3: decoder support diagnostics describe access to the explicit detail branches, whereas window attention reads the complete records without that gate.

I GEO2VEC ADAPTATION AND CODE ACQUISITION

I.1 SHARED GEOMETRY AND CODE ACQUISITION

The external Geo2Vec adapter preserves the latent-conditioned distance-field mechanism and uses the common canonical geometries and object IDs. GeoParquet ingestion precedes this interface. Training jointly optimizes 512-dimensional codes for the same 355,820 curve/polygon objects and one shared decoder, using unsigned curve distances and signed polygon distances. Validation/test objects are absent from the training latent table; their codes are fitted from geometric observations after decoder freezing. Both acquisition routes use reference geometry: VecMALT constructs encoder evidence, while Geo2Vec samples distances for optimization. Neither uses external area/window task labels to acquire a code.

Both methods are compared through canonical shape codes, without a separate location representation. Geo2Vec training codes are jointly fitted with the decoder, while new-object codes are fitted after it is frozen. This difference can affect code distributions and reader transfer. The reported comparison therefore evaluates these acquisition workflows together with their respective architectures and training objectives.

I.2 TRAINING AND LATENT-FITTING BUDGETS

The external Geo2Vec adaptation uses latent width 512, hidden width 256, eight intermediate blocks, eight Fourier frequencies, polar-coordinate augmentation, and LeakyReLU slope 0.01. Shared training uses Adam with learning rate 0.001, latent regularization 0.0001, a 100,000-step cap, global batch 128 on eight GPUs, and 512 queries per object. The decoder is selected by a geometry-family macro-average field MAE on independently sampled queries of training objects, rather than the VecMALT Area-priority criterion on internal-validation objects. The selected step is 70,000; training stops at 80,000.

Table I1: Validation pilot for new Geo2Vec codes. Lower holdout field MAE is better. Fitting times are amortized over batches and differ from single-object acquisition latency. Bold denotes the lower error and lower fitting time, separately within each geometry family.

Fitting steps	Geometry family	Objects	Holdout field MAE ↓	Amortized fitting (seconds/object)
250	Curve	128	0.010794	**0.0988**
250	Polygon	128	0.006131	**0.0985**
1000	Curve	128	**0.010654**	0.3952
1000	Polygon	128	**0.005436**	0.3952

The validation pilot compares 250 and 1,000 fitting steps on 128 new objects per family. New latents use Gaussian initialization with standard deviation 0.01 and Adam with learning rate 0.01, batch size 32, 512 sampled queries per step, and regularization 0.0001. Each object has 2,048 fitting observations and 512 independent uniform holdout samples. The longer budget reduces polygon holdout field MAE from 0.00613 to 0.00544 and curve MAE from 0.01079 to 0.01065; all reported validation/test exports use 1,000 steps. External holdout field MAE is 0.00430 for polygons and 0.01059 for curves. These diagnostics justify the selected budget without establishing convergence. The same frozen external codes are used for common reconstruction and area/window reading.

I.3 ACQUISITION-TIME MEASUREMENT

Table I2 and Figure A10 report acquisition on the same 32 external objects—24 polygons and eight curves—using an A100-SXM4-80GB and four CPU threads. Timing starts from the canonical reference and includes evidence construction or distance sampling plus code generation. It excludes file loading, CRS conversion, canonicalization, and reader training. Encoder-only measurements start with device-resident evidence. Geo2Vec's 1,000-step single-object acquisition averages 3.176 s; a separate batch-32 export gives approximately 0.397 s per object. The latter measures amortized throughput, not single-object latency.

Table I2: Acquisition on shared objects. Mean, median, and P95 include evidence construction or sampling and code generation (s/object), starting from canonical geometry. The final columns isolate encoding from prepared, device-resident evidence. A dash indicates that Geo2Vec has no encoder. Bold marks lower acquisition/encoding times and higher encoding throughput, separately by column.

Representation	Acquisition mean (s)	Median (s)	P95 (s)	Encoding batch 1 (ms)	Encoding batch 16 (objects/s)
Geo2Vec (adapted)	3.176	3.279	**3.325**	—	—
Global-only	**2.338**	1.016	9.380	**3.37**	**2670.5**
Uniform-local	2.342	**0.998**	9.329	18.07	512.8
Demand-adaptive	2.349	1.010	9.348	19.27	529.1
VecMALT	2.352	1.023	9.403	18.73	462.2

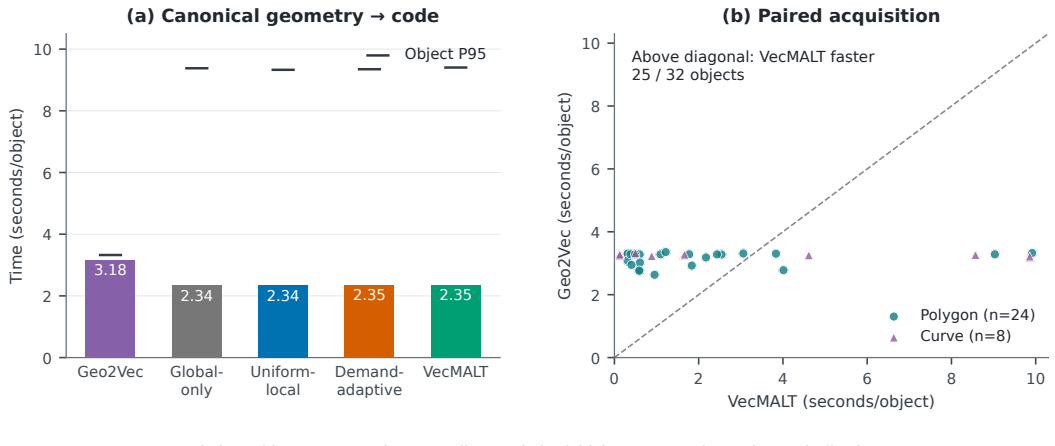


Figure A10: Code acquisition on 32 shared objects. (a) Mean time from canonical geometry to code and the object-level 95th percentile. (b) Paired VecMALT/Geo2Vec times; points above the equal-time diagonal favor VecMALT. Evidence construction or distance sampling and latent fitting are included; file loading and initial CRS conversion/canonicalization are excluded. Encoder-only and cached-reader timings measure different stages.

The Geo2Vec runner records 2,782.70 s for shared training, including in-stage validation, checkpointing, and final training-code export, but excluding data preparation. Its timer accumulates elapsed time across resumed invocations. Cached area/window readout uses three synchronized repetitions after warm-up, starting with standardized CPU arrays and ending with CPU predictions. Device transfers are included; disk loading, standardization, and code acquisition are excluded. Reconstruction timings in Appendix G.3 use different objects and CPU settings and are kept separate.

J Common-Object Reconstruction and Readout Analysis

J.1 Common-Object Reconstruction Protocol

The five-method $B = 512$ comparison uses the same 54,379 external objects: 39,473 polygons and 14,906 curves. Stable IDs align canonical reference geometries and coordinate records. Source composition is given in Appendix B.3. Geo2Vec uses the frozen decoder and 1,000-step new-code fitting specified in Appendix I.

All methods query a 512×512 grid over $[-0.75, 0.75]^2$, with no contour refinement and 2,048 polygon-boundary samples. Canonical distances measure normalized shape error; meter distances multiply them by each object's positive metric scale. Curve extraction differs: the encoder–decoder models use predicted endpoints and tangents, whereas Geo2Vec uses UDF thresholding and skeleton extraction. Boundary F1 is omitted because its canonical tolerances differ (0.005870819 for the VecMALT family and 0.005 for Geo2Vec); this tolerance does not affect the reported IoU and distances. All complexity strata use the common training-frozen registry. Geo2Vec's prepared corpus differs in 11,597 complexity labels (8,008 polygons and 3,589 curves), with unchanged reference geometries.

Table J1: Common-object reconstruction at $B = 512$. Polygon IoU includes all 39,473 objects. Canonical CD and HD95 are scaled by 10^3 for display; meter-valued HD95 is reported separately. Means use the stated finite counts. Uniform-local and Demand-adaptive each have one empty polygon, with IoU zero and undefined boundary distance; all 14,906 curves have finite distances. Buffered-curve IoU and endpoint scores lack matched Geo2Vec outputs and are omitted here. Bold denotes the best unrounded value per metric within each geometry family.

Polygons **Method**	**IoU (%) ↑**	**CD $\times 10^3$ ↓**	**HD95 $\times 10^3$ ↓**	**HD95 (m) ↓**	**Finite *n***
Geo2Vec (adapted)	98.5285	1.8273	4.9012	0.4014	39,473
Global-only	98.7891	1.4601	4.1834	0.7714	39,473
Uniform-local	**99.3783**	0.9132	2.2402	0.6517	39,472
Demand-adaptive	99.3210	0.9813	2.5825	**0.3682**	39,472
VecMALT	99.3772	**0.9116**	**2.2059**	0.5980	39,473
Curves **Method**	**CD $\times 10^3$ ↓**	**HD95 $\times 10^3$ ↓**	**HD95 (m) ↓**	**Finite *n***	
Geo2Vec (adapted)	3.8022	10.2332	1.3176	14,906	
Global-only	**1.7674**	**3.0205**	**0.8945**	14,906	
Uniform-local	2.2900	4.1726	0.9374	14,906	
Demand-adaptive	2.5557	4.8401	1.2159	14,906	
VecMALT	2.2289	4.0224	0.9684	14,906	

VecMALT has higher polygon IoU than adapted Geo2Vec on 98.18% of objects. This per-object comparison complements the population mean in Table J1 by showing how broadly the overlap gain is distributed.

Table C2 retains the original capacity-study summaries, whereas Table J1 uses the common-object reconstruction export. For Demand-adaptive, the archived mean polygon HD95 is 0.3683 m in Table C2 and 0.3682 m in Table J1. Each table retains its corresponding source value.

J.2 Scale-Dependent Boundary Errors

The mean polygon HD95 in meters is lower for Demand-adaptive and Geo2Vec than for VecMALT, although VecMALT has lower HD95 than Geo2Vec on 98.37% of polygons. Multiplication by

positive object scales preserves each paired ordering but changes the influence of objects in the mean. Table J2 reports the full distribution and a shared scale-limited diagnostic subset.

Table J2: Distribution of per-polygon HD95 in meters. Median and P95 are across object-level HD95 values. The shared diagnostic subset has 39,078 reference polygons with pose scale at or below the full population's 99th percentile, 362.0883 m. Finite-distance counts are reported for that subset; full-population results remain the primary evaluation. Bold denotes the lowest value in each distance column.

Method	Full mean ↓	Full median ↓	Full P95 ↓	Common-scale mean ↓	Subset finite objects
Geo2Vec (adapted)	0.4014	0.02978	1.1401	0.2061	39,078
Global-only	0.7714	0.00902	1.5839	0.2280	39,078
Uniform-local	0.6517	**0.00896**	0.6556	0.1063	39,077
Demand-adaptive	**0.3682**	0.00937	0.6769	0.1187	39,077
VecMALT	0.5980	0.00900	**0.6421**	**0.1044**	39,078

One valid São Paulo land-use polygon, with 2,116 vertices and pose scale 85,662 m, contributes 53.73% of VecMALT's summed meter-valued polygon HD95. Its canonical HD95 is 0.148061, or 12,683 m, compared with 1,912 m for Geo2Vec and 3,374 m for Demand-adaptive. Its stable ID is `014700c306c8f0f9a6c6c17a160c2275d6bf3485d4217c83129878a13106f456`. The object remains in every full-population result. The common-scale subset identifies the large-scale tail behind the mean-ranking reversal; it does not replace the full evaluation.

J.3 Reconstructed Components and Holes

Extracted structure reveals errors that can have little effect on overlap. Geo2Vec predicts more holes than the reference on 6,448 polygons (16.335%) and a nonintersecting extra component on 10,271 (26.020%). VecMALT's corresponding counts are 150 (0.380%) and 205 (0.519%). Component counts match exactly on 71.991% of Geo2Vec predictions and 99.341% of VecMALT predictions. This exact-match rate differs from the exported `area_connectivity_count_accuracy`, which averages $1/(1+|\hat{c}-c|)$; Geo2Vec scores 0.80993 on that measure.

Geo2Vec's nonintersecting extra components occupy only 0.0066% of predicted area on average, explaining how frequent fragments can coexist with high IoU. Hole areas were not measured, and equal hole counts do not establish spatial correspondence. The reference set contains only 17 polygons with holes (21 holes total) and three multipart polygons, so these results mainly characterize extraction artifacts rather than broad topology preservation.

J.4 Reconstruction and Independent Readout

Table J3 aligns reconstruction and readout on the same 20,000 test polygons by object keys and canonical references. Each object has 64 common windows. Decoder-area error is $|\operatorname{Area}(\widehat{\widetilde{G}}_i) - \operatorname{Area}(\tilde{G}_i)|$; the area MLP predicts the same canonical quantity directly from the code. Window MAE is averaged over each object's windows and then over objects.

The high-IoU panel isolates the 13,394 polygons for which Geo2Vec reaches IoU ≥ 0.99 and uses those same objects for every method. Geo2Vec's independent area MAE remains 0.027730, compared with 0.000968 for VecMALT. The readout gap therefore persists when the analysis is restricted to objects that Geo2Vec reconstructs well; it is not confined to its largest reconstruction failures.

This subset does not preserve every full-cohort ranking: Uniform-local has lower chart-record window MAE than VecMALT, and Demand-adaptive has lower direct area MAE (Table J3). The diagnostic supports the distinction between reconstruction and readout while showing that allocation rankings depend on the evaluated objects. Window predictions obtained by reconstructing and then intersecting were not evaluated.

Table J3: Reconstruction and independent prediction on the same polygons. Panel 1 uses the fixed 20,000-object readout test set. Panel 2 uses the same 13,394 objects with Geo2Vec IoU ≥ 0.99 for all methods as a post-hoc diagnostic. Canonical-area MAE uses squared canonical units; window MAE is an occupied fraction. The reconstruction path uses its declared side information. MAE $\downarrow$; a dash marks an inapplicable chart-record reader. Bold denotes the best value per metric within each cohort panel.

Method	Decoder IoU (%) $\uparrow$	Decoder-area MAE $\downarrow$	Area MLP MAE $\downarrow$	Flat window MLP MAE $\downarrow$	Chart-record window MAE $\downarrow$
Fixed readout cohort: 20,000 polygons					
Geo2Vec (adapted)	98.5432	0.001319	0.042676	0.090872	—
Global-only	98.7948	0.001023	0.002316	**0.042288**	—
Uniform-local	99.3851	**0.000506**	0.001895	0.043485	0.031054
Demand-adaptive	99.3262	0.000709	0.002093	0.046265	0.032207
VecMALT	**99.3872**	0.000527	**0.001809**	0.044872	**0.030585**
Common Geo2Vec-IoU ≥ 0.99 subset: 13,394 polygons					
Geo2Vec (adapted)	99.5702	0.000796	0.027730	0.061092	—
Global-only	99.8862	0.000247	0.001351	**0.032972**	—
Uniform-local	**99.8996**	**0.000202**	0.000991	0.034207	**0.019444**
Demand-adaptive	99.8633	0.000356	**0.000794**	0.035411	0.019772
VecMALT	99.8988	0.000227	0.000968	0.036000	0.019668

K Public Benchmarks: Protocols and Detailed Results

K.1 Datasets and Task-Label Splits

Public tasks test geometric attributes beyond area and local occupancy. Shape classification concerns overall configuration, while Edge predicts the stored boundary discretization. We train separate representations on ShapeClassification, Singapore polygons, and the polygon layer of NYC_total_data. Shape contains E, F, H, I, L, O, T, U, Y, and Z, with 75 test objects per class. Table K1 reports the actual cohorts and task splits; NYC point and line layers are excluded, and none of the three cohorts contains multipart polygons.

All five representations are trained from scratch on every geometry in each cohort, yielding 15 core models, without Shape or Edge labels. After freezing the codes, task-training labels fit the readers, validation labels select their epochs, and test labels determine the reported scores. This is a transductive geometry protocol: test geometries are visible during representation learning, but their task labels are withheld. Exact raw-geometry duplicates are grouped within a split. Splits are stratified by Shape class or merged log-edge-count bins; this grouping does not remove near duplicates or spatial dependence.

Canonicalization preserves orientation and uses translation and isotropic scaling. Shape has no declared CRS and is treated as intrinsic Cartesian geometry; Singapore and NYC retain their recorded projected frames. There is no repair, simplification, or vertex thinning. Edge counts all exterior- and interior-ring segments, subtracting the duplicated closing vertex from each ring's coordinate count and summing over components. Collinear vertices are retained, so the target is not the minimal number of geometric sides. MAE and RMSE use continuous predictions without rounding.

Table K1: Public polygon cohorts and task-label splits. Every listed geometry participates in representation learning; labels are split for reader training, validation, and testing. Exact raw duplicates stay together. Orientation is preserved without repair or simplification. NYC counts refer only to the selected polygon layer.

Dataset	Geometry cohort	Task train	Task validation	Task test	Objects with holes	Tasks	Source frame
Shape	5,010	3,510	750	750	0	Shape; Edge	Intrinsic Cartesian
Singapore	109,877	76,912	16,482	16,483	866	Edge	EPSG:3414
NYC polygons	20,000	13,998	3,001	3,001	38	Edge	EPSG:32118

K.2 Model Parameters and Code Storage

Each representation exports a complete 512-dimensional shape code. Local methods use a 128-dimensional global embedding and 24 records of two anchor, three metric, and eleven feature coordinates. Table K2 separates shared network parameters from per-object storage and reader parameters. Encoder–decoder totals include the geometry encoder, global feature pyramid, field predictor, and landmark predictor; the landmark count is part of the decoder total. Independent readers do not receive landmark outputs.

Public Geo2Vec codes are the jointly optimized latent-table entries at the selected cohort checkpoint. Unlike the external protocol, this setting requires no new-object fitting with a frozen decoder. Every evaluated public object therefore has a code learned jointly with its decoder. The experiment is a common 512-dimensional shape-code adaptation, with the protocols in this appendix rather than the full original shape-and-location benchmark.

Geo2Vec registers 2,421,189 shared parameters, including an unused 2,756-parameter coordinate projection frozen by the runner. Active trainable shared parameters therefore total 2,418,433. Joint training also optimizes $N \times 512$ latent coordinates: 2,565,120 for Shape, 56,257,024 for Singapore, and 10,240,000 for NYC. Encoder-based models have no independently optimized object table, but

saving their output codes costs the same $N \times 512$ scalars. Each FP32 code occupies 2 KiB; pose, header, IDs, buffers, networks, and readers are separate. External Geo2Vec inference can load the shared decoder without its training latent table.

Table K2: Shared network parameters at $B = 512$, excluding codes, readers, buffers, pose, and headers. The decoder includes the feature pyramid, field predictor, and landmark predictor; the landmark column is a subset, not an extra term. Geo2Vec registers 2,421,189 shared parameters, including 2,756 in an unused frozen projection, leaving 2,418,433 active trainable parameters. It has no geometry encoder and separately optimizes an $N \times 512$ latent table.

Model	Geometry encoder	Shared decoder	Landmark subset	Shared total
Geo2Vec (adapted)	—	2,421,189	—	2,421,189
Global-only	3,946,944	4,092,722	724,480	8,039,666
Uniform-local	4,290,278	3,318,133	724,480	7,608,411
Demand-adaptive	4,402,407	3,318,133	724,480	7,720,540
VecMALT	4,327,911	3,318,133	724,480	7,646,044

K.3 Representation Training and Checkpoint Selection

Public checkpoints use geometry-only cohort monitors, with no downstream labels. The encoder–decoder monitor evaluates deterministic cohort objects and cached queries; Geo2Vec uses fixed independent field queries from cohort objects. These monitor sets are not unseen-geometry validation. Encoder–decoder runs have a 200,000-step cap; Geo2Vec caps are 120,000 for Shape, 500,000 for Singapore, and 120,000 for NYC. Table K3 distinguishes the cap, last recorded step, and monitor-best checkpoint.

Public encoder–decoder training uses AdamW with learning rate 10^{-4}, weight decay 10^{-6}, cost-based batches capped at 24 objects, and at most 7,168 queries per object. Geo2Vec uses Adam with learning rate 10^{-4}, latent regularization 10^{-4}, global batch 256, and 64 queries per object. Shape Uniform-local and VecMALT were stopped manually and exported their saved best checkpoints; last logged steps need not equal exact stop times. Optimizer steps, queries per step, and network sizes differ, so matching code width and label splits does not match compute.

Table K3: Public representation training. Cap, last logged step, and geometry-monitor-best checkpoint are distinct. Shape Uniform-local and VecMALT were stopped manually and use saved best checkpoints; exact stop times need not match the last logs. Monitors use cohort field error without task labels. Step counts are not matched compute.

Dataset	Method	Step cap	Last recorded step	Monitor-best step	Stop status
Shape	Geo2Vec (adapted)	120,000	120,000	118,000	Budget reached
Shape	Global-only	200,000	104,000	94,000	Early stopping
Shape	Uniform-local	200,000	91,000	85,000	Manual stop; saved best
Shape	Demand-adaptive	200,000	55,000	45,000	Early stopping
Shape	VecMALT	200,000	85,000	84,000	Manual stop; saved best
Singapore	Geo2Vec (adapted)	500,000	500,000	498,000	Budget reached
Singapore	Global-only	200,000	86,000	76,000	Early stopping
Singapore	Uniform-local	200,000	43,000	33,000	Early stopping
Singapore	Demand-adaptive	200,000	29,000	19,000	Early stopping
Singapore	VecMALT	200,000	52,000	42,000	Early stopping

Continued on next page

Table K3 (continued)

Dataset	Method	Step cap	Last recorded step	Monitor-best step	Stop status
NYC polygons	Geo2Vec (adapted)	120,000	120,000	120,000	Budget reached
NYC polygons	Global-only	200,000	162,000	152,000	Early stopping
NYC polygons	Uniform-local	200,000	48,000	38,000	Early stopping
NYC polygons	Demand-adaptive	200,000	52,000	42,000	Early stopping
NYC polygons	VecMALT	200,000	58,000	48,000	Early stopping

K.4 Task Readers and Shape Classifier Refinement

The common flat MLP is 512 → 256 → 256 → output, with ReLU and dropout 0.5. The original object-query chart reader projects the global embedding and complete records to width 64 and pools them with a learned query and four-head attention. It remains the Edge reader. The revised Shape classifier uses two-layer width-128 global and local-record encoders. The global feature generates a four-head query over local tokens only; the resulting context is concatenated with the global feature through a direct path to the classifier. It has no chart-slot embeddings or geometry side input. The revision changes three Shape heads only, retaining the core models, codes, splits, and Edge heads.

Table K4 lists the public and external readers separately. Public readers use Adam with learning rate 0.001, weight decay 10^{-5}, batch size 32, seed 42, and at most 100 epochs. Shape uses cross-entropy and validation accuracy; Edge uses L1 loss and validation MAE. Standardization is fitted on task-training codes, sharing each local channel's statistics across records. All predictions use the frozen-code inputs defined in Section 3.4.

Table K4: Reader architectures and registered parameter counts for complete frozen 512-dimensional codes. Only window readers receive a four-coordinate rectangle. Edge outputs are continuous, without sigmoid or rounding. The revised Shape classifier changes only classification heads. Standardization uses task-training data, with shared local-channel statistics for record readers.

Protocol / target	Reader	Architecture / query	Registered parameters
A. External area/window readers (different protocol)			
External area	Flat-code MLP	512 → 128 → 64 → 1; SiLU, sigmoid	73,985
External window	Flat-code MLP	[512-D code; 4-D window] → 128 → 64 → 1; SiLU, sigmoid	74,497
External window	Code-chunk attention	25 padded vector chunks; width 64, 4 heads; window query	28,353
External window	Chart-record attention	Global token + 24 full records; width 64, 4 heads; window query	34,689
B. Public Shape/Edge readers			
Public Shape	Flat-code MLP	512 → 256 → 256 → 10; ReLU, dropout 0.5	199,690
Public Edge	Flat-code MLP	512 → 256 → 256 → 1; ReLU, dropout 0.5	197,377

Continued on next page

Table K4 (continued)

Protocol / target	Reader	Architecture / query	Registered parameters
Public Shape (original)	Original object-query chart-record reader	Global token + records; learned query; width 64, 4 heads; logits	34,954
Public Edge	Object-query chart-record reader	Global token + records; learned query; width 64, 4 heads; scalar	34,369
Public Shape (revised)	Global-conditioned chart-record classifier	2-layer global/record MLPs; width 128, 4 heads over local tokens; direct global path; 10 logits	168,586

Table K5 retains the original Shape reader, the flat MLP, and the revised classifier on unchanged codes. The revision restores competitive test accuracy, but its selected validation accuracy is lower than the flat MLP's for all three local methods. It was developed after inspecting earlier results, so these small test gains are exploratory rather than a new blind comparison. The three additional heads bring the public record to 35 reader results without additional core training.

Table K5: Shape readers on unchanged codes and the same 750 test objects. Accuracy and macro-F1 are percent. All heads are selected by validation accuracy; the original record-reader results remain alongside the exploratory revision. Revised test accuracy is slightly higher than the flat MLP's, but selected validation accuracy is lower for all three methods. Core models and Edge readers are unchanged. Bold denotes the best unrounded score among readers of the same representation, separately for test and validation metrics.

Method	Shape reader	Test accuracy (%) ↑	Test macro-F1 (%) ↑	Selected validation accuracy (%) ↑	Epoch	Reader parameters
Uniform-local	Flat-code MLP	97.07	97.07	**98.53**	26	199,690
Uniform-local	Original object-query chart-record reader	90.00	90.03	92.13	95	34,954
Uniform-local	Global-conditioned chart-record classifier	**97.20**	**97.20**	97.47	86	168,586
Demand-adaptive	Flat-code MLP	97.60	97.60	**99.07**	98	199,690
Demand-adaptive	Original object-query chart-record reader	92.93	92.91	95.33	77	34,954
Demand-adaptive	Global-conditioned chart-record classifier	**97.87**	**97.88**	98.40	82	168,586
VecMALT	Flat-code MLP	95.60	95.64	**98.13**	23	199,690
VecMALT	Original object-query chart-record reader	88.27	88.33	91.60	97	34,954
VecMALT	Global-conditioned chart-record classifier	**96.67**	**96.69**	97.20	45	168,586

K.5 Complete Edge-Counting Results

Table K6 reports all count MAE and RMSE values; Figure A11 shows the relative MAE change for each same-code reader comparison. The errors are computed from continuous predictions on every task-test object.

The MAE improvement does not imply an RMSE improvement in every case. For Uniform-local on NYC, record-aware reading lowers MAE while increasing RMSE from 1.2870 to 1.3194. The two losses weight large errors differently. This exception is retained alongside the nine MAE improvements, rather than subsumed under a claim that every error measure improves.

Table K6: Complete public boundary-segment count errors (segments), using continuous predictions on all test objects. Record rows use the object-query Edge reader, not the revised Shape classifier. All nine within-code comparisons reduce MAE; RMSE need not improve. Bold denotes the lowest error within each dataset across the listed representation–reader pairs.

Dataset	Method	Reader	Test objects	MAE ↓	RMSE ↓	Selected epoch
Shape						
Shape	Geo2Vec (adapted)	Flat-code MLP	750	2.6452	4.1794	22
Shape	Global-only	Flat-code MLP	750	0.8993	1.5105	89
Shape	Uniform-local	Flat-code MLP	750	1.2041	1.9419	90
Shape	Uniform-local	Object-query chart-record reader	750	1.0472	1.7754	97
Shape	Demand-adaptive	Flat-code MLP	750	1.0221	1.6553	86
Shape	Demand-adaptive	Object-query chart-record reader	750	**0.8142**	**1.3071**	93
Shape	VecMALT	Flat-code MLP	750	1.2167	1.9350	94
Shape	VecMALT	Object-query chart-record reader	750	0.9883	1.4233	88
Singapore						
Singapore	Geo2Vec (adapted)	Flat-code MLP	16,483	2.3416	7.7800	28
Singapore	Global-only	Flat-code MLP	16,483	0.6724	2.7520	36
Singapore	Uniform-local	Flat-code MLP	16,483	0.8364	2.8094	9
Singapore	Uniform-local	Object-query chart-record reader	16,483	0.6799	2.6683	91
Singapore	Demand-adaptive	Flat-code MLP	16,483	0.9292	3.7236	7
Singapore	Demand-adaptive	Object-query chart-record reader	16,483	0.6620	2.7596	74
Singapore	VecMALT	Flat-code MLP	16,483	0.7647	2.5201	12
Singapore	VecMALT	Object-query chart-record reader	16,483	**0.5581**	**2.2924**	82
NYC polygons						
NYC polygons	Geo2Vec (adapted)	Flat-code MLP	3,001	1.0447	2.8666	95
NYC polygons	Global-only	Flat-code MLP	3,001	0.4227	1.6630	43
NYC polygons	Uniform-local	Flat-code MLP	3,001	0.4369	1.2870	68

Continued on next page

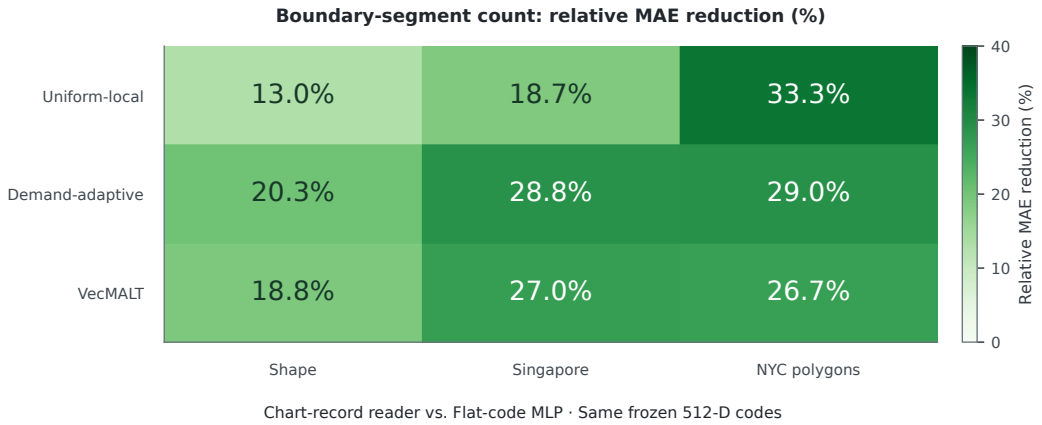


Figure A11: Relative boundary-segment count MAE reduction from reading complete records instead of flattening the same frozen 512-dimensional code. Each cell is $100(\mathrm{MAE}_{\mathrm{flat}} - \mathrm{MAE}_{\mathrm{record}})/\mathrm{MAE}_{\mathrm{flat}}$, computed from unrounded errors. The color scale is in percent. All geometries participate in representation learning; task-test labels are held out.

Table K6 (continued)

Dataset	Method	Reader	Test objects	MAE ↓	RMSE ↓	Selected epoch
NYC polygons	Uniform-local	Object-query chart-record reader	3,001	0.2914	1.3194	100
NYC polygons	Demand-adaptive	Flat-code MLP	3,001	0.3261	1.4946	36
NYC polygons	Demand-adaptive	Object-query chart-record reader	3,001	**0.2314**	**0.7302**	86
NYC polygons	VecMALT	Flat-code MLP	3,001	0.4062	1.2198	94
NYC polygons	VecMALT	Object-query chart-record reader	3,001	0.2978	1.1416	82

Figure A12 groups errors by reference boundary-segment count, using the same bins across methods. On Singapore, VecMALT has the lowest MAE in every analyzed bin above four segments. The highest-count bins contain only three Shape objects (33–64 segments) and one NYC object (65+), so they describe individual cases rather than stable tail estimates. Hollow markers and dotted connections identify bins with fewer than ten objects; empty bins remain blank.

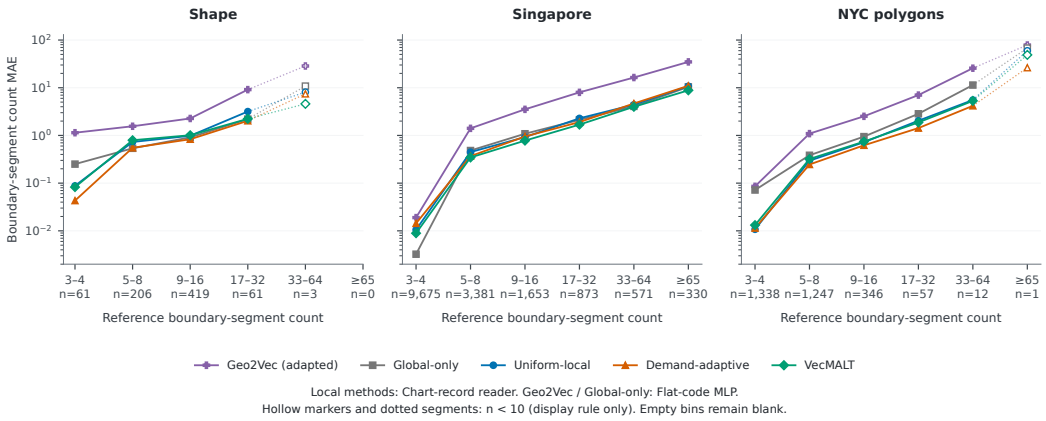


Figure A12: Boundary-segment count MAE by reference segment count (log scale). Local methods use object-query chart-record readers; Global-only and Geo2Vec use flat MLPs. Bin counts are shared across methods. Hollow markers and dotted connections mark bins with fewer than ten objects; empty bins remain blank. Shape's 33–64 bin has three objects, and NYC's 65+ bin has one. All geometries participate in representation learning; task-test labels are held out.

L Extended Related Work

Geospatial geometry encoding. Representation learning for geospatial objects seeks reusable features from heterogeneous vector data (Chen et al., 2025). Polygon encoders explicitly address invariances, multiple parts, and holes (Mai et al., 2023); Poly2Vec uses Fourier-based encoding across points, polylines, and polygons (Siampou et al., 2025). Geo2Vec learns distance-field representations with geometry-adaptive sampling, jointly optimizing object-specific codes and shared network parameters (Algorithm 1 of Chu & Shahabi, 2026). These methods establish geometric encoding as a route to downstream learning. VecMALT focuses on the internal organization of a fixed-width shape code: a shared encoder allocates evidence to spatially addressed records that can be decoded or read independently. Position and context encodings such as Space2Vec (Mai et al., 2020) address a different aspect of geospatial representation. Section 4 specifies the Geo2Vec shape-code adaptations used in our comparisons.

Vector graphics and sketches. DeepSVG represents the hierarchy of vector drawing commands (Carlier et al., 2020), while SketchINR maps variable-length sketches to fixed-dimensional codes and predicts coordinates from time and stroke parameters (Bandyopadhyay et al., 2024). Thus fixed-width encoding alone does not distinguish VecMALT. Our focus is the spatial organization of polygon and curve evidence, rather than a drawing sequence or stroke parameterization.

Implicit fields and structured local representations. DeepSDF and Occupancy Networks condition continuous shape functions on global codes (Park et al., 2019; Mescheder et al., 2019). IF-Nets and Convolutional Occupancy Networks provide spatial features near a query (Chibane et al., 2020; Peng et al., 2020); DeepLS and Local Implicit Grids organize local latent codes on grids (Chabra et al., 2020; Jiang et al., 2020). These designs reduce reliance on a single global feature when decoding local geometry.

Spatial frames are also central to structured shape representations. SIF learns local shape elements and demonstrates shape-analysis uses (Genova et al., 2019). LDIF pairs those elements with local latent vectors and includes both geometric and latent parameters in its representation-size comparison (Figure 4 of Genova et al., 2020). PatchNets learns local patches with explicit positions, orientations, and scales (Tretschk et al., 2020). VecMALT builds on this approach to locality. Its specific focus is to construct frames and features from a constrained evidence distribution through joint transport and conditional moments, alongside a global embedding, and to examine the records with readers separate from the original decoder.

Specialization and allocation. LP-DIF specializes implicit functions to local patterns, and ALTO alternates point and grid latent structures (Wang et al., 2023a;b). DCC-DIF uses continuous, learnable code positions guided toward complex geometry (Li et al., 2022); ACORN and OctField adapt hierarchical spatial decompositions (Martel et al., 2021; Tang et al., 2021). VecMALT instead adjusts target mass over evidence tokens while holding record count and width fixed at each code budget. Entropy-regularized transport (Cuturi, 2013) couples assignments across records, and reference-boundary supervision addresses the spatial support induced by their frames. This separates the choice of participating evidence from the reach of the resulting local representation.

Independent use of structured codes. Deep Sets and Set Transformer provide established mechanisms for processing sets with shared functions and attention (Zaheer et al., 2017; Lee et al., 2019). We use MLP and attention readers to evaluate the representation rather than propose a new reader architecture. Same-code interface comparisons and feature–frame reassignment test whether retaining record structure helps a new model access geometry. They complement reconstruction and raw-feature retrieval, which test different uses of a code.