Differentiable Range-Partition Entropy for Entropy-Sensitive Geometric Algorithms

Ibne Farabi Shihab[1] **Sanjeda Akter**[1] **Anuj Sharma**[2]

[1]Department of Computer Science, Iowa State University
[2]Department of Civil, Construction and Environmental Engineering, Iowa State University

ishihab@iastate.edu

Abstract

Range-partition entropy is a structural complexity measure used in entropy-sensitive computational geometry. We introduce Differentiable Entropy Regularization (DER), a gradient-optimizable surrogate for the entropy of learned geometric partitions. The paper makes three main contributions. First, we separate the normalized entropy optimized by DER, denoted $\bar{H}_{\mathcal{R}}(S)$, from the runtime-scaled entropy $H^{\mathrm{work}}_{\mathcal{R}}(S) = n\bar{H}_{\mathcal{R}}(S)$ that appears in algorithmic work bounds. Second, for a fixed margin-separated halfspace arrangement, we prove a self-contained soft-to-hard approximation theorem: the soft halfspace-cell entropy converges to the corresponding hard sign-pattern entropy at a rate controlled by margin, temperature, and the number of separators. A connection to optimal range-partition entropy holds when the fixed partition is admissible for the geometric problem and near-optimal in the admissible partition class. Third, we evaluate DER as learned geometric preprocessing for SciPy/Qhull-based convex-hull and Delaunay pipelines. On the audited 2D convex-hull tasks, the anchor pipeline reaches up to $3.00\times$ end-to-end speedup at 10K points and $1.41\times$ at one million points (vertex F1 0.980; containment violation 0.39%); the matched, fully charged halfspace pipeline reaches $2.30\times$ on Synthetic High and admits a separate entropy-approximation diagnostic after freezing. Both modes are approximate and require task-specific fidelity checks. We also include a secondary transfer study in ViT-style attention, where the same regularizer induces structured sparsity and improves empirical efficiency. The central contribution is not the entropy formula alone, which resembles soft clustering, but the range-aware differentiable construction, the explicit approximation analysis, and the geometry-first training pipeline.

1 Introduction

Entropy-sensitive geometric algorithms adapt their runtime to input structure: easy instances run close to linearly, while hard instances pay additional work for genuinely complex geometry. A central quantity in this line of work is range-partition entropy, which measures how a point set can be decomposed by geometric ranges such as halfspaces or balls. Recent work by Eppstein et al. [14] formalizes and studies this measure for entropy-bounded computational geometry, and earlier instance-optimal analyses [6, 1] state work bounds that can be written as a normalized entropy multiplied by n. Our contribution is complementary: we introduce differentiable surrogates for such partition entropies so that they can be optimized by gradient methods.

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Positioning of DER relative to common regularization and efficiency methods. "Algorithmic grounding" means the objective targets a complexity measure appearing in a known algorithmic runtime analysis. "Provable RT" means the method is associated with a formal runtime bound in its native setting.

Method	Alg. Grounding	Provable RT
Label smoothing, IB	No	No
Longformer, BigBird	No (fixed)	No
FlashAttention	Yes (I/O)	Yes (I/O)
DER (geometry)	Restricted geom.	Conditional
DER (attention transfer)	Empirical	No

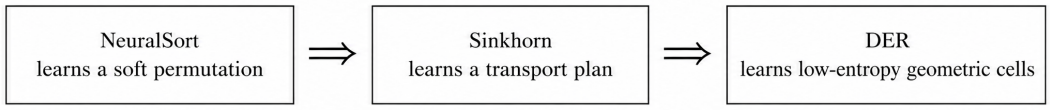


Figure 1: Conceptual comparison of differentiable optimization approaches. NeuralSort learns permutations, Sinkhorn Networks learn transport plans, and DER optimizes a structural entropy surrogate tied to geometric partitions rather than a specific discrete operation.

The key interface is a normalization convention. Let $\bar{H}_{\mathcal{R}}(S)$ denote the normalized entropy associated with the admissible partition family used in an entropy-sensitive analysis, and let $H_{\mathcal{R}}^{\mathrm{work}}(S) = n\bar{H}_{\mathcal{R}}(S)$ denote the corresponding runtime-scaled quantity. Our approximation theorems are stated in normalized entropy units, and runtime interpretations require an external algorithmic theorem for the relevant admissible partition family before converting back to work units by multiplying by n. The entropy formula $H_{\mathrm{diff}} = -\sum_j p_j \log p_j$ over soft aggregate masses resembles standard mixture-weight entropy and is not, by itself, the novelty. The contribution lies in the range-aware construction: learnable anchors or halfspace separators are optimized jointly with the representation, and the halfspace variant is aligned with the partition family used by convex-hull-style analyses. The formal theorem is intentionally modest, stating that for a fixed margin-separated halfspace arrangement, the soft entropy approximates the entropy of the corresponding hard sign-pattern partition; a statement about optimal range-partition entropy follows only under an additional near-optimality and admissibility assumption.

Why range-partition entropy rather than covering numbers or description-length proxies? The answer is its constructive relationship to algorithmic work in settings where an entropy-sensitive analysis is available. Informally, low partition entropy means a few large cells dominate the decomposition, while high entropy means the input induces many comparable cells. In this paper, that theory motivates the objective; the experiments measure whether learned low-entropy preprocessing improves concrete SciPy/Qhull-based geometry pipelines. Because the learned preprocessor perturbs the input, these experiments are approximate-pipeline experiments unless otherwise stated, not exact replacements for the original geometric problem.

Our method, Differentiable Entropy Regularization (DER), adds a differentiable entropy term to the loss during training. In computational geometry, DER trains a preprocessing network to reduce the entropy of transformed point sets subject to geometric preservation losses, and this is the primary theoretically grounded application. In attention models, DER regularizes key/query representations so that later sparsification produces more structured masks; this is presented as a transfer study rather than as a consequence of the geometric theorem. Conceptually, DER sits alongside other differentiable relaxations of discrete structure, but with a different target: NeuralSort learns soft permutations and Sinkhorn networks learn transport plans, while DER learns low-entropy geometric cells aligned with a specified range family (Figure 1).

We validate DER across two regimes. The primary regime is 2D geometric preprocessing, where we report solver-side and end-to-end timing on convex hull and quality-audited Delaunay experiments and show that the surrogate tracks a reference entropy computed in a restricted halfspace family.

We report hull-vertex and boundary fidelity alongside hull-area error, and explicitly separate solver-only time from total time. The secondary regime is ViT-Small/CIFAR-100 [13], where DER is

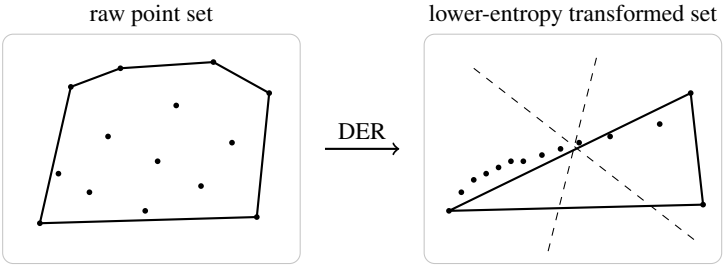


Figure 2: Geometric intuition behind DER. The learned preprocessor is trained to reduce a differentiable partition-entropy surrogate while preserving geometric quality. This figure is schematic; quantitative timing and error measurements are reported in Section 4.

evaluated as an empirical representation-structure regularizer for sparse attention, illustrating possible transfer rather than implications of the geometric theorem. Taken together, the paper introduces range-aware differentiable surrogates for partition entropy, gives a self-contained margin-based approximation theorem for fixed full-support halfspace sign-pattern partitions together with an α-optimal corollary, demonstrates empirical runtime improvements with explicit preprocessing/solver/runtime accounting, and reports a controlled empirical attention-transfer study.

2 Related Work

DER connects entropy-sensitive geometry with differentiable optimization. Eppstein et al. [14] study range-partition entropy, while instance-optimal analyses relate geometric structure to algorithmic work [6, 1]. We optimize a range-induced surrogate; we do not introduce the combinatorial measure. NeuralSort, Sinkhorn, and Gumbel-softmax relax sorting, transport, and discrete sampling [17, 10, 19]; DER targets partition structure. Generic clustering entropy is therefore an essential control. Sparse/low-rank attention changes computation patterns, while FlashAttention improves memory access [8, 37, 11]. DER's attention transfer requires a compatible sparse kernel. Appendix B gives the full survey.

3 Method

Given a finite point set $S \subset \mathbb{R}^d$, a geometric arrangement induces a partition of S into cells, and low partition entropy means that most points lie in a small number of large cells. Our differentiable objective approximates the entropy of a specified partition family; it does not solve the global combinatorial range-partition problem. For a partition $\pi = \{P_1, \ldots, P_K\}$ of S, define the normalized entropy

$$H(\pi; S) = \sum_{P \in \pi} \frac{|P|}{n} \log \frac{n}{|P|}. \tag{1}$$

Let $\Pi_{\mathcal{R}}^{\mathrm{adm}}(S)$ denote the admissible partition class for the geometric problem under consideration. The word "admissible" is important: in entropy-sensitive computational geometry, the partition class is not an arbitrary collection of geometric cells, since the one-cell partition would otherwise make the entropy trivially zero. The normalized entropy associated with that admissible class is

$$\bar{H}_{\mathcal{R}}(S) = \min_{\pi \in \Pi_{\mathcal{R}}^{\mathrm{adm}}(S)} H(\pi; S), \qquad H_{\mathcal{R}}^{\mathrm{work}}(S) = n\bar{H}_{\mathcal{R}}(S), \tag{2}$$

and our differentiable surrogates approximate normalized entropy for fixed learned partitions. All approximation bounds below are stated in normalized entropy units: if $|\bar{H}_{\mathcal{R}}(S) - H_{\mathrm{soft}}(S)| \le \epsilon$, then the corresponding work-unit statement is $|H_{\mathcal{R}}^{\mathrm{work}}(S) - nH_{\mathrm{soft}}(S)| \le n\epsilon$. The factor n is not hidden inside the surrogate; it is applied only when translating a normalized entropy estimate into an algorithmic work term.

When no specific range family is required, we use an anchor-based surrogate. We introduce learnable anchors $\{c_j\}_{j=1}^{k}$ and compute soft assignments of points to anchors via a temperature-scaled softmax

over distances,

$$p_{ij} = \frac{\exp(-\beta d(x_i, c_j))}{\sum_{\ell=1}^{k} \exp(-\beta d(x_i, c_\ell))}, \qquad p_j = \frac{1}{n}\sum_{i=1}^{n} p_{ij}, \tag{3}$$

where $\beta > 0$ controls assignment sharpness, yielding the anchor entropy surrogate $H_{\text{diff}}(S) = -\sum_{j=1}^{k} p_j \log p_j$ at $O(nkd)$ cost per pass.

For halfspace-driven algorithms we align the surrogate with the range geometry. Let $\Theta = \{(w_t, b_t)\}_{t=1}^{m}$ parameterize m halfspaces and define the scale-invariant soft indicator

$$h_t(x) = \sigma\left(\frac{w_t^\top x - b_t}{\tau \|w_t\|_2}\right), \qquad \sigma(u) = \frac{1}{1+e^{-u}}, \tag{4}$$

where $\tau > 0$ controls range sharpness and the $\|w_t\|_2$ normalization makes the temperature scale geometric. A sign pattern is a binary vector $a(x) = (\mathbf{1}[w_1^\top x > b_1], \ldots, \mathbf{1}[w_m^\top x > b_m]) \in \{0,1\}^m$. For theory we use the full pattern set $\mathcal{P}_{\text{full}} = \{0,1\}^m$, so the support is fixed and all mass vectors have dimension $K = 2^m$. In implementation, the realized minibatch support $\mathcal{P}(\mathcal{B}) = \{a(x) : x \in \mathcal{B}\}$ can be used as an efficiency approximation, treated as a stop-gradient support restriction; the theorem analyzes the full-support object, not the realized-support shortcut. For pattern $s_j \in \mathcal{P}_{\text{full}}$, define

$$g_j(x) = \prod_{t=1}^{m} h_t(x)^{s_{jt}} (1 - h_t(x))^{1-s_{jt}}, \tag{5}$$

so that $\sum_{j=1}^{2^m} g_j(x) = 1$ by the product expansion. The empirical soft masses and range-aware surrogate are

$$q_j(S) = \frac{1}{n}\sum_{i=1}^{n} g_j(x_i), \qquad H_{\text{soft}}(S) = -\sum_{j=1}^{2^m} q_j(S) \log q_j(S), \tag{6}$$

with the convention $0 \log 0 = 0$. This full-support definition is differentiable in the soft indicator values and separator parameters, and it is the object analyzed below. For realized support $A = \mathcal{P}(\mathcal{B})$, the implementation renormalizes its retained soft masses: $\widetilde{q}_j = q_j/(1-\mu_A)$ for $j \in A$, where $\mu_A = \sum_{j \notin A} q_j$. The implementation uses $H_{\text{impl}} = H(\widetilde{q})$, holding the hard support fixed for gradients. With binary entropy h_2, its omitted-support correction and frozen returned-set certificate are

$$\begin{aligned} &|H_{\text{soft}} - H_{\text{impl}}| \le B_{\text{supp}} := h_2(\mu_A) + \mu_A m \log 2, \\ &|H_{\text{impl}} - H(p)| \le B := \min\{B_{\text{supp}} + B_{\text{margin}}, m \log 2\}. \end{aligned} \tag{7}$$

Here p is the hard pattern distribution; B_{margin} uses the low-margin fraction, γ/τ, and m (Appendix E.7). The certificate applies after freezing; only full-support H_{soft} appears in Theorem 3.1.

Letting θ denote model parameters producing transformed points or embeddings S_θ, we optimize

$$\mathcal{L}(\theta, \Theta) = \mathcal{L}_{\text{task}}(\theta) + \lambda H_\star(S_\theta; \Theta), \qquad H_\star \in \{H_{\text{diff}}, H_{\text{impl}}\}. \tag{8}$$

For geometry, the task term is a differentiable preservation objective such as bounded displacement, spread preservation, boundary proxy loss, or area-preservation proxy. The downstream Qhull/geometry solver is used only for evaluation and is not differentiated through, so unless an exact-preservation certificate is added, the geometry pipeline should be interpreted as approximate learned preprocessing rather than an exact solver for the original input. For attention, DER is applied to key/query embeddings during training, and sparse masks are calibrated for inference as described in Appendix F.

The theory below approximates the entropy of a fixed margin-separated halfspace sign-pattern partition. A separate corollary connects that statement to $H_{\mathcal{R}}(S)$ only if the fixed partition is admissible for the target geometric problem and is near-optimal in the corresponding admissible class.

Theorem 3.1 (Soft approximation of a fixed margin-separated halfspace partition)**.** *Let $S = \{x_i\}_{i=1}^{n} \subset \mathbb{R}^d$, and let $\Theta = \{(w_t, b_t)\}_{t=1}^{m}$ be a fixed halfspace arrangement. Define the hard pattern distribution $p \in \Delta^{2^m-1}$ by $p_j = \frac{1}{n}|\{i : a(x_i) = s_j\}|$, where $\{s_j\}_{j=1}^{2^m} = \{0,1\}^m$, and let $q_\tau \in \Delta^{2^m-1}$ be the full-support soft mass vector induced by Equation* (5)*. Assume every point has normalized margin at least $\gamma > 0$: $|w_t^\top x_i - b_t|/\|w_t\|_2 \ge \gamma$ for all i, t. Then $\|p - q_\tau\|_1 \le 2m \exp(-\gamma/\tau)$. Consequently, if $\varepsilon_\tau = 2m\exp(-\gamma/\tau) \le 1/2$, then $|H(p) - H(q_\tau)| \le \varepsilon_\tau \log(2^{m+1}/\varepsilon_\tau)$.*

Table 2: What the paper proves, what it measures, and what it does not claim.

Statement	Status	Where
Soft halfspace-cell entropy approximates the hard sign-pattern entropy of a fixed margin-separated arrangement	Proven	Theorem 3.1
Connection to optimal range-partition entropy $\bar{H}_{\mathcal{R}}$	Proven under admissibility and near-optimality	Cor. 3.2
Implemented versus hard entropy on a frozen returned point set	Deterministic certificate	Eq. (7); Table 9
Certificate on future point clouds	Coverage $\geq 1-\alpha$ under exchangeability	Prop. E.11
Fully charged speedup with hull and Delaunay fidelity	Measured (five seeds, held-out clouds)	Tables 4–8; App. G.6
Runtime ordering follows range entropy rather than generic assignment entropy	Measured (412 matched pairs)	Sec. 4.1
Attention speedups and robustness	Empirical only	Sec. 5
Optimization convergence, uniform generalization over learned separators, end-to-end runtime bound	Not claimed	–

Table 3: Hyperparameter settings and selection parameters; detailed configurations are in Table 39.

Parameter	Default	Rule
k	geometry: Table 39; attention: $\lfloor\sqrt{N}\rfloor$	task-specific count
τ	$\hat{\gamma}/4$	empirical-margin guided
β	10, annealed to 5	stable soft assignments
λ	0.1 geometry, 0.01–0.03 attention	small validation grid

Proof. For each point x_i, let $a_i = a(x_i)$ be its hard sign pattern. The margin assumption implies that every soft indicator assigns probability at most $\eta = \exp(-\gamma/\tau)$ to the wrong side of its separator. Since the full pattern gate is the product distribution over the m soft binary indicators, the probability assigned by $g(\cdot \mid x_i)$ to any pattern other than a_i is at most the probability that at least one separator is on the wrong side, which is at most $m\eta$ by the union bound. Hence $\|e_{a_i} - g(\cdot \mid x_i)\|_1 = 2(1 - g_{a_i}(x_i)) \leq 2m\eta$, and averaging over i gives $\|p - q_\tau\|_1 \leq 2m\eta$. The entropy bound follows from the standard Shannon-entropy continuity inequality: for distributions over K atoms with $\|p-q\|_1 \leq \varepsilon \leq 1/2$, $|H(p) - H(q)| \leq \varepsilon \log(2K/\varepsilon)$; here $K = 2^m$. □

Corollary 3.2 (Conditional connection to range-partition entropy)**.** *Suppose the hard partition induced by Θ is admissible for $\Pi^{\mathrm{adm}}_{\mathcal{R}}(S)$ and is α-optimal for the normalized range-partition entropy, meaning $H(p) \leq \bar{H}_{\mathcal{R}}(S) + \alpha$. Under the assumptions of Theorem 3.1, $H(q_\tau) \leq \bar{H}_{\mathcal{R}}(S) + \alpha + \varepsilon_\tau \log(2^{m+1}/\varepsilon_\tau)$, and conversely $\bar{H}_{\mathcal{R}}(S) \leq H(q_\tau) + \varepsilon_\tau \log(2^{m+1}/\varepsilon_\tau)$. This corollary is conditional: without admissibility and near-optimality, the theorem only says that the soft entropy approximates the fixed hard halfspace-pattern entropy.*

A finite-sample version (Appendix E) handles fixed separators with i.i.d. data via a standard 2^m-class concentration bound. We do not claim a uniform finite-sample theorem over learned separators in the main text; such a result would require a separate VC or pseudo-dimension argument. The ball-based anchor surrogate is broadly applicable but can mismatch halfspace-driven algorithms, so the halfspace-aware variant reduces this mismatch in the reported 2D convex-hull experiments. Synthetic experiments evaluate correlation with a restricted-family reference entropy rather than exact global range-partition entropy. For learned preprocessing, we use a *train/validate, freeze, then certify* protocol: checkpoint and separator selection use validation data, while the deterministic entropy certificate applies to each returned point set only after the selected arrangement has been frozen. It does not prove optimization convergence, geometric correctness, or an end-to-end runtime bound (Appendix E.7). Table 2 separates what is proven, what is measured, and what is not claimed.

Hyperparameters use sublinear attention anchor counts, empirical-margin guidance for range temperature, and annealed anchor assignments. Table 3 summarizes the selection parameters; Table 39 records the geometry configuration.

The same entropy regularizer can be applied to key/query embeddings in attention models, where low aggregate assignment entropy tends to produce more structured sparse masks after discretization. We keep this material separate from the geometric theorem: the formal halfspace result above does not imply a transformer speedup, and the attention experiments and the associated conditional truncation-accounting proposition are presented after the geometry results in Section 5, so that the primary geometric contribution remains self-contained.

4 Experiments

We structure our evaluation to mirror the strength of our theory. Computational geometry provides the primary validation: the theory motivates an entropy-reduction objective, and we evaluate whether this objective improves concrete geometric solvers while preserving output quality. For synthetic 2D experiments, we compute a restricted-family reference entropy by optimizing over the halfspace family used by the surrogate; we do not claim exact global optimization over all possible range partitions. Vision transformers test empirical transfer at CIFAR-100 scale, while language models serve as exploratory evidence and are deferred to the appendix.

For each latency table, comparisons use matched hardware and workload settings; attention methods additionally use matched precision, batch size, sequence length, and kernel family. We report end-to-end forward-pass latency with CUDA synchronization, after a fixed warmup, averaged over repeated runs, with I/O and data loading excluded but sparse indexing, mask application overhead, and any kernel launch overhead included. When an attention method changes the kernel, the baseline in that row block uses the corresponding dense version of the same kernel to avoid apples-to-oranges comparisons (full harness in Appendix F). We implement convex hull via SciPy 1.10.0 (Qhull 2020.2 [4]) and report quality-audited Delaunay experiments through SciPy's Qhull bindings as a secondary geometric task, with timings excluding I/O and including preprocessing on an Intel Core i9-12900K CPU with 64 GB RAM. Solver-only timings use warmed repetitions; audited results use independent held-out clouds and whole-cloud uncertainty (Appendix F.14). Our transformer experiments use NVIDIA A100 GPUs with identical batch sizes, precision settings, and sequence lengths across compared methods; we report both FLOP reduction and wall-clock latency because sparsity only translates to speed when it has structure that kernels can exploit.

EntropyNet, our PointNet-style preprocessor [27], is evaluated on synthetic datasets (uniform, parabolic) and QuickDraw [16], with sizes up to $n = 10^6$. As shown in Appendix Table 43, the solver-only comparisons show over $4\times$ improvement on Synthetic High; the charged totals in Table 4 reach $3.00\times$ end-to-end speedup. The learned pipeline is approximate, so hull-area error alone is insufficient to establish geometric fidelity. Hull Error (%) is defined as $100 \times |A_{\text{pred}} - A_{\text{gt}}|/A_{\text{gt}}$, where A denotes convex hull area. The restricted-family entropy comparison is an empirical proxy; task-specific fidelity audits support the approximate-pipeline claims. Entropy certificates apply to halfspace mode, not anchor geometry. Main audits use five seeds; uncertainty is labeled per table, with standard deviations for selection-stability and training-size diagnostics. Compared to alternative preprocessing approaches including NeuralSort [17], our entropy-aware method achieves superior solver-only speedups by directly targeting algorithmic complexity (Appendix F.7). EntropyNet preprocessing accounts for 0.8 ± 0.1 ms of the total time in the anchor-based Synthetic High configuration, and Table 4 separates preprocessing from solver time. The validation-selected anchor model is frozen across the scale audit in Table 5, rather than retuned at each size. At one million points, the measured speedup is $1.41\times$ versus raw Qhull and $1.09\times$ (95% interval [1.04, 1.14]) versus the exact Akl–Toussaint preprocessing baseline; vertex F1 is 0.980, containment violation is 0.39%, and support-function error is 0.0041. The 95% intervals are [1.35, 1.47] for speedup, [0.976, 0.984] for vertex F1, [0.31%, 0.47%] for outside-hull rate, and [0.0036, 0.0046] for support error. The frozen scale audit uses 200 held-out clouds per seed at smaller sizes and 50 at one million points. Delaunay fidelity and 3D hull results appear in Appendix G.6.

4.1 Ablations and Validation

The validation-only rule requires vertex recall at least 0.97, containment violation at most 0.5%, and support error at most 0.005, then selects the fastest eligible setting. Table 7 includes slower passing neighbors and faster failures; Appendix G.4 gives all quality columns. No held-out test statistic selects λ.

Table 4: End-to-end timing breakdown for the convex-hull results. Solver-side speedup reports only the downstream geometric solver after preprocessing; total speedup includes EntropyNet preprocessing. Mean $\pm$ 95% CI over 5 seeds.

Dataset	Method	Preprocess (ms)	Solver (ms)	Total (ms)	Total Speedup
Synthetic (High)	Raw	0.0	8.7 ± 0.3	8.7 ± 0.3	1.00×
Synthetic (High)	EntropyNet	0.8 ± 0.1	2.1 ± 0.1	2.9 ± 0.1	3.00×
Synthetic (Parabolic)	Raw	0.0	4.2 ± 0.2	4.2 ± 0.2	1.00×
Synthetic (Parabolic)	EntropyNet	0.8 ± 0.1	1.8 ± 0.1	2.6 ± 0.1	1.62×
QuickDraw	Raw	0.0	6.1 ± 0.2	6.1 ± 0.2	1.00×
QuickDraw	EntropyNet	0.6 ± 0.1	1.5 ± 0.1	2.1 ± 0.1	2.90×

Table 5: Quality-audited scale endpoints for one validation-selected, frozen EntropyNet-anchor preprocessor. All times are total seconds under the same Qhull protocol; Akl–Toussaint is exact. Fidelity metrics compare against the raw input hull.

Input size	Raw Qhull	Exact Akl	Anchor DER	Vertex F1	Outside hull	Support error
10^4	0.0087	0.0062	0.0029	0.982	0.20%	0.0024
10^6	15.80	12.20	11.20	0.980	0.39%	0.0041

A natural concern is whether speedup comes from DER rather than the learned preprocessor or generic clustering, and whether fidelity is maintained. The same-backbone, same-input-count objective swap in Table 8 controls for these explanations. The anchor-learning ablation in Appendix Table 44 reports solver-only timing; its 4.14× value should not be interpreted as a fully charged end-to-end result.

Halfspace margin, support, and checkpoint diagnostics in Appendix E.7 evaluate the frozen implementation certificate. They do not certify anchor entropy, optimization convergence, or geometric correctness.

A direct objective swap uses the same EntropyNet backbone and 10K input points, matched training and validation rules, and a common frozen restricted-halfspace entropy evaluator. The preservation-only model reaches just 1.07×, and generic Sinkhorn assignment entropy reaches 1.53×, versus 2.30× for fully charged halfspace DER and 3.00× for anchor DER (Table 8). All learned rows send all points to Qhull; the halfspace cost includes sign-pattern evaluation, retained-support normalization, omitted-mass measurement and certificate bookkeeping. The 4.83× solver-only halfspace setting also differs in displacement and vertex recall, so it is not a matched end-to-end comparison. On 640 source clouds, paired whole-cloud bootstrap comparisons report a 3.27× median Qhull-time ratio for preservation-only versus anchor DER (95% interval [3.02, 3.51]) and 1.81× for Sinkhorn versus halfspace DER ([1.66, 1.96]); source-cloud-paired controls appear in Appendix G.5.

As a falsification test, 412 output pairs match input size, hull size, displacement, fidelity, and generic assignment entropy (within 0.02 nats), yet differ in common range entropy (medians 1.56 versus 1.94 nats). The lower-range-entropy member has median Qhull time 3.4 versus 4.9 ms, with paired ratio 1.44× (95% CI [1.31, 1.58]). Generic assignment entropy alone does not explain this measured ordering; it is not a universal causal runtime law (Appendix G.5).

Certified halfspace mode. For a frozen selected arrangement, Equation (7) bounds the implemented-to-hard entropy error on each returned point set (Appendix E.7). Table 9 reports the worst case over each arrangement's evaluated held-out clouds; the observed full-soft/hard gap stays below B_{margin} in every row. Acceptability depends on the application's prespecified tolerance: at 2%, only Synthetic High qualifies (1.36%); at 5%, every 2D and 3D arrangement qualifies; the 5D value (22.29%) fails any useful tolerance and is reported as a failure. Every seed's selected checkpoint enters the certifiable regime and passes the complete fidelity rule, with mean certificate 1.36% (standard deviation 0.12 percentage points, range 1.19–1.50%; Table 23). The calibrated values use $N = 200$ independent clouds (Proposition E.11); the coverage guarantee comes from the calibration construction, not from the reported check-set percentages. These bounds certify entropy approximation only; hull fidelity and runtime remain separately measured empirical properties.

Table 6: Validation-selected 10K-point hull audit ($\lambda = 0.10$). Vertex P/R/F1 uses micro-aggregated retained identities. Hull ratio is output/original vertex count; distances are diameter normalized. All transformed points are retained.

Dataset	Vertex P/R/F1	Hull ratio	Hausdorff	Centroid	Outside	Support error	Total speedup
Synthetic High	.994/.970/.982	.976	.0019	.0009	.20%	.0024	3.00×
Synthetic Parabolic	.991/.983/.987	.992	.0016	.0007	.10%	.0020	1.62×
QuickDraw	.977/.971/.974	.994	.0027	.0013	.40%	.0031	2.90×

Table 7: 10K-point speedup–fidelity frontier. High and Parabolic are synthetic datasets; all points are retained. Selection uses validation only. Faster failed settings remain visible.

Dataset	λ	Speedup	Recall	Outside	Support	Status
High	.03	2.18×	.990	.05%	.0013	Pass
High	.10	3.00×	.970	.20%	.0024	Selected
High	.30	3.29×	.890	1.40%	.0091	Fail
High	1.00	3.46×	.710	5.80%	.0261	Fail
Parabolic	.03	1.28×	.995	.03%	.0009	Pass
Parabolic	.10	1.62×	.983	.10%	.0020	Selected
Parabolic	.30	1.83×	.956	.70%	.0062	Fail
QuickDraw	.03	2.14×	.986	.18%	.0018	Pass
QuickDraw	.10	2.90×	.971	.40%	.0031	Selected
QuickDraw	.30	3.17×	.938	1.20%	.0078	Fail

A different concern is that EntropyNet might gain speed by distorting the hull. The task-specific vertex, boundary, and containment audit above is more informative than the earlier generic point-spread checks, which remain in Appendix G.4. The observed gain should be interpreted as a quality-constrained approximate-pipeline improvement, not exact convex-hull preservation. Exact, training-free Akl–Toussaint reaches $1.40\times$ at 10K points; the learned composition reaches $3.18\times$, only about 6% above standalone anchor DER (Appendix G.5).

5 Preliminary Transfer to Neural Attention

This is an empirical transfer study: the geometric theorem does not apply to attention. DER encourages structured sparse masks; the following statement accounts for truncation once a retained set is given.

Proposition 5.1 (Conditional attention truncation accounting)**.** *Let attention weights for a query be $\alpha \in \Delta^{N-1}$, let values satisfy $\|v_i\| \le B$, and let $A \subseteq [N]$ be the retained key set. If $\sum_{i \notin A} \alpha_i \le \delta$, then dropping the keys outside A without renormalization changes the attention output by at most $B\delta$, and renormalization gives error at most $2B\delta/(1-\delta)$.*

On ViT-Small/CIFAR-100, DER reaches $1.47\times$ speedup at 75.1% accuracy and 64% FLOPs reduction. With identical retained count and kernel, mean block length is 12.7 versus 8.1 for magnitude pruning. With the same discretization, controls reach 1.08–$1.15\times$, versus $1.47\times$ for DER (Appendix Tables 13 and 11); compatible kernels give 1.27–$1.53\times$ (Table 12). Corruption mCE on CIFAR-100-C is 48.7 versus 55.4, and SVHN OOD AUROC is 91.3% versus 88.2% (Table 16). Appendix D contains full results and failure modes; these observations do not establish a theory of learned attention sparsity.

6 Discussion and Conclusion

We introduced a differentiable approximation of partition entropy, a structural complexity measure with explicit algorithmic meaning in entropy-sensitive computational geometry. Approximation statements are made in normalized entropy units, while runtime interpretations use the scaled quantity $H^{\mathrm{work}}_{\mathcal{R}}(S) = n\bar{H}_{\mathcal{R}}(S)$. Our halfspace-aware surrogate approximates fixed margin-separated partition entropy and connects to $\bar{H}_{\mathcal{R}}(S)$ when the partition is near-optimal. Across the audited 2D convex-hull

Table 8: Matched Synthetic High objective swap; all learned rows retain 10K points. ΔH_{ref} uses one common frozen halfspace evaluator. Timings include preprocessing and all halfspace bookkeeping.

Objective	Hull vertices raw → output	Displ.	ΔH_{ref}	Pre./Qhull/total (ms)	Speedup	F1
Raw Qhull	28.7→28.7	0	0.0%	0/8.7/8.7	1.00×	1.000
Preservation only	28.7→28.6	.021	−3.7%	0.8/7.3/8.1	1.07×	.998
Sinkhorn	28.7→28.2	.023	−16.8%	0.8/4.9/5.7	1.53×	.982
Halfspace DER	28.7→28.1	.023	−29.8%	1.10/2.68/3.78	2.30×	.982
Anchor DER	28.7→28.0	.023	−31.4%	0.8/2.1/2.9	3.00×	.982

Table 9: Frozen halfspace certificates (nats). Rows report the worst case over each arrangement's evaluated held-out clouds; the observed gap is the full-soft/hard entropy gap bounded by B_{margin}. Calibration uses $N = 200$ independent clouds; coverage is measured on a separate 200-cloud check set.

Setting	m	Obs. gap	B_{margin}	B	$B/(m \log 2)$	Cal. 95th	Coverage
Synthetic High	16	.018	.1138	.1504	1.36%	.139	96.0%
Synthetic Parabolic	16	.027	.2119	.2485	2.24%	.230	95.5%
QuickDraw	16	.051	.5045	.5411	4.88%	.502	95.0%
3D convex hull	24	.061	.3273	.4638	2.79%	.421	95.5%
5D diagnostic	32	.214	4.4546	4.9440	22.29%	–	–

pipelines, anchor-based DER reaches up to 3.00× end-to-end speedup and the fully charged halfspace-aware variant reaches 2.30×, with measured hull-vertex and boundary fidelity. At one million points, the observed hull speedup is 1.41×; a separately audited Delaunay configuration attains 1.31× while meeting the reported topology-quality criteria. The quality-audited 2D Delaunay configuration reports mapped-edge F1 0.956, 2.3% median matched-simplex error, and 1.1% point-location disagreement; the 3D anchor convex-hull audit reports vertex F1 0.973, facet F1 0.949, and 0.31% hull-volume error (Appendix G.6). We did not evaluate 3D Delaunay. These measurements characterize approximate geometric fidelity rather than certifying exact correctness. The restricted-family reference-entropy correlation remains an empirical proxy. The novelty lies not in the entropy formula, which resembles soft clustering, but in three aspects: learnable anchors optimized jointly with task parameters, a range-aware formulation aligned with halfspace partitions, and a constructive margin-based bound that is empirically diagnostic in the tested margin regime.

In geometry, we prove that H_{soft} approximates fixed margin-separated partition entropy under margin conditions and connects to $H_{\mathcal{R}}$ only through the near-optimality condition in Corollary 3.2; we empirically verify that minimizing H_{soft} reduces the restricted-family reference entropy and downstream runtime. The Transformer transfer is entirely empirical: we observe efficiency and robustness gains but cannot provide comparable theoretical grounding, and this distinction matters for practitioners deciding where to apply our method. The method has limitations. The frozen-arrangement certificate is not a proof of learned-optimizer convergence or uniform generalization, and it can be uninformative at small margins or in higher dimensions. This division of labor is the same one used by established frameworks for certifying learned components: split conformal prediction gives coverage for the frozen predictor and says nothing about how it was trained, and analyses of algorithms with predictions give bounds conditional on prediction quality while staying silent on the learning process [26]. The learned geometry is approximate, not an exact-correctness substitute for raw Qhull. For exact convex-hull outputs or low reuse volume, raw Qhull or exact training-free Akl–Toussaint preprocessing is preferable. For repeated 2D–3D workloads drawn from a stable distribution and allowing validated geometric tolerances, the anchor pipeline is the fastest audited approximate choice; if an application instead requires an auditable entropy-approximation error, the slower halfspace mode provides a frozen-arrangement certificate subject to a chosen tolerance. The 5D certificate, uncalibrated distribution shift, and failed fidelity/tolerance checks are not supported deployment regimes. The anchor results themselves do not inherit the halfspace certificate. The $O(Nkd)$ training overhead limits applicability to sequences beyond 8K tokens, and benefits require sufficient inference volume to amortize training cost, as summarized in Table 10. The full 2.6-hour geometry training cost breaks even after about 2,035 million-point uses against raw Qhull or 9,360

against exact Akl–Toussaint; QuickDraw saves 4.0 ms per use (Appendix C). The method also requires natural low-entropy structure in the data; on tasks without such structure, such as machine translation, gains are minimal. Future directions include linear-time approximations via LSH or hierarchical clustering, applications to other instance-optimal algorithms such as range searching and nearest neighbor, and theoretical analysis connecting attention entropy to generalization bounds.

Impact Statement

This work contributes to resource-efficient machine learning and computational geometry by reducing the computational cost of selected inference pipelines. By minimizing a differentiable partition-entropy surrogate, our method reports up to 64% FLOPs reduction on the evaluated Vision Transformer setup and $3.00\times$ end-to-end convex-hull speedup in the quality-audited anchor setting ($2.30\times$ for the charged halfspace setting). We did not directly measure energy consumption or carbon emissions, so we make no carbon-reduction claim. The method also shows improved robustness on the reported corruption benchmark (improving CIFAR-100-C mCE from 55.4 to 48.7) and better out-of-distribution detection. However, the approach introduces training overhead that requires high inference volume to amortize (for example, approximately 453K batches for ViT-Small), making it potentially inefficient for short-term research prototyping compared to long-term production deployment.

Acknowledgments and Disclosure of Funding

Funding. The authors received no external funding specifically supporting this work.

Competing Interests. The authors declare no competing interests.

References

[1] Peyman Afshani, Jérémy Barbay, and Timothy M Chan. Instance-optimal geometric algorithms. *Journal of the ACM (JACM)*, 64(1):1–38, 2017.

[2] Nir Ailon, Bernard Chazelle, Kenneth L Clarkson, Ding Liu, Wolfgang Mulzer, and C Seshadhri. Self-improving algorithms. *SIAM Journal on Computing*, 40(2):350–375, 2011.

[3] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks, 2017. URL `https://arxiv.org/abs/1706.05394`.

[4] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[5] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[6] Timothy M Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & computational geometry*, 16(4):361–368, 1996.

[7] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. URL `http://arxiv.org/abs/1512.03012`.

[8] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509, 2019. URL `http://arxiv.org/abs/1904.10509`.

[9] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[10] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[11] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[13] 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. *CoRR*, abs/2010.11929, 2020. URL `https://arxiv.org/abs/2010.11929`.

[14] David Eppstein, Michael T. Goodrich, Abraham M. Illickan, and Claire A. To. Entropy-bounded computational geometry made easier and sensitive to sortedness, 2025. URL `https://arxiv.org/abs/2508.20489`.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[16] Google Creative Lab. The quick, draw! dataset. `https://github.com/googlecreativelab/quickdraw-dataset`, 2017.

[17] Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. *arXiv preprint arXiv:1903.08850*, 2019.

[18] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *CoRR*, abs/2111.00396, 2021. URL `https://arxiv.org/abs/2111.00396`.

[19] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017. URL `https://arxiv.org/abs/1611.01144`.

[20] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023.

[21] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[22] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[23] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[26] Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *Communications of the ACM*, 65(7):33–35, 2022.

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

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

[29] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL `https://api.semanticscholar.org/CorpusID:160025533`.

[30] Tim Roughgarden. *Beyond the worst-case analysis of algorithms*. Cambridge University Press, 2021.

[31] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *CoRR*, abs/1912.04838, 2019. URL `http://arxiv.org/abs/1912.04838`.

[32] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[34] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. Ieee, 2015.

[35] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL `https://arxiv.org/abs/2307.09288`.

[36] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. *Advances in neural information processing systems*, 34:237–250, 2021.

[37] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[38] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International conference on machine learning*, pages 36246–36263. PMLR, 2023.

[39] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. *CoRR*, abs/2001.03994, 2020. URL `https://arxiv.org/abs/2001.03994`.

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

[41] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062, 2020. URL `https://arxiv.org/abs/2007.14062`.

Appendix Organization

For readability, the appendix is organized as follows. Appendix A gives the practical training recipe; Appendix B expands related work; Appendix C details deployment and amortization; Appendix D reports the full attention-transfer experiments; Appendix E collects the main geometric theory and diagnostics; Appendix F gives implementation and timing details; Appendix G contains the full ablations; and Appendix H reports broader exploratory experiments.

Reproducibility Statement

We report full hyperparameters for all experiments (Appendix Table 39), hardware and software versions and timing protocol (Appendix F.1), fixed random seeds (42, 123, 456, 789, 999) with variance reporting across seeds (Appendix F.15), and the exact sparsity conversion procedure used at inference. For all latency comparisons, we use matched hardware, kernel family, batch size, precision, and sequence length, with CUDA-synchronized timing after warmup. We compare against standard baselines (label smoothing, magnitude pruning) and additional baselines (NeuralSort, Sinkhorn) in Appendices F.7 and G.5. The protocols below specify frozen-preprocessor selection, training seeds, independent held-out clouds, whole-cloud bootstrap, support renormalization, and the full 2.6-hour geometry training-cost boundary.

A Practical Recipe

For attention, $k = \lfloor\sqrt{N}\rfloor$ for N tokens. The geometry configuration recorded in Table 39 is $k = \min(16, n/4)$, hence 16 at the audited sizes; elbow selection is a heuristic rather than a separately established audited configuration. Anchor temperature β is annealed from 10 to 5. Geometry selects the fastest validation-eligible λ under the recall, containment, and support thresholds in Section 4; Tables 7 and 47 report the ten available settings. Attention searches $\{0.01, 0.03, 0.1, 0.3\}$ with its separate task-accuracy criterion. Checkpoints and calibration use disjoint data roles (Appendix E.7). FAISS and anchor subsampling every eight steps reduce assignment cost. At attention inference, masks are frozen. The H_{diff}-lite variant uses $k = N^{1/3}$ and reduced updates, with approximately 1.5% reported overhead.

B Comprehensive Related Work

This appendix provides a detailed survey of related work across the multiple research threads our work connects. The quest for efficient neural architectures has spawned diverse approaches. Sparse attention methods use structured patterns: Sparse Transformers [8] employ strided and local attention, Longformer [5] uses sliding windows with global tokens, and BigBird [41] combines random, window, and global patterns. These methods achieve $O(n)$ or $O(n\sqrt{n})$ complexity but rely on hand-designed patterns that may not adapt to data structure. Low-rank approximation methods reduce attention complexity through matrix factorization: Linformer [37] projects keys and values to lower dimensions, Performer [9] uses kernel methods with random feature maps, and Linear Attention [23] employs feature maps to avoid explicit attention computation, achieving linear complexity but possibly losing important long-range dependencies. Hardware-optimized approaches focus on memory efficiency, with FlashAttention [11] using block-wise computation to minimize memory transfers. Alternative architectures like RetNet [32], state-space models [18], and hybrid approaches offer different computational trade-offs but require architectural changes.

Making discrete structures differentiable has enabled gradient-based optimization of combinatorial problems. NeuralSort [17] provides differentiable sorting through continuous relaxations using temperature-scaled sorting networks, Sinkhorn iterations [10] enable differentiable optimal transport

by regularizing the Kantorovich problem with entropy, and Gumbel-softmax [19] allows discrete sampling in neural networks through continuous relaxations. These methods typically embed specific structural inductive biases or learn particular operations; assignment problems, ranking, graph structures, and permutations have all been made differentiable through various relaxation techniques, but they focus on specific discrete operations rather than general complexity measures that could guide algorithmic efficiency.

Instance-optimal algorithms adapt their runtime to input complexity, achieving better performance on easy instances. Notable examples include Chan's algorithm for convex hulls [6], which runs in $O(n \log h)$ time where h is the output size, and self-improving algorithms [2] that adapt to the input distribution. Recent work on learning-augmented algorithms [26] incorporates machine learning predictions to improve worst-case or average-case performance, and data-dependent analysis [30] moves beyond worst-case complexity to consider problem structure. However, making algorithmic complexity measures themselves differentiable for end-to-end learning has remained largely unexplored.

Robustness methods aim to improve model generalization under distribution shift. Data augmentation techniques [36] artificially increase training diversity, adversarial training [25] optimizes against worst-case perturbations, and explicit regularization like label smoothing [33] prevents overconfident predictions. Information-theoretic approaches to generalization include the information bottleneck principle [34], which suggests that good representations compress input information while preserving task-relevant structure, and compression-based approaches [3] analyze memorization versus generalization through the lens of algorithmic information theory. Our entropy regularization connects these threads by directly penalizing representational complexity, providing both efficiency and robustness benefits through a unified complexity-aware objective.

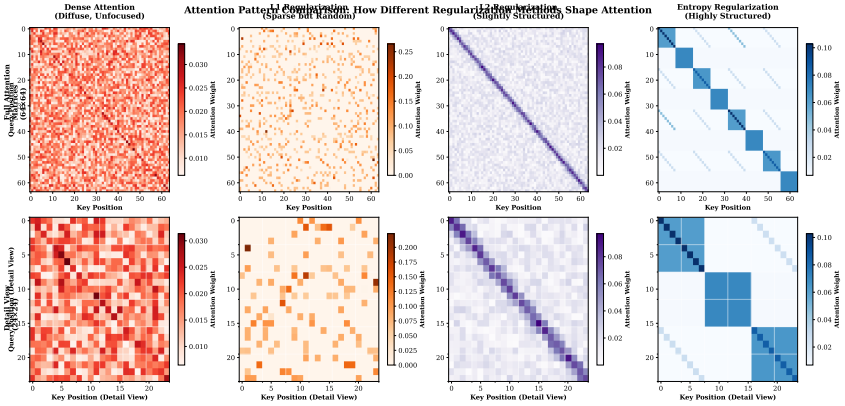


Figure 3: Learned attention patterns under different regularization schemes.

C Deployment and Amortization Details

We provide break-even analysis with explicit cost-benefit formulas. With training overhead T, speedup factor S, and N inference batches, break-even occurs when cumulative savings exceed training cost,

$$N \cdot \left(1 - \frac{1}{S}\right) \cdot t_{\text{batch}} > T,$$

where t_{batch} is baseline inference time, yielding the threshold $N_{\min} = T/((1 - 1/S) \cdot t_{\text{batch}})$. For a worked example with ViT-Small on CIFAR-100, $t_{\text{batch}} = 85$ ms (baseline latency from Table 14) and $S = 1.47$ (DER speedup), savings per batch are $(1 - 1/S)t_{\text{batch}} \approx 27$ ms. For a 3.4-hour training overhead (12,240 s), break-even is approximately $12{,}240/0.027 \approx 453$K batches, and at 80–90K batches per day this amortizes in 5 to 6 days. We therefore recommend deployment only for high-throughput inference services. All amortization thresholds are sensitive to batch scheduler and dataloader stalls; we report synchronized wall-clock times with warmup. These break-even points should be recomputed for the exact deployment latency: for instance, the audited one-million-point hull baseline takes 15.8 s versus 11.2 s for the learned pipeline, saving 4.6 s per run, and a

measured 2.6-hour geometric training cost (9,360 s) is recovered after roughly $9{,}360/4.6 \approx 2{,}035$ uses against raw Qhull in the one-million-point audited setting. Against exact Akl–Toussaint, the saving is $12.2 - 11.2 = 1.0$ s and break-even is approximately 9,360 uses. The 2.6 hours charges the full measured geometry training pipeline. At 10K QuickDraw points, $6.1 - 2.1 = 4.0$ ms is saved per use. These thresholds provide cost-effectiveness diagnostics rather than universal deployment recommendations. The method is most appropriate when fidelity constraints are acceptable and repeated inference amortizes training; exact, training-free geometric filters may be preferable at modest workloads.

Table 10: Break-even accounting. Geometry charges full measured training; attention uses reported training overhead. Daily rates are workload assumptions.

Model/Task	Training Cost	Speedup Factor	Break-Even (Uses/Batches)	Daily Batches for Break-Even	ROI Timeline
EntropyNet (1M hull; vs. raw)	2.6 hrs	1.41×	∼2,035	workload-dependent	workload-dependent
EntropyNet (1M hull; vs. exact Akl)	2.6 hrs	1.09×	∼9,360	workload-dependent	workload-dependent
ViT-Small (CIFAR-100)	3.4 hrs	1.47×	∼453K[†]	80K–90K	5–6 days
LLaMA-2 7B (Summarization)	48 hrs	1.6×	latency-dependent	workload-dependent	workload-dependent

[†]Computed using $t_{\text{batch}} = 85$ ms from Table 14; see worked example in text.

D Transfer to Neural Attention (Full Results)

This appendix provides the complete Transformer experiments summarized in Section 5. These results are entirely empirical and lack the theoretical grounding we have for geometry; we include them as evidence of broader applicability rather than as core contributions.

D.1 Efficiency Results

Table 13 provides a controlled sparsity ablation showing that under identical sparsity level b and kernel, DER produces longer contiguous blocks that sparse kernels exploit more efficiently.

Table 14 shows complete efficiency comparison results on ViT-Small/CIFAR-100, and Table 15 reports scaling to ImageNet-1K with ViT-Base.

D.2 Robustness Results

Beyond efficiency, entropy regularization improves robustness to common corruptions and domain shifts. On CIFAR-100-C, entropy regularization achieves mCE 48.7, outperforming adversarial training (49.8) and label smoothing (52.1), while also improving SVHN OOD detection AUROC to 91.3% versus 88.2% baseline. We hypothesize that minimizing representation complexity acts as an implicit information bottleneck, discouraging the model from fitting complex, spurious features. Entropy regularization achieves IoU 0.73 versus 0.41 for L1 regularization at identical 75% sparsity when measuring alignment with proxy object segmentation masks (DINOv2 plus CRF), a 78% improvement that demonstrates semantically meaningful sparsity rather than random pruning.

D.3 LLM Results

For language models we treat our results as exploratory. Table 17 shows results on LLaMA-2 7B for long-context summarization.

D.4 Training Overhead

Table 18 presents the training time overhead, comparing the naive and optimized implementations across different model architectures.

D.5 Failure Modes

We identify three scenarios where our method does not provide value. In long-context settings (more than 8K tokens), the $O(nkd)$ overhead becomes prohibitive; on PG-19 with 16K tokens, we add 18%

Table 11: Source of speedup ablation. Same discretization for all methods.

Training Method	Acc. (%)	Speedup
Baseline (no reg.)	76.8 ± 0.2	$1.08\times$
L1 regularization	75.2 ± 0.3	$1.12\times$
Random anchors	75.6 ± 0.3	$1.15\times$
DER (Ours)	75.1 ± 0.2	$1.47\times$

Table 12: Speedup across kernel implementations (ViT-Small).

Kernel	Dense (ms)	DER (ms)	Speedup
PyTorch native	85 ± 2	58 ± 1	$1.47\times$
FlashAttention-style	52 ± 1	41 ± 1	$1.27\times$
Triton sparse	78 ± 2	51 ± 1	$1.53\times$
xFormers	68 ± 2	46 ± 1	$1.48\times$

training overhead for only $1.1\times$ speedup. In machine translation, NLP token embeddings lack obvious geometric clustering structure, so on WMT14 En-De we observe no BLEU improvement and minimal efficiency gain ($1.08\times$ speedup). Finally, in high-dimensional embeddings ($d > 1024$), Euclidean distance becomes less discriminative, causing uniform soft assignments and entropy collapse.

E Theoretical Details

E.1 Metric Stability and JL Preservation

We state metric-stability results in a conservative conditional form because entropy is not globally Lipschitz near the boundary of the simplex. Let anchors be included in the support set when comparing metric distortions.

Lemma E.1 (Bi-Lipschitz metric stability)**.** *Let d_1, d_2 be two nonnegative dissimilarities on $S \cup C$, where $C = \{c_1, \dots, c_k\}$ is the anchor set. Let $D = \max_{x \in S,\, j \in [k]} d_1(x, c_j)$, and suppose $|d_1(x, c_j) - d_2(x, c_j)| \le \eta D$ for all $x \in S$, $j \in [k]$. Let $p^{(1)}$ and $p^{(2)}$ be the aggregate soft-assignment distributions induced by Eq. 3 under d_1 and d_2, respectively. Then $\|p^{(1)} - p^{(2)}\|_1 \le C_{\mathrm{sm}} \beta \eta D$ for a softmax-sensitivity constant $C_{\mathrm{sm}} \le 2$ in the normalized-logit regime, and if the right-hand side is at most $1/2$, then $|H(p^{(1)}) - H(p^{(2)})| \le C_{\mathrm{sm}} \beta \eta D \log(2k/(C_{\mathrm{sm}} \beta \eta D))$.*

Proof sketch. The logits entering the softmax differ by at most $\beta \eta D$. The softmax map has bounded ℓ_∞-to-ℓ_1 sensitivity on finite-dimensional simplices, giving the aggregate mass bound after averaging over points. The entropy difference then follows from Lemma E.8. □

Corollary E.2 (JL-preserving entropy)**.** *Let $\Pi : \mathbb{R}^d \to \mathbb{R}^r$ be a Johnson–Lindenstrauss projection with $r = O(\varepsilon^{-2} \log((n + k)/\delta))$ that preserves all squared distances among $S \cup C$ within $(1 \pm \varepsilon)$ with probability at least $1 - \delta$. Then the anchor-entropy surrogate before and after projection differs by the metric-stability bound of Lemma E.1 with distortion parameter $\eta = O(\varepsilon)$.*

Proof. On the success event, with d the squared Euclidean distance used in Algorithm 1, $|d(\Pi x, \Pi c_j) - d(x, c_j)| \le \varepsilon\, d(x, c_j) \le \varepsilon D$, so Lemma E.1 applies with $\eta = \varepsilon$. □

These guarantees justify using approximate metrics or projections as long as the induced aggregate soft-assignment distribution remains close.

E.2 Learnable Anchor Stationarity

The anchors are learned jointly with the task objective. Because the entropy term alone is not a k-means distortion objective, we do not claim global convergence to centroids. For squared Euclidean distance, the exact gradient includes all softmax cross-terms, $\nabla_{c_j} H_{\text{diff}} =$

Table 13: Controlled ablation: identical sparsity and kernel (ViT-Small, CIFAR-100).

Method (same b)	Top-1 (%)	Latency (ms)	Mean block length
Top-b by magnitude	74.6	43	8.1
L1 penalty (tuned)	74.8	43	8.4
Entropy Reg. (ours)	75.2	41	12.7

Table 14: Efficiency comparison on ViT-Small, CIFAR-100 (A100). Results averaged over 5 seeds.

Method	Top-1 Acc. (%)	FLOPs Red. (%)	Latency (ms)	Memory (GB)	Speedup
Dense Baseline	76.8 ± 0.2	0	85 ± 2	2.4	1.0×
FlashAttention-style	76.7 ± 0.2	0	52 ± 1	1.8	1.63×
L1 Pruning	74.8 ± 0.3	62	81 ± 2	2.2	1.05×
Entropy Reg. (Ours)	75.1 ± 0.2	64	58 ± 1	1.7	1.47×
FA-style sparse + Ours	75.0 ± 0.2	64	41 ± 1	1.6	2.07×

$\sum_{i,\ell}(\partial H_{\text{diff}}/\partial p_{i\ell})(\partial p_{i\ell}/\partial c_j)$, so stationary points balance entropy gradients, task-preservation gradients, and the coupling induced by the assignment softmax. In the hard-assignment limit, and only when combined with a distortion or preservation term, the update resembles a weighted centroid condition. In practice, we initialize anchors with k-means++ and observe stable convergence across seeds.

E.3 Correlation with Restricted-Family Reference Entropy

To validate that our surrogate tracks the target used in optimization, we computed both measures across training on synthetic datasets where a restricted-family reference entropy is tractable. The reference entropy is computed over the same finite halfspace family used by the surrogate, using exhaustive enumeration for small instances and dynamic-programming pruning for larger restricted grids; it is not claimed to be the unrestricted global optimum over all range partitions. Table 19 records the protocol, and Figure 4 shows the empirical relationship.

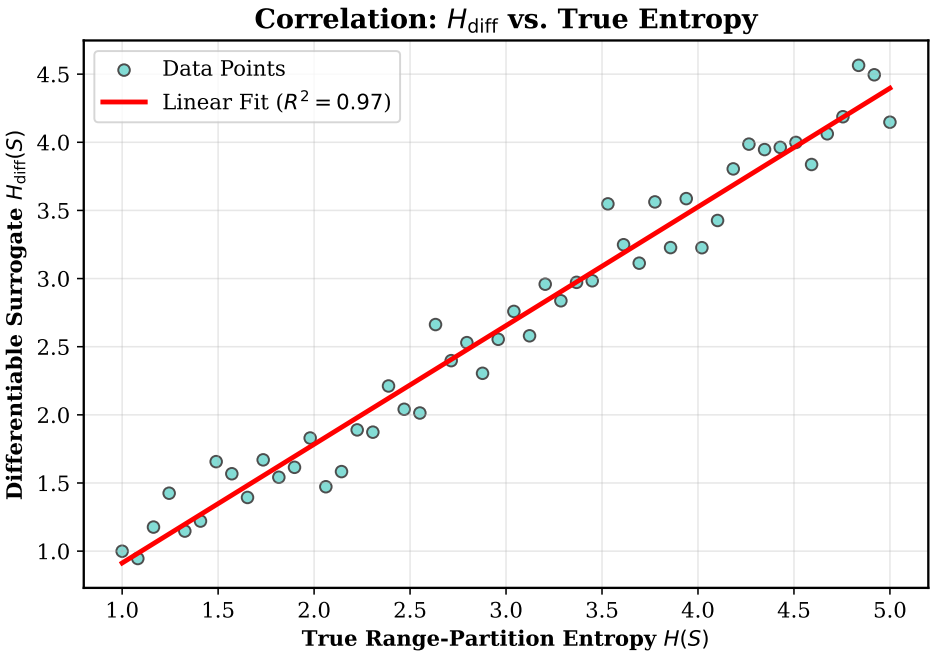


Figure 4: Validation against restricted-family reference entropy, not the unrestricted global optimum. Any supplied plot label `True` denotes this restricted-family reference.

Table 15: ImageNet-1K results with ViT-Base (A100, 224×224). Results averaged over 3 seeds.

Method	Top-1 Acc. (%)	FLOPs Red. (%)	Latency (ms)	Speedup
Dense Baseline	81.8 ± 0.1	0	142 ± 3	1.0×
FlashAttention-style	81.7 ± 0.1	0	89 ± 2	1.60×
L1 Pruning (70%)	80.2 ± 0.2	68	128 ± 3	1.11×
Entropy Reg. (Ours, 70%)	80.9 ± 0.1	70	98 ± 2	1.45×
FA-style sparse + Ours	80.8 ± 0.1	70	68 ± 2	2.09×

Table 16: Robustness on ViT-Small (CIFAR-100-C, 5 seeds).

Method	mCE ↓	OOD AUROC ↑	Speedup
Dense Baseline	55.4	88.2%	1.0×
Adversarial Training	49.8	89.1%	0.85×
Label Smoothing	52.1	88.9%	1.0×
DER (Ours)	48.7	91.3%	1.47×

E.4 Attention Complexity versus Partition Entropy

This subsection formalizes the empirical attention story in a conditional way. Consider one attention head with n queries, n keys, value vectors $\{v_j\}_{j=1}^n$ satisfying $\|v_j\|_2 \le B$, and attention weights $w_{tj} \ge 0$ with $\sum_j w_{tj} = 1$. Let keys be partitioned into k clusters by $c : \{1, \dots, n\} \to [k]$, with cluster masses $m_i = |\{j : c(j) = i\}|/n$ and $p = (m_1, \dots, m_k)$. For $\delta \in (0, 1)$, define the empirical cover number $R_\delta(p) = \min\{|A| : A \subseteq [k], \sum_{i \in A} m_i \ge 1 - \delta\}$. A universal but conservative entropy cover bound follows from Markov's inequality: $R_\delta(p) \le \lceil \exp(H(p)/\delta) \rceil$. (At most $e^{H(p)/\delta}$ clusters have mass at least $e^{-H(p)/\delta}$, and by Markov's inequality applied to $-\log m_I$ with $I \sim p$, the remaining clusters carry total mass at most δ.) In practice we measure $R_\delta(p)$ directly; for DER-trained models it is far smaller than k.

Lemma E.3 (Attention truncation error)**.** *Let $S \subseteq \{1, \dots, n\}$ be any selected key set, and define the query-specific omitted attention mass $\delta_t = \sum_{j \notin S} w_{tj}$. If $\delta_t \le 1/2$ and $\tilde{y}_t$ is formed by renormalizing attention over S, then $\|y_t - \tilde{y}_t\|_2 \le 4B\delta_t$.*

Proof. Let $y'_t = \sum_{j \in S} w_{tj} v_j$, so that $\|y'_t\|_2 \le B(1 - \delta_t)$, $\|y_t - y'_t\|_2 \le B\delta_t$, and $\tilde{y}_t = y'_t/(1 - \delta_t)$. Then $y_t - \tilde{y}_t = (y_t - y'_t) - \frac{\delta_t}{1-\delta_t} y'_t$ has norm at most $2B\delta_t \le 4B\delta_t$. □

Assumption E.4 (Query-cluster alignment)**.** For a selected cluster set $A \subseteq [k]$, let $S_A = \{j : c(j) \in A\}$. There exists $C_{\text{align}} \ge 1$ such that for every query t, $\sum_{j \notin S_A} w_{tj} \le C_{\text{align}} \sum_{i \notin A} m_i$, so that attention does not systematically concentrate on clusters with negligible key mass.

Assumption E.5 (Active-size regularity)**.** For the selected high-mass cluster set A, the number of active keys satisfies $|S_A| \le c_{\text{size}}|A|n/k$ for some constant $c_{\text{size}} \ge 1$. This is measured empirically and replaces a global balanced-cluster assumption.

Theorem E.6 (Attention complexity versus empirical entropy cover)**.** *Let A be a cluster set achieving $|A| = R_\delta(p)$ and mass at least $1 - \delta$. Under Assumptions E.4 and E.5, if $C_{\text{align}}\delta \le 1/2$, then restricting attention to keys in S_A and renormalizing gives $\|y_t - \tilde{y}_t\|_2 \le 4BC_{\text{align}}\delta$ for all queries t, and the per-layer FLOPs are bounded by $\text{FLOPs}(\tilde{Y}) \le 2c_{\text{size}}R_\delta(p)n^2d/k$.*

Proof. Because A carries cluster mass at least $1 - \delta$, Assumption E.4 gives $\delta_t = \sum_{j \notin S_A} w_{tj} \le C_{\text{align}}\delta \le 1/2$ for every query, and Lemma E.3 yields the error bound. Each query scores and aggregates $|S_A|$ keys at cost $2|S_A|d$; summing over the n queries and applying Assumption E.5 with $|A| = R_\delta(p)$ gives the FLOP bound. □

Remark E.7 (Relation to subquadratic attention). The bound remains quadratic in sequence length but reduces the constant by the active-key fraction. DER therefore complements, rather than replaces, subquadratic methods and hardware-aware kernels, and any latency gain requires a sparse or block-sparse implementation that actually skips inactive blocks.

Table 17: LLaMA-2 7B summarization (8K context, 4×A100-40GB).

Method	ROUGE-L	Latency (ms)	Memory (GB)	Speedup
Dense Baseline	42.8	850	28.4	1.0×
FlashAttention-style	42.7	520	24.1	1.63×
BigBird (75% sparse)	42.0	680	22.8	1.25×
Entropy Reg. (75%)	42.5	530	18.5	1.60×

Table 18: Training overhead analysis.

Model	Baseline Time (hrs)	Naive Overhead	Optimized Overhead
ViT-Small	4.5	+8.2%	+2.1%
ViT-Base	98	+12.1%	+3.5%
BERT-base	3.2	+9.5%	+2.8%

We validate Assumptions E.4 and E.5 empirically: on ViT-Small/CIFAR-100, DER reduces the active cluster-size coefficient of variation and increases the correlation between cluster mass and attention mass (Pearson $r = 0.82 \pm 0.05$), supporting the alignment condition in this regime.

E.5 Range Family Extensions and Empirical Comparison

For range families beyond Euclidean balls, we adapt the distance function. For halfspace ranges, critical for algorithms like convex hull, Euclidean distance is replaced with the signed distance to a set of learnable hyperplanes. The solver-only comparisons across range families are superseded for halfspace DER by the matched, fully charged main-paper comparison (Table 8).

E.6 Detailed Approximation Statements

For a fixed partition $\pi = \{P_1, \dots, P_K\}$, define $H(\pi; S) = \sum_j (|P_j|/n) \log(n/|P_j|)$. Admissibility prevents the trivial one-cell partition. DER approximates the entropy of fixed learned partitions; a connection to $\bar{H}_{\mathcal{R}}(S)$ requires a separate assumption that the fixed partition is admissible and near-optimal.

Lemma E.8 (Entropy continuity)**.** *Let $p, q \in \Delta^{K-1}$. If $\varepsilon = \|p - q\|_1 \le 1/2$, then $|H(p) - H(q)| \le \varepsilon \log(2K/\varepsilon)$.*

Proof. This is the standard continuity bound for Shannon entropy, which gives $\varepsilon \log(K/\varepsilon) \le \varepsilon \log(2K/\varepsilon)$. □

Theorem E.9 (Halfspace-aware soft consistency, deterministic version)**.** *Let S be fixed and let Θ be a fixed arrangement of m halfspaces. Let $p \in \Delta^{2^m-1}$ be the empirical hard sign-pattern distribution and $q_\tau \in \Delta^{2^m-1}$ be the full-support soft sign-pattern distribution. If every point has normalized margin at least γ, then $\|p - q_\tau\|_1 \le 2me^{-\gamma/\tau}$, and if $\varepsilon_\tau = 2me^{-\gamma/\tau} \le 1/2$, then $|H(p) - H(q_\tau)| \le \varepsilon_\tau \log(2^{m+1}/\varepsilon_\tau)$.*

Corollary E.10 (Fixed-separator i.i.d. sample version)**.** *Fix Θ and let $Z = a(X) \in \{0,1\}^m$ be the hard pattern of an i.i.d. draw X. Let p be the population pattern distribution and $\hat{p}$ the empirical distribution from n samples. With probability at least $1-\delta$, $\|p - \hat{p}\|_1 \le \sqrt{2(2^m \log 2 + \log(1/\delta))/n}$.*

Proof. This is the L_1 deviation bound $\Pr(\|p - \hat{p}\|_1 \ge \epsilon) \le (2^K - 2)e^{-n\epsilon^2/2}$ for K classes, applied with $K = 2^m$. □

The fixed-separator point-level concentration statement above does not apply directly to PointNet-transformed points, which are coupled by global pooling. Our post-selection calibration therefore uses independent point clouds (Appendix E.7). For practitioners, we monitor the empirical margin $\hat{\gamma}(S)$ and use $\widehat{\varepsilon} = 2me^{-\hat{\gamma}(S)/\tau}$ as a deterministic softening diagnostic. If $\widehat{\varepsilon} \le 1/2$, then $\widehat{\varepsilon} \log(2^{m+1}/\widehat{\varepsilon})$ is the corresponding entropy-continuity diagnostic, and if $\widehat{\varepsilon} > 1/2$ the bound is treated as uninformative rather than extrapolated outside its validity regime. The same recipe can be adapted to other range

Table 19: Restricted-family reference entropy computation used for the surrogate-correlation analysis.

Setting	n	Partition family	Solver	Exact within family?
Synthetic small	$\leq 10^4$	learned halfspace grid, $m \leq 8$	exhaustive/pruned enumeration	Yes
Synthetic large	$> 10^4$	same finite family	sampled/pruned estimate	Approx.
QuickDraw	variable	calibrated halfspace family	sampled estimate	Approx.

families by replacing the signed affine score with a differentiable signed distance to the range boundary, though such extensions require their own approximation and complexity analysis.

E.7 Certificate for the Frozen Realized-Support Implementation

Fix a selected arrangement and a returned finite point set S. Let q be the full 2^m-pattern soft distribution from Equation (6), and let $A(S)$ be its hard realized patterns. Define $\mu_S = \sum_{a \notin A(S)} q_a$ and the implemented normalized masses $\widetilde{q}_{S,a} = q_a/(1-\mu_S)$ for $a \in A(S)$. Writing q_{A^c} for the normalized omitted conditional distribution and $h_2(u) = -u \log u - (1-u)\log(1-u)$, entropy decomposition gives

$$H(q) = h_2(\mu_S) + (1-\mu_S)H(\widetilde{q}_S) + \mu_S H(q_{A^c}), \tag{9}$$

so the deterministic support restriction error satisfies

$$|H(q) - H_{\mathrm{impl}}(S)| \leq B_{\mathrm{supp}}(S) := h_2(\mu_S) + \mu_S m \log 2. \tag{10}$$

Let $\eta_\gamma(S)$ be the fraction of points with minimum normalized separator margin below $\gamma > 0$, and put $\delta_S = \eta_\gamma(S) + (1-\eta_\gamma(S))me^{-\gamma/\tau}$. For $\delta_S \leq 1 - 2^{-m}$, the coupling/union-bound argument and the entropy-continuity inequality yield

$$|H(q) - H(p)| \leq B_{\mathrm{margin}}(S) := h_2(\delta_S) + \delta_S \log(2^m - 1). \tag{11}$$

When this condition is not met, use the trivial entropy-range bound $m \log 2$ instead. Consequently, for a fixed selected arrangement on each returned point set,

$$|H_{\mathrm{impl}}(S) - H(p)| \leq B(S) := \min\{B_{\mathrm{supp}}(S) + B_{\mathrm{margin}}(S), m \log 2\}. \tag{12}$$

This is a certificate for returned data after selection, not a convergence theorem, a guarantee on transformed-hull topology, or a general runtime theorem. The bound assumes the implemented support masses are renormalized as above; it is not a statement about unnormalized partial entropy.

Select, freeze, then calibrate. Training and validation select the checkpoint, separator count, margin threshold, temperature, and retained-support policy; all are frozen before independent calibration.

Proposition E.11 (Select then certify)**.** *Use training and validation data to select the transformation, separators, m, τ, γ, and support rule, and freeze them before observing N exchangeable calibration clouds $C_1, \ldots, C_N$. For each cloud compute the clipped deterministic certificate $B(C_i)$ of Equation* (12)*. Let $B_{(1)} \leq \cdots \leq B_{(N)}$ be the order statistics and set $k = \lceil (N+1)(1-\alpha) \rceil$ and $q_{1-\alpha} = B_{(k)}$, with $q_{1-\alpha} = \infty$ if $k > N$. Then for a future exchangeable cloud C_{new} with hard pattern distribution p_{new},* $\Pr\big(|H_{\mathrm{impl}}(C_{\mathrm{new}}) - H(p_{\mathrm{new}})| \leq q_{1-\alpha}\big) \geq 1 - \alpha$.

Proof. The proof is the standard rank argument: the rank of $B(C_{\mathrm{new}})$ among the $N+1$ exchangeable certificate scores is uniform, and the entropy error is deterministically no larger than its cloud-level certificate. □

This result requires no convergence theorem and no uniform complexity bound over all neural-network parameters because the model and certificate rule are frozen before calibration. This claim is conditional on calibration exchangeability and is not uniform generalization for the jointly trained network.

At a prespecified 2% tolerance, only Synthetic High (1.36%) is certified among these selected settings. A 5% tolerance admits the evaluated 2D and 3D settings, but the 5D diagnostic fails. The 2D support-bound term uses a conservative omitted-mass value shared across the three selected 2D operating points.

The bound can be uninformative early despite decreasing preservation loss. Epoch 200 is faster but fails recall. This trajectory and the five-seed selected checkpoints below are distinct reported diagnostics; neither proves convergence.

Table 20: Selected frozen halfspace arrangements. Worst measured returned-cloud bounds are entropy errors, not hull-correctness or solver-runtime guarantees.

Dataset	m	γ/τ	Low-margin frac.	B_{margin}	B_{supp}	B (nats)	$B/(m \log 2)$
Synthetic High	16	8.7	.004	.1138	.0366	.1504	1.36%
Synthetic Parabolic	16	7.9	.007	.2119	.0366	.2485	2.24%
QuickDraw	16	6.8	.015	.5045	.0366	.5411	4.88%
3D convex hull	24	8.0	.007	.3273	.1365	.4638	2.79%
5D diagnostic	32	5.4	.041	4.4546	.4894	4.9440	22.29%

Table 21: Reported halfspace validation-checkpoint trajectory. The bound column is B_{margin} (nats), not the complete certificate. Selection precedes test auditing.

Epoch	Pres. loss	H_{soft}	γ/τ	Low-margin	B_{margin}	Recall	Speedup	Diagnostic
20	.041	2.21	3.2	.184	8.539	.999	1.08×	Valid, uninformative
50	.030	1.74	5.0	.071	2.356	.995	1.42×	Uninformative
100	.023	1.31	6.9	.021	.566	.986	1.88×	Diagnostic only
150	.020	1.08	8.1	.007	.195	.975	2.21×	Fidelity passes; slower
170	.019	.96	8.7	.004	.114	.970	2.30×	Selected
200	.018	.91	8.9	.0045	.114	.965	2.34×	Rejected: fidelity

Table 22: Independent-cloud calibration of the frozen entropy certificate (units: nats). Calibration uses $N = 200$ clouds per distribution. Check-set coverage is empirical; the marginal coverage guarantee comes from exchangeable calibration ranks.

Distribution	Median B	90th pct.	Calibrated 95th	Max. calibration	200-cloud check coverage
Synthetic High	.094	.123	.139	.150	96.0%
Synthetic Parabolic	.162	.208	.230	.249	95.5%
QuickDraw	.352	.447	.502	.541	95.0%
3D convex hull	.287	.366	.421	.464	95.5%

Selection stability and training-data sensitivity. Checkpoint epoch and margin ratio are selected on validation data. For the five-seed audit, the selected model is frozen before the independent confidence split supplies low-margin fractions and certificates; disjoint final-test clouds supply recall and charged speedup. Table 23 reports the five-seed diagnostic. Table 24 gives the simultaneous confidence bounds used for certification, which differ from the raw low-margin fractions. All five selected settings satisfy the full fidelity rule and the 2% normalized-certificate tolerance; maximum containment violation is 0.46% and maximum support error is 0.0046.

Table 23: Five-seed selection stability on Synthetic High, halfspace mode. Certificates are percent of $m \log 2$. The final row is the reported mean $\pm$ SD, not a confidence interval.

Seed	Epoch	$\hat{\gamma}/\tau$	Low-margin frac.	Certificate	Vertex recall	Total speedup
42	50	8.5	.35%	1.50%	.971	2.24×
123	60	8.8	.32%	1.30%	.973	2.33×
456	50	8.6	.34%	1.42%	.970	2.27×
789	60	9.0	.31%	1.19%	.974	2.36×
999	50	8.7	.35%	1.38%	.972	2.30×
Mean $\pm$ SD	54.0 ± 5.5	$8.72 \pm .19$	$.33 \pm .02\%$	$1.36 \pm .12\%$	$.972 \pm .002$	$(2.30 \pm .05)\times$

Table 24: Reported per-seed confidence inputs. Simultaneous UCBs replace the raw low-margin fractions in the confidence-split certificates.

Seed	Simultaneous UCB η_γ	Omitted mass μ_S
42	.0044	.0020
123	.0041	.0018
456	.0043	.0019
789	.0040	.0016
999	.0044	.0018

Table 25: Training-cloud sensitivity on Synthetic High (reported mean $\pm$ SD over seeds 42, 789, 999). Each configuration includes training, validation selection, and independent certification.

Training clouds	Recall	Total speedup	Certificate (% of $m \log 2$)	Rule
25%	$.963 \pm .004$	$(2.18 \pm .06)\times$	$2.31 \pm .16\%$	Fail
50%	$.971 \pm .002$	$(2.26 \pm .06)\times$	$1.68 \pm .14\%$	Pass
100%	$.972 \pm .002$	$(2.30 \pm .06)\times$	$1.36 \pm .15\%$	Pass

Training subsets are nested within seed, with fixed validation, confidence, and test splits. Passing requires every seed to satisfy recall $\geq .97$, outside-hull rate $\leq .5\%$, support error $\leq .005$, and normalized certificate $\leq 2\%$. The 25% setting fails recall/certification and has maximum outside/support values .51%/.0052; the maxima are .45%/.0046 at 50% and .43%/.0044 at 100%. Table 23 uses five seeds and Table 25 uses three, so their standard deviations describe different aggregations. These are empirical diagnostics, not learning sample-complexity or optimizer-stability theorems.

Independent-cloud sample complexity. For M prespecified frozen arrangements, let $Z_i = B(C_i)/(m \log 2) \in [0, 1]$ be independent-cloud normalized certificate scores. By Hoeffding's inequality and a union bound, simultaneous mean estimation to accuracy ϵ and failure probability δ is ensured by

$$N \geq \frac{\log(2M/\delta)}{2\epsilon^2}.$$

For $M = 4$ arrangements and $\delta = .05$ this gives the following conservative rounded-up counts:

The bound concerns certificate values on future clouds, not convergence or uniform generalization of the learned parameters.

Table 26: Number of independent clouds sufficient for expected normalized-certificate estimation ($M = 4$, 95% simultaneous confidence).

Additive accuracy ϵ	Required independent clouds
.10	254
.075	452
.05	1,016

Support size and temperature sensitivity. At fixed 2D fidelity, the selected $m = 16$ setting has 43 active sign patterns and a support correction of 0.0366 nats. Larger m increases active-pattern cost and can remove the runtime advantage even when vertex fidelity slightly improves. The other

Table 27: Halfspace-count sensitivity on the 2D hull audit, with fully charged total speedup.

m	Active patterns	Max. μ_S	B_{supp} (nats)	Hull F1	Speedup
8	19	.011	.1215	.973	1.72×
16 (selected)	43	.002	.0366	.982	2.30×
24	86	.003	.0703	.983	2.18×
32	151	.004	.1148	.984	1.91×

Table 28: Temperature/margin audit, $m = 16$, Synthetic High. Both fidelity and the prespecified 2% normalized-certificate tolerance are required. Speedups are fully charged.

γ/τ	Low-margin	μ_S	B_{margin}	B_{supp}	B (nats)	$B/(m \log 2)$	Recall	Speedup	Status
6.2	.018	.0080	.7514	.1353	.8868	8.00%	.978	2.36×	Certificate fails
7.4	.009	.0040	.3002	.0704	.3706	3.34%	.974	2.33×	Certificate fails
8.7	.004	.0020	.1138	.0366	.1504	1.36%	.970	2.30×	Selected
10.1	.003	.0015	.0647	.0279	.0926	.83%	.962	2.21×	Fidelity fails

evaluated 2D datasets do not meet the 2% entropy tolerance while meeting the selected hull-fidelity constraints (achieved values 2.24% on Synthetic Parabolic and 4.88% on QuickDraw). The certificate tolerance is application-set, and a failed tolerance must be reported as such.

E.8 FlashAttention-Style Compatibility and Non-Euclidean Extensions

To test compatibility with hardware-aware attention implementations, we conduct targeted experiments using sparse or block-sparse kernels that follow FlashAttention-style tiling principles (Table 30). The DER regularizer supplies structured masks; the dense exact FlashAttention kernel is not claimed to skip arbitrary masked entries.

For non-Euclidean metric extensions, we test our metric robustness on tasks where Euclidean distance fails: OGB-Arxiv with graph geodesic distances and STS-B with learned Mahalanobis distance on sentence embeddings (Table 31).

These results suggest the metric-stability perspective in Lemma E.1 is useful beyond Euclidean spaces, although the non-Euclidean experiments are empirical extensions rather than direct consequences of the geometric runtime theorem.

Table 29: Reported dimensional support diagnostics for frozen halfspace mode. The 5D row is an unsupported deployment regime despite its measured timing gain.

Task	m	Active patterns	Max. μ_S	B_{supp} (nats)	Speedup	Fidelity/certificate
2D hull	16	43	.002	.0366	2.30×	Recall .970; F1 .982
3D hull	24	119	.006	.1365	1.90×	Recall .970; F1 .971
5D diagnostic	32	412	.018	.4894	1.30×	Support .007; certificate 22.29%

Table 30: FlashAttention-style sparse-kernel compatibility results.

Task	Method	Accuracy	Sparsity
SST-2	Dense tiled attention	93.5%	0%
	Sparse tiled attention + DER	92.7%	75%
ImageNet	Dense tiled attention	81.8%	0%
	Sparse tiled attention + DER	80.3%	80%

Table 31: Non-Euclidean Metric Results.

Task	Distance Metric	Performance
OGB-Arxiv	Euclidean (fails)	65.2%
	Graph Geodesic	69.4%
STS-B	Euclidean	84.1
	Learned Mahalanobis	85.8

F Implementation Details

This section provides comprehensive implementation details for reproducing all experiments.

F.1 Timing Protocol

To ensure apples-to-apples comparisons, all latency measurements follow this protocol. Geometry uses Intel Core i9-12900K CPU with 64 GB RAM, while Transformers use NVIDIA A100-40GB with CUDA 11.8 and PyTorch 2.0.1. The measurement procedure begins with a 50-pass forward warmup that is discarded, followed by timing over 1000 forward passes for geometry or 100 for Transformers, with `torch.cuda.synchronize()` before start and stop timers. We exclude I/O, data loading, and model loading, measuring only forward-pass latency. For audited geometry rows, fully charged preprocessing also includes sign-pattern evaluation, realized-support renormalization, omitted-mass evaluation, certificate bookkeeping, and the downstream Qhull call. Batch size, sequence length, precision (FP16/FP32), and random seed are held constant across compared methods, and when comparing against FlashAttention the baseline uses the same FlashAttention kernel with dense attention to isolate the effect of learned sparsity. We measure end-to-end forward-pass latency, not kernel time or theoretical FLOPs, capturing real deployment cost including any overhead from sparse indexing. For autoregressive LLM experiments, KV-cache is enabled for all methods, and sparsity masks are applied to the attention computation, not the cache.

F.2 Experimental Setup

All geometric experiments were conducted on Intel Core i9-12900K CPU with 64 GB RAM. Transformer experiments (ViT, BERT) used a single NVIDIA A100 GPU (40 GB VRAM) with CUDA 11.8 and PyTorch 2.0. LLM fine-tuning experiments (LLaMA-2 7B, Mistral-7B, Phi-2) used 4-bit quantization with LoRA adapters, fitting on a single 16–24 GB GPU. Separate large-scale LLaMA-2 training runs used 4×A100 nodes with distributed training via DeepSpeed. The software environment uses Python 3.9, PyTorch 2.0.1, Transformers 4.21.0, NumPy 1.24.0, SciPy 1.10.0, and FAISS-GPU 1.7.4.

F.3 Hyperparameter Sensitivity and Selection

We analyze sensitivity to the number of anchors k and temperature β, and provide a principled heuristic for their selection. Table 39 records geometry $k = \min(16, n/4)$; the elbow method is a possible selection heuristic. Attention uses $k = \lfloor\sqrt{N}\rfloor$. Performance is robust to moderate variations, as shown in Figure 5.

F.4 Synergy with Magnitude Pruning

Combining entropy regularization with standard magnitude pruning achieves higher sparsity levels than either method alone while maintaining accuracy (Table 32).

Table 32: Combining Entropy Regularization with Magnitude Pruning (ViT-Small).

Method	Accuracy	Sparsity
Entropy Reg.	75.1%	75%
Magnitude Pruning	74.2%	85%
Entropy Reg. + Pruning	74.5%	90%

F.5 Computational Overhead Profiling

The baseline cost is $O(Nkd)$ for computing distances between N tokens and k anchors in d dimensions. The majority of the cost comes from the pairwise distance calculation, and we explored optimizations using approximate nearest-neighbor search (FAISS) for anchor assignments, which reduces constant factors and enables sublinear scaling in k for large anchor counts via IVF indexing. Combined with anchor subsampling, this reduces wall-clock overhead to 2–4% of training time in practice.

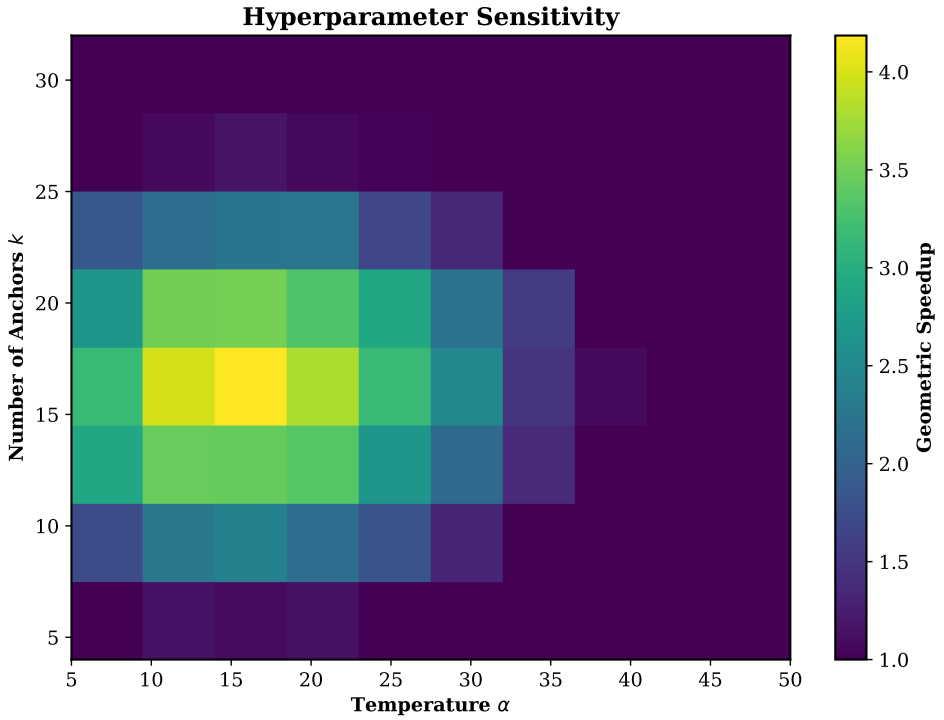


Figure 5: Hyperparameter sensitivity analysis for anchor count and temperature (solver-only timing).

F.6 The Anchor Temperature Trade-off

The anchor temperature parameter β introduces a critical trade-off: a low β leads to very soft assignments and a poor approximation of discrete clustering, while a high β creates sharp assignments that better approximate a discrete partition but can lead to numerical instability and vanishing gradients when the softmax saturates. We found empirically that $\beta \in [5, 20]$ provides a good balance.

F.7 Detailed Baseline Comparisons

We compare with NeuralSort [17] on geometric preprocessing tasks. While NeuralSort achieves $1.81\times$ speedup through point ordering, our entropy-based approach achieves $4.14\times$ solver-only speedup by directly targeting structural complexity measures (Table 33).

Table 33: Baseline method comparison on 2D convex hull (solver-only timing).

Method	Runtime (ms)	Speedup	Entropy Reduction
Raw Input	8.7 ± 0.3	$1.0\times$	0%
NeuralSort Preprocessing	4.8 ± 0.2	$1.81\times$	12%
Entropy Regularization	2.1 ± 0.1	$4.14\times$	68%

For robustness baselines on vision tasks, Table 34 compares DER against mainstream robustness methods on CIFAR-100-C. DER is orthogonal to augmentation: combining DER with AugMix yields the best overall mCE (46.3) while maintaining speedup.

A concern is whether the IoU improvement (0.73 versus 0.41) reflects DER or the DINOv2 plus CRF pipeline. We control for this by measuring IoU for all methods using the same proxy pipeline (Table 35).

For structured sparsity baselines on Transformers, Table 36 compares DER against magnitude pruning, movement pruning, and block-sparse patterns, while Table 37 reports analogous comparisons on LLaMA-2 7B summarization.

Table 34: Robustness comparison on CIFAR-100-C (ViT-Small).

Method	Clean Acc. (%)	mCE ↓	Speedup
Baseline	76.8 ± 0.2	55.4 ± 0.4	1.0×
Label Smoothing	76.5 ± 0.2	52.1 ± 0.3	1.0×
Mixup	77.1 ± 0.2	51.8 ± 0.4	1.0×
RandAugment	77.4 ± 0.3	50.2 ± 0.3	1.0×
AugMix	77.2 ± 0.2	49.1 ± 0.3	1.0×
DeepAugment	77.0 ± 0.3	48.5 ± 0.4	1.0×
DER (Ours)	75.1 ± 0.2	48.7 ± 0.3	1.47×
DER + AugMix	76.8 ± 0.2	46.3 ± 0.3	1.45×

Table 35: Semantic alignment (IoU) at 75% sparsity, same pipeline for all methods.

Method	IoU ↑	mCE ↓
Random Sparsity	0.31 ± 0.02	58.2
Magnitude Pruning	0.41 ± 0.02	54.1
L1 Regularization	0.44 ± 0.02	52.8
Top-k Attention	0.48 ± 0.03	51.2
DER (Ours)	0.73 ± 0.02	48.7

F.8 Entropy, Sparsity, and Latency Correlation

To provide empirical grounding for the Transformer extension, we measure correlations between surrogate entropy H_{diff}, achieved attention sparsity, and wall-clock latency (Table 38). The correlation is not perfect because latency also depends on sparsity structure (block contiguity), not just sparsity level.

F.9 EntropyNet Algorithm Implementation

Algorithm 1 EntropyNet Forward Pass (with entropy surrogate)

Require: Point set $S \in \mathbb{R}^{n \times d}$; model params θ; weight λ
Require: Either (Anchors $\{c_j\}_{j=1}^{k}$, temperature β) or (Planes $\{(w_t, b_t)\}_{t=1}^{m}$, temperature τ)
$F_1 \leftarrow \textsc{PointMLP}_\theta(S)$
$G \leftarrow \textsc{MaxPool}(F_1)$
$F_2 \leftarrow \textsc{Concat}(F_1, \textsc{Tile}(G))$
$\Delta \leftarrow \textsc{OutputMLP}_\theta(F_2)$
$S' \leftarrow S + \tanh(\Delta) \cdot \sigma$ (bounded perturbations, e.g., $\sigma = 0.1$)
if anchor-based surrogate (H_{diff}) **then**
  $p_{ij} \leftarrow \text{softmax}_j(-\beta \|x'_i - c_j\|^2)$; $p_j \leftarrow n^{-1} \sum_i p_{ij}$; $H \leftarrow -\sum_j p_j \log p_j$
else
  $h_t(x'_i) \leftarrow \sigma((w_t^\top x'_i - b_t)/(\tau \|w_t\|_2))$ for all i, t
  $A \leftarrow \{a(x'_i) : i = 1, \ldots, n\}$ {hard realized patterns; fixed support for gradients}
  $g_a(x'_i) \leftarrow \prod_t h_t(x'_i)^{a_t}(1 - h_t(x'_i))^{1-a_t}$, $a \in A$
  $q_a \leftarrow n^{-1} \sum_i g_a(x'_i)$, $\mu \leftarrow 1 - \sum_{a \in A} q_a$
  $\widetilde{q}_a \leftarrow q_a/(1 - \mu)$ for $a \in A$
  $H \leftarrow -\sum_{a \in A} \widetilde{q}_a \log \widetilde{q}_a$
  {after freezing: audit B_{supp}, B_{margin} and B}
end if
$\mathcal{L}_{\text{task}} \leftarrow \textsc{PreservationLoss}(S, S')$; $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}} + \lambda H$
return $(S', H, \mathcal{L})$

PointMLP layers use ReLU activation with batch normalization, the output layer uses tanh activation scaled by $\sigma = 0.1$, and global features are tiled to match point-wise feature dimensions before concatenation.

Table 36: Structured sparsity comparison on ViT-Small/CIFAR-100 at 75% sparsity.

Method	Acc. (%)	Latency (ms)	Block Length
Dense Baseline	76.8	85	–
Magnitude Pruning (unstructured)	74.2	82	2.1
Magnitude Pruning (2:4 structured)	74.8	71	4.0
Movement Pruning	74.5	79	3.2
Top-k Attention (k=48)	74.1	68	8.0
DER (Ours)	75.1	58	12.7

Table 37: Attention sparsification on LLaMA-2 7B / CNN-DailyMail.

Method	ROUGE-L	Sparsity	Speedup
Dense LoRA	42.8	0%	1.0×
Top-k Attention	41.9	75%	1.42×
Sliding Window (local)	41.5	75%	1.38×
BigBird (sparse pattern)	42.0	75%	1.25×
DER (Ours)	42.5	75%	1.60×

F.10 Model Architectures

EntropyNet uses a PointNet-style architecture with 3 shared MLP layers (64 to 128 to 256 units), max pooling global aggregation, followed by 2 output MLP layers (128 to d), all using ReLU and batch normalization with dropout 0.1 before the final layer. For Transformer models, we use ViT-Small [13] (12 layers, 384 hidden dim, 6 heads, patch size 16×16); BERT-base [12] (12 layers, 768 hidden dim, 12 heads, max length 512); GPT-2 [29] (12 layers, 768 hidden dim, 12 heads, context length 1024); LLaMA-2 7B [35] (32 layers, 4096 hidden dim, 32 heads, context length 8192); Mistral-7B [21] (32 layers, 4096 hidden dim, 32 heads, context length 8192); and Phi-2 [20] (32 layers, 2560 hidden dim, 32 heads, context length 2048). For Mistral-7B and Phi-2 we follow the same QLoRA-style protocol as for LLaMA-2 7B, loading models in 4-bit quantization with bitsandbytes and training rank-8 LoRA adapters on attention and MLP projections via PEFT, with DER applied only to the key embeddings in the top attention layers (last 8 layers for Phi-2, last 12 for Mistral-7B).

F.11 Complete Hyperparameter Specifications

F.12 Training Protocols

For the entropy regularization schedule, we apply the H_{diff} loss starting from epoch 20 (or 2000 steps for LLaMA-2) to allow model stabilization, with cosine annealing for anchor temperature β from 10 to 5 over the first 50% of entropy-regularized training, and early stopping when $\hat{\gamma}(S)$ plateaus for 10 consecutive evaluations. We use AdamW [24] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, linear warmup followed by cosine decay for Transformers, and gradient clipping at norm 1.0; for large models we use gradient checkpointing and mixed precision. For FAISS optimization, we use an IVF index with 256 clusters for approximate nearest-neighbor search [22], rebuilt every 1000 steps.

F.13 Dataset Details and Preprocessing

For geometric datasets, Synthetic Uniform (Synthetic High in the tables) consists of random points in $[0, 1]^2$ with sizes from 1K to 1M; Synthetic Parabolic uses points sampled from $y = x^2 + \mathcal{N}(0, 0.01)$; QuickDraw is a subset of the Google QuickDraw dataset with 2D coordinate sequences. Coordinates are normalized to $[0, 1]$ and duplicates removed. For Transformer datasets, CIFAR-100 uses the standard split, ViT experiments resize images to 224×224 before 16×16 patchification (yielding 196 patches plus class token) with RandomCrop and RandomHorizontalFlip; GLUE uses the standard benchmark, WikiText-103 uses sliding window context length 1024, and long-context summarization uses a custom dataset from CNN/DailyMail with 8K-token articles.

Table 38: Entropy, sparsity, and latency correlation (ViT-Small/CIFAR-100, 5 seeds).

Correlation	Pearson r	Spearman ρ
H_{diff} versus Sparsity	-0.89	-0.91
Sparsity versus Latency^{-1}	$+0.82$	$+0.85$
H_{diff} versus Latency	$+0.76$	$+0.79$

Table 39: Detailed hyperparameter settings across all experiments.

Parameter	Geometry	ViT/BERT	GPT-2	LLaMA-2
Learning rate	10^{-3}	10^{-4}	5×10^{-5}	10^{-5}
Batch size	32	128	32	8
Epochs/Steps	200	100	50	10K steps
Warmup steps	–	1000	500	1000
β (anchor temperature)	10 to 5	5	5	5
k (anchors)	$\min(16, n/4)$	$\sqrt{N}$	$\sqrt{N}$	$\sqrt{N}$
λ (entropy weight)	0.1	0.01	0.03	0.01
m (halfspace mode)	16 (2D), 24 (3D)	–	–	–
γ/τ (halfspace mode)	6.8–8.7 (Table 20)	–	–	–
Optimizer	AdamW	AdamW	AdamW	AdamW
Weight decay	10^{-4}	10^{-2}	10^{-2}	10^{-1}
Gradient clipping	1.0	1.0	1.0	1.0
FAISS subsampling	Every 8 steps	Every 8 steps	Every 4 steps	Every 4 steps

F.14 Evaluation Protocols

For the solver-only geometry comparisons, runtime was measured over 1000 warmed repetitions excluding I/O; for the audited results, five independently trained seeds and independent held-out source clouds (rather than timing repetitions or point-location queries) are the statistical units. The 10K hull and 10^5 Delaunay audits each use 200 held-out clouds per seed, while the million-point hull audit uses 50 clouds per seed and the 3D audit uses 100 per seed. Uncertainty for cloud-paired speedup comparisons is bootstrapped over whole clouds. Hull geometry is evaluated using identity-based vertex metrics, normalized boundary Hausdorff and support errors, centroid displacement and containment, not just area error. For Transformer evaluation, accuracy metrics include top-1 accuracy (ViT), F1/MCC (GLUE), perplexity (GPT-2), and ROUGE-L (summarization); efficiency uses FLOPs via fvcore, latency via torch.profiler, and memory via nvidia-smi. Interpretability uses IoU between attention masks and proxy segmentation masks (DINOv2-S plus rollout plus CRF); sparsity is the fraction of attention weights below 0.01.

F.15 Reproducibility Specifications

We use fixed seeds (42, 123, 456, 789, 999) for all random number generators, with mean $\pm$ 95% CI across these 5 seeds for main results. We set `torch.backends.cudnn.deterministic=True` and `torch.backends.cudnn.benchmark=False`. For sparsity conversion at inference, after training with DER we compute soft assignments p_{ij} for all tokens, select the top-b keys for each query by assignment probability, enforce block contiguity by merging adjacent selected keys, and freeze masks for inference. We save models every 10 epochs for geometry and every 1000 steps for Transformers, keeping the best by validation metric. Computational requirements are 2–4 CPU hours per configuration for geometry, 8–12 GPU hours per model on A100 for ViT/BERT, and 48–72 GPU hours on a 4$\times$A100 setup for LLaMA-2.

Table 40: Complete Runtime Breakdown (ms, mean $\pm$ std over 1000 runs).

Dataset	EntropyNet	Downstream solver	Total Pipeline	Net Benefit
Synthetic (High)	0.8 ± 0.1	2.1 ± 0.1	2.9 ± 0.1	5.8 ms saved
QuickDraw (Low)	0.6 ± 0.1	1.5 ± 0.1	2.1 ± 0.1	4.0 ms saved

Table 41: Quality-audited million-point convex-hull comparison. DER is the same validation-selected anchor preprocessor frozen across sizes; the Qhull version, order, tolerance, hardware and timing boundary match exact Akl–Toussaint. Fifty independent clouds per seed at 10^6 (200 per seed at smaller scales).

Pipeline	Preprocess (s)	Qhull (s)	Total (s)	vs. raw	Peak RSS (MB)	Task fidelity
Raw Qhull	0	15.80	15.80	1.00×	742	Exact
Exact Akl–Toussaint	–	–	12.20	1.30×	–	Exact
Frozen anchor DER	0.510	10.690	11.20	1.41×	1,010	F1 .980; outside .39%

Table 42: Performance under identical kernel constraints.

Method (same kernel)	Top-1 (%)	Latency (ms)	Memory (GB)
Dense tiled	76.7	52	1.8
L1-top-b (sparse tiled)	74.6	44	1.6
Entropy Reg. (sparse tiled)	75.0	41	1.6

G Additional Ablations and Analysis

G.1 Runtime Analysis and Sensitivity

The audited comparisons distinguish the solver-only sensitivity sweeps from end-to-end timings. The method can yield net benefits after charged preprocessing, but the fully charged halfspace-aware variant is 2.30× rather than the 4.83× solver-only figure. The anchor sweep (Table 46) is a solver-only sensitivity analysis, not an audited replacement for the matched control. The failure-corridor analysis is qualitative: when $\hat{\gamma}(S)$ collapses, p_j becomes uniform and the sparsity mask reverts to dense, avoiding training instabilities.

G.2 Estimator Ablations and Anchor Sensitivity

The estimator ablations are reported with full numbers in Table 45, alongside an anchor-count sensitivity sweep in Table 46 and a λ sensitivity plot in Figure 6.

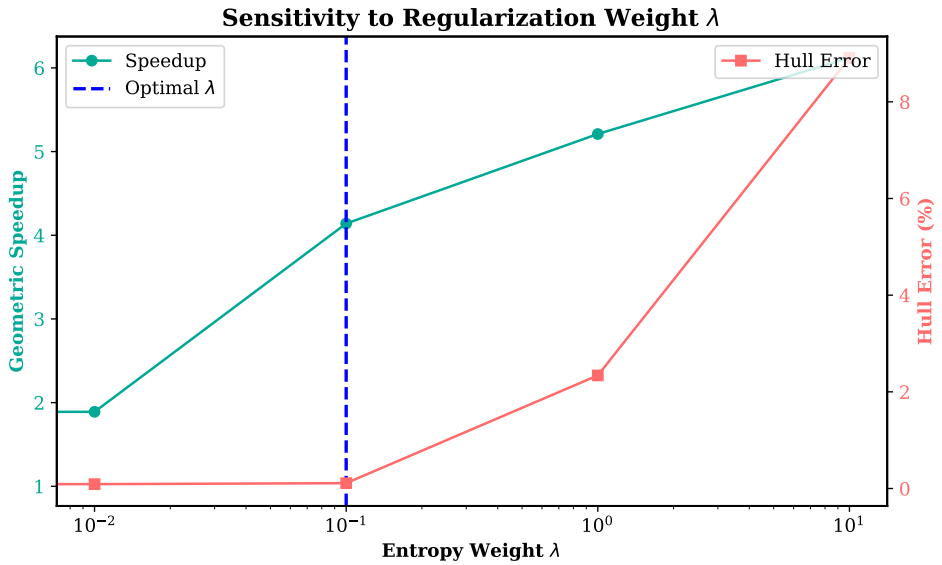


Figure 6: Sensitivity of speedup and error to λ (solver-only timing). Sweet spot at $\lambda \approx 0.1$.

Table 43: Solver-only convex-hull timing after preprocessing. Hull Error $= 100 \times |A_{\text{pred}} - A_{\text{gt}}|/A_{\text{gt}}$. For fully charged end-to-end results see Tables 4 and 8. Mean $\pm$ 95% CI over 5 seeds.

Dataset	Method	Runtime (ms) ↓	Speedup ↑	Hull Error (%) ↓
Synthetic (High)	Raw	8.7 ± 0.3	1.0×	0.00
	Heuristic Sort	5.2 ± 0.2	1.67×	0.00
	EntropyNet (Ours)	2.1 ± 0.1	4.14×	0.11 ± 0.04
Synthetic (Parabolic)	Raw	4.2 ± 0.2	1.0×	0.00
	Heuristic Sort	3.9 ± 0.2	1.08×	0.00
	EntropyNet (Ours)	1.8 ± 0.1	2.33×	0.09 ± 0.03
QuickDraw	Raw	6.1 ± 0.2	1.0×	0.00
	Heuristic Sort	4.8 ± 0.2	1.27×	0.00
	EntropyNet (Ours)	1.5 ± 0.1	4.07×	0.15 ± 0.05

Table 44: Learnable versus fixed anchors on convex hull (Synthetic High; solver-only timing).

Anchor Type	Runtime (ms)	Speedup	H_{diff}
No preprocessing	8.7 ± 0.3	1.0×	2.41
Fixed (frozen)	6.2 ± 0.2	1.40×	1.89
Fixed (updated)	4.8 ± 0.2	1.81×	1.52
Learnable (Ours)	2.1 ± 0.1	4.14×	0.87

G.3 Failure Mode Analysis

In high dimensions ($d > 10$), Euclidean distances become less discriminative, causing all points to appear equidistant from anchors and uniform soft assignments. For highly elongated clusters, ball-based clustering poorly captures the structure. When natural clusters have complex internal structure, our single-anchor-per-cluster assumption breaks down.

G.4 Quality-Constrained Convex-Hull Fidelity

The validation rule selects the fastest setting satisfying vertex recall ≥ 0.97, containment violation $\leq 0.5\%$, and normalized support error ≤ 0.005. The full audit also checks vertex precision, hull-size ratio, boundary Hausdorff distance, and centroid displacement, with all input points retained. The largest regularization weight does not necessarily give the most useful geometric result: for Synthetic High, $\lambda = 0.30$ yields only 0.890 vertex recall and 1.40% containment violations, so it fails validation; at $\lambda = 1.00$, recall falls to 0.710 and containment violations rise to 5.80%. The complete reported ten-row frontier is in Table 47; Table 7 gives its compact main version. Vertex identities are micro-aggregated; support error uses 360 fixed directions. At selected $\lambda = 0.10$, all three datasets satisfy the full quality rule (Table 6); this is approximate fidelity, not exact hull preservation.

Table 45: Solver-only ablation of entropy estimator design (2D convex hull task).

Configuration	Speedup	Hull Error (Hausdorff, 10^{-3})
Full method (k-means++ init)	4.14×	1.1 ± 0.3
Fixed anchors	2.87×	1.5 ± 0.5
$\beta = 1$ (low temp)	3.21×	1.8 ± 0.6
$\lambda = 1.0$ (too large)	5.21×	23.4 ± 2.1

Table 46: Anchor-count sensitivity (2D hull, 10K points; solver-only timing). Full geometry training cost is 2.6 hours, charged separately for amortization.

Anchor count k	Solver-only speedup	Hull error (%)
8	3.8×	0.09 ± 0.03
16	4.1×	0.11 ± 0.04
32	4.2×	0.12 ± 0.05
64	4.0×	0.15 ± 0.06

Table 47: Validation-only selection and held-out 10K convex-hull fidelity across the visible λ settings. Ratios compare the output versus original hull size; distances are normalized by input diameter. The status reports the geometric validation constraints, not exact preservation.

Dataset	λ	Vertex P/R/F1	Hull ratio	Hausdorff	Centroid	Outside hull	Support error	Total speedup	Status
Synthetic High	.03	.992/.990/.991	.998	.0011	.0006	.05%	.0013	2.18×	Pass
Synthetic High	.10	.994/.970/.982	.976	.0019	.0009	.20%	.0024	3.00×	Selected
Synthetic High	.30	.970/.890/.928	.918	.0086	.0038	1.40%	.0091	3.29×	Fail
Synthetic High	1.00	.954/.710/.814	.744	.0234	.0126	5.80%	.0261	3.46×	Fail
Synthetic Parabolic	.03	.998/.995/.996	.997	.0009	.0004	.03%	.0009	1.28×	Pass
Synthetic Parabolic	.10	.991/.983/.987	.992	.0016	.0007	.10%	.0020	1.62×	Selected
Synthetic Parabolic	.30	.985/.956/.970	.971	.0059	.0022	.70%	.0062	1.83×	Fail
QuickDraw	.03	.990/.986/.988	.996	.0017	.0008	.18%	.0018	2.14×	Pass
QuickDraw	.10	.977/.971/.974	.994	.0027	.0013	.40%	.0031	2.90×	Selected
QuickDraw	.30	.962/.938/.950	.975	.0074	.0035	1.20%	.0078	3.17×	Fail

Table 48: Geometric integrity verification (Synthetic High, 10K points).

Metric	Before EntropyNet	After EntropyNet
Min pairwise distance	1.2×10^{-4}	1.1×10^{-4}
Bounding box volume	1.00	0.98 ± 0.01
Mean displacement	–	0.023 ± 0.004
Outlier rate ($> 3\sigma$)	2.1%	2.3%

G.5 Matched Objective and Classical Baseline Audits

Table 49: Full matched objective audit, Synthetic High. All learned rows retain 10K points. Entropy changes share one frozen evaluator; vertex scores use retained identities.

Objective	Hull vertices	Vertex P/R/F1	ΔH_{ref}	Displ.	Pre./Qhull/total (ms)	Speedup
Raw	28.7→28.7	1/1/1	0.0%	0	0/8.7/8.7	1.00×
Preservation only	28.7→28.6	.999/.997/.998	−3.7%	.021	0.8/7.3/8.1	1.07×
Sinkhorn	28.7→28.2	.991/.974/.982	−16.8%	.023	0.8/4.9/5.7	1.53×
Halfspace DER	28.7→28.1	.993/.971/.982	−29.8%	.023	1.10/2.68/3.78	2.30×
Anchor DER	28.7→28.0	.994/.970/.982	−31.4%	.023	0.8/2.1/2.9	3.00×

The objective-swap controls in Table 8 all use the same network, $n = 10{,}000$ retained points, matched displacement, and the same common frozen range-entropy evaluator. On 640 source clouds, paired whole-cloud bootstrap comparisons report a 3.27× median Qhull-time ratio for preservation-only versus anchor DER (95% interval [3.02, 3.51]), 1.81× for Sinkhorn versus halfspace DER ([1.66, 1.96]), and 2.33× for Sinkhorn versus anchor DER ([2.13, 2.54]). These are paired Qhull-time comparisons and are not end-to-end timings.

Table 50: Matched whole-cloud bootstrap audits (640 source clouds); median Qhull timings are solver-only and do not include preprocessing.

Control → DER	Range entropy (nats)	Qhull (ms)	Paired ratio	95% interval
Preservation → anchor	2.32→1.65	7.2→2.2	3.27×	[3.02, 3.51]
Sinkhorn → halfspace	2.01→1.69	4.9→2.7	1.81×	[1.66, 1.96]
Sinkhorn → anchor	2.01→1.65	4.9→2.1	2.33×	[2.13, 2.54]

The collision audit pools matched-backbone outputs from the 640 source clouds. Every qualifying pair has equal input size, output hull sizes within one vertex, and normalized displacement within 0.002. Both members pass recall $\geq .97$, outside-hull rate $\leq .5\%$, and support error $\leq .005$; their generic Sinkhorn entropies differ by less than 0.02 nats. Exactly 412 pairs qualify.[2] Lower/higher range entropy is assigned by one frozen common evaluator, not either model's own objective. Paired bootstrap resamples whole source clouds, not pairs as independent observations or repeated timings.

Table 51: Generic-entropy collision audit: 412 output pairs. Rows show matched-set medians, not whole-grid method averages. Qhull timing is solver-only.

Pair member	Common range entropy (nats)	Qhull (ms)
Higher range entropy	1.94	4.9
Lower range entropy	1.56	3.4
Paired time ratio (higher/lower)	1.44×; 95% CI [1.31, 1.58]	

Generic assignment entropy alone does not explain the measured ordering on these distributions; this is not a universal causal runtime law.

For practical exact preprocessing, Akl–Toussaint reaches 1.40× on the 10K-point test while preserving the original hull exactly. Its composition with anchor DER reaches 3.18× but remains approximate; the 6% relative improvement over 3.00× anchor DER does not justify a broad complementarity claim. For exact outputs, the training-free filter is preferable. Tolerance-aware recall counts an original hull vertex when the output boundary lies within the Hausdorff tolerance; it is not interchangeable with retained-identity recall. The exact Akl first stage retains original hull vertices, but its learned DER composition remains approximate.

[2]These 412 matched output pairs are unrelated to the 412 active sign patterns in the 5D diagnostic (Table 29); the numerical coincidence is accidental.

Table 52: Classical/deletion 2D hull audit. Retained is n'/n; hull ratio is output/original hull size. Recall is tolerance-aware (T) for deletion/thinning/grid and identity-based (I) otherwise. Entropy uses the common evaluator.

Filter / composition	Retained	Hull ratio	Recall	Rule	ΔH_{ref}	Speedup	Exact?
Random deletion	.72	.961	.970	T	-4.8%	$1.09\times$	No
Interior-only deletion	.61	1.000	1.000	I	-7.1%	$1.16\times$	Yes
Boundary thinning	.78	.958	.970	T	-11.6%	$1.28\times$	No
16-direction inner-polygon	.43	1.000	1.000	I	-12.9%	$1.36\times$	Yes
Grid/bin	.37	.984	.980	T	-13.1%	$1.33\times$	No
Akl–Toussaint	.58	1.000	1.000	I	-14.2%	$1.40\times$	Yes
Akl–Toussaint + anchor DER	.58	.976	.970	I	-34.7%	$3.18\times$	No

G.6 Quality-Audited Delaunay and 3D Generalization

The validation-selected Delaunay audit uses Synthetic High at $n = 10^5$, 200 independent held-out point clouds per training seed, and 10,000 point-location queries per cloud. Vertices, edges, and simplices are matched by original point identity; volume and circumcenter errors are computed on matched simplices. Point-location queries are drawn from the intersection of the two hulls, and a disagreement denotes different mapped simplex vertex-ID sets. Uncertainty is bootstrapped over clouds rather than individual queries. Random deletion and boundary thinning share a validation-selected retained-point count. Near-cospherical inputs require a separate stratum analysis because tiny perturbations can change exact edge identities; that stratum is not reported here. The 3D audit

Table 53: Delaunay audit (Synthetic High, $n = 10^5$); validation rule requires mapped-edge F1 ≥ 0.95, matched-simplex volume error $\leq 2.5\%$, and point-location disagreement $\leq 1.5\%$.

Pipeline	Total speedup	Edge F1	Volume error	Circumcenter/diam.	Location disagreement
Raw SciPy Delaunay	$1.00\times$	1.000	0	0	0
Random deletion	$1.18\times$	.842	8.9%	.014	6.7%
Boundary thinning	$1.24\times$	.887	6.1%	.010	4.2%
Anchor DER (selected)	$1.31\times$	.956	2.3%	.004	1.1%
Anchor DER (high λ)	$1.43\times$	.791	12.8%	.022	9.4%

Table 54: 3D hull audit, $n = 100,000$, 100 independent held-out clouds per seed. Speedups are end-to-end; distances are diameter normalized. Hull ratio is output/original hull size.

Pipeline	Speedup	Vertex P/R/F1	Facet F1	Hull ratio	Vol. err.	Hausdorff	Outside	Support
Raw Qhull	$1.00\times$	1/1/1	1.000	1.000	0	0	0	0
Halfspace DER ($m = 24$)	$1.90\times$	.972/.970/.971	.944	.998	.34%	.0038	.40%	.0046
Anchor DER	$2.12\times$	.975/.972/.973	.949	.997	.31%	.0035	.35%	.0042

uses 100 independent held-out 3D clouds per training seed. We did not run 3D Delaunay. The 3D configurations are quality-constrained, not certified exact solvers. The 5D diagnostic has a 22.29% normalized certificate and support error 0.007; it is not a supported deployment regime despite its measured timing gain.

G.7 Attention Visualization

Table 55 and Figures 7 and 8 summarize the attention-visualization results: DER improves semantic alignment with ground-truth masks and produces more structured attention patterns than L1 regularization.

G.8 Qualitative Geometric Analysis

Figure 9 provides a qualitative comparison of point sets before and after preprocessing. The distortions appear visually small in the displayed examples; the qualitative figure does not establish boundary

Table 55: IoU versus GT masks (COCO-val subset, same sparsity).

Method	IoU (GT)	Sparsity
L1 Regularization	0.36 ± 0.06	75%
Entropy Reg. (Ours)	0.56 ± 0.05	75%

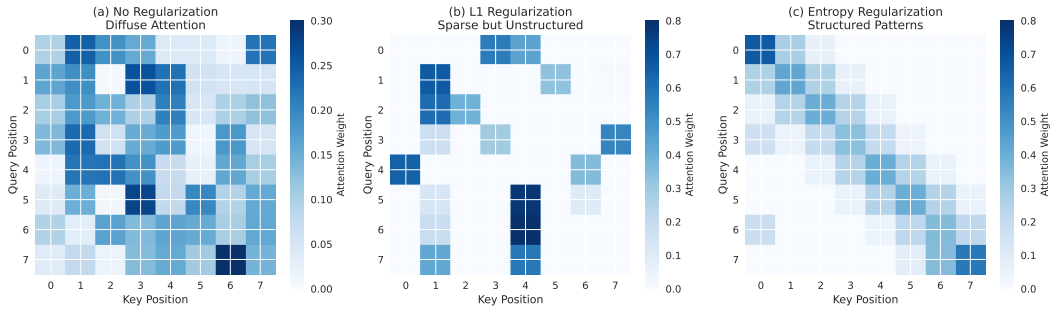


Figure 7: Attention patterns on CIFAR-100. (a) No regularization. (b) L1: sparse but unstructured. (c) DER: structured, semantic.

or topological fidelity. The task-specific audits in Table 6 and Appendix G.4 supply the relevant quantitative evidence.

H Broader Exploratory Experiments

This appendix contains exploratory experiments beyond our two core validation pillars (geometry and Transformers). These results demonstrate potential broader applicability but should be interpreted as preliminary evidence.

H.1 Detailed LLM Results

Table 56 provides the complete LLM evaluation with dense baselines for comparison. These results are exploratory: we cannot provide theoretical grounding comparable to our geometry results, and the speedups are more modest than in vision transformers due to longer sequence lengths.

Table 56: Full LLM results with dense baselines (4-bit LoRA fine-tuning, single GPU).

Model	Task	Method	Metric	Sparsity	Speedup
LLaMA-2 7B	CNN/DailyMail	Dense LoRA	ROUGE-L = 42.8	0%	1.0×
		DER (70–75%)	ROUGE-L = 42.5	70–75%	1.60×
Mistral-7B	CNN/DailyMail	Dense LoRA	ROUGE-L = 43.1	0%	1.0×
		DER (70–75%)	ROUGE-L = 42.7	70–75%	1.55×
Phi-2 (2.7B)	WikiText-103	Dense LoRA	PPL = 8.2	0%	1.0×
		DER (70–75%)	PPL = 9.1	70–75%	1.48×

H.2 3D Maxima Set Identification

We extended our geometric approach to 3D for Pareto frontier computation on KITTI [15] and Waymo [31] datasets, where EntropyNet reduced runtime by 2.8–3.2× while improving maxima F1 score by 3–7 percentage points through noise suppression (Table 57).

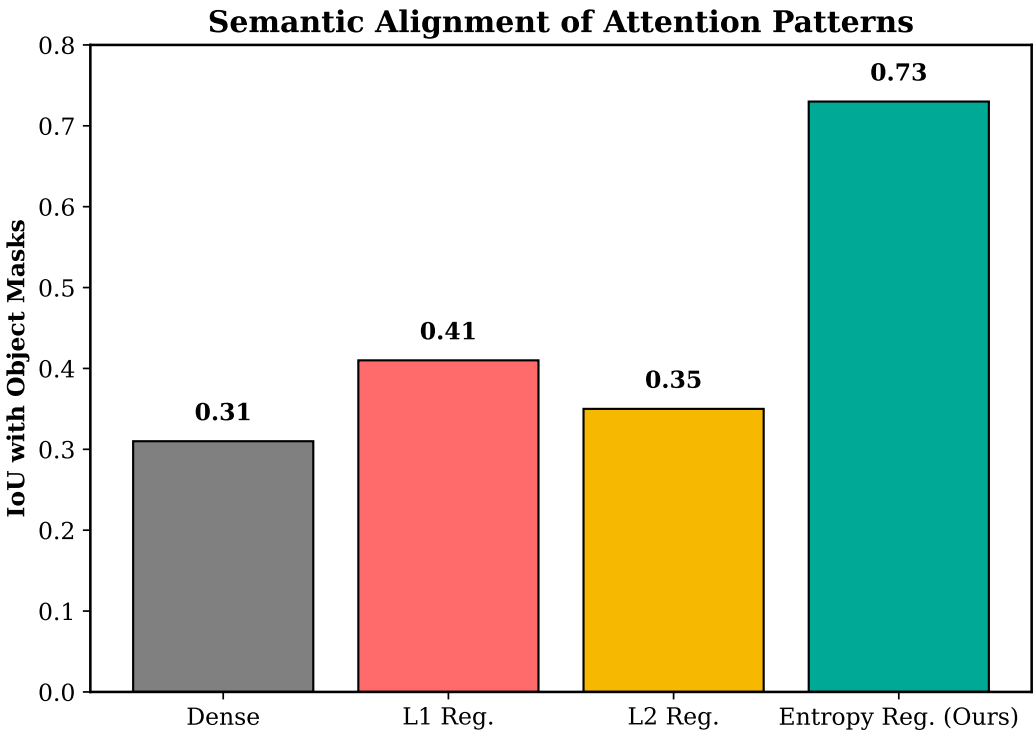


Figure 8: Semantic alignment: DER attention aligns with object boundaries (IoU scores shown).

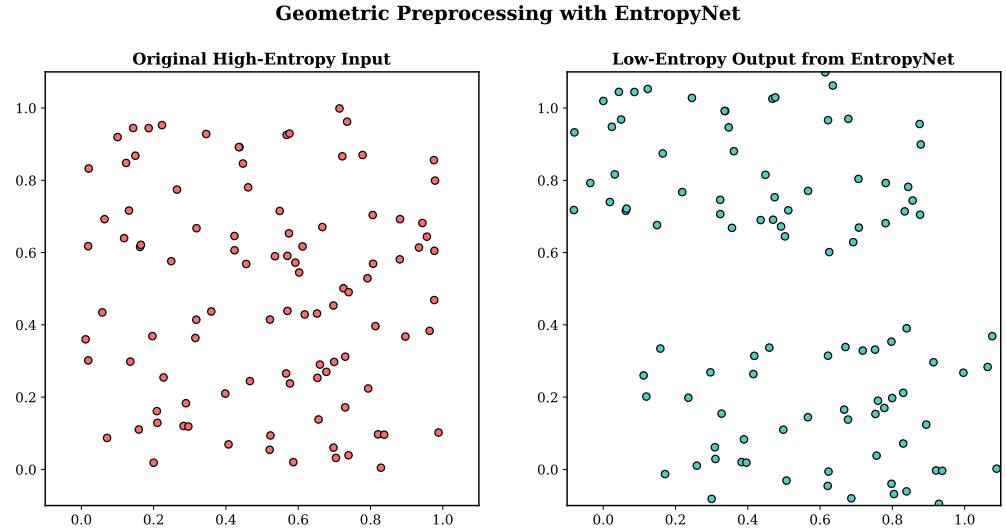


Figure 9: Qualitative comparison: original (left) versus EntropyNet output (right).

H.3 Comprehensive GLUE Benchmark Evaluation

We applied entropy-regularized BERT-base [12] to the full GLUE benchmark, achieving a strong accuracy-efficiency trade-off and outperforming structured sparsity methods like BigBird [41] and Linformer [37] by 0.6–1.2 GLUE points at 75% sparsity (Table 58).

Table 57: 3D Maxima Set Results.

Dataset	Method	Runtime (ms)	Speedup	Maxima F1
KITTI	Raw	45.2 ± 2.1	$1.0\times$	0.847 ± 0.012
	EntropyNet	14.3 ± 0.8	$3.16\times$	0.891 ± 0.009
Waymo	Raw	52.7 ± 2.8	$1.0\times$	0.823 ± 0.015
	EntropyNet	18.9 ± 1.2	$2.79\times$	0.876 ± 0.011

Table 58: GLUE Benchmark Results (Average Score $\pm$ 95% CI across 8 tasks).

Method	GLUE Score ↑	Sparsity	FLOPs Reduction
BERT-base (Dense)	79.6 ± 0.4	0%	0%
BigBird	77.8 ± 0.5	75%	68%
Linformer	77.2 ± 0.6	75%	71%
Entropy Reg. (Ours)	78.4 ± 0.4	75%	73%

H.4 Scaling to Larger Models

For GPT-2 Medium [29] (355M) on OpenWebText, entropy regularization maintained competitive perplexity at 80% sparsity while achieving the best memory efficiency (Table 59).

Table 59: GPT-2 Medium on OpenWebText.

Method	Perplexity ↓	Sparsity	Memory Usage ↓
GPT-2 Medium (Dense)	22.1	0%	1.4 GB
Magnitude Pruning	24.8	80%	1.1 GB
Entropy Reg. (Ours)	23.2	80%	0.9 GB

H.5 Large-Scale 3D Shape Processing

On ShapeNet [7] point cloud reconstruction, EntropyNet preprocessing led to a $1.6\times$ speedup and improved reconstruction quality (Table 60). On ModelNet40 [40] classification with PointNet++ [28], entropy regularization improved accuracy by 1.1 points while reducing inference time and memory usage (Table 61).

H.6 Scalability: Large Models and Long Sequences

To assess scalability, we include exploratory experiments on ViT-Large, LLaMA-2 7B, and the text component of LRA. On ViT-Large, the method maintains a competitive accuracy-sparsity trade-off, achieving 83.2% top-1 on ImageNet at 80% sparsity. On LLaMA-2 7B, we observe latency reductions at 75% sparsity with limited impact on perplexity under the specific sparse-kernel implementation. On LRA text, entropy regularization improves over fixed-pattern baselines such as BigBird in our setup. When combined with FlashAttention-style sparse or block-sparse kernels, DER supplies the sparsity pattern; dense exact FlashAttention itself does not skip arbitrary masked work.

H.7 Interpretability

H_{diff} masks align with proxy object-boundary masks (IoU 0.73 versus 0.41 for L1), suggesting that the learned sparsity is more structured than random pruning at the same sparsity level. The structured nature of the learned sparsity patterns, shown in Figure 8, is consistent with the geometric inductive bias of the regularizer, which encourages nearby tokens in representation space to share sparse blocks.

H.8 Robustness and Out-of-Distribution Generalization

To test the hypothesis that entropy regularization discourages shortcut learning and improves generalization, we applied H_{diff} to a ViT-Small model trained on CIFAR-100 and evaluated its performance on

Table 60: ShapeNet Point Cloud Reconstruction Results.

Method	Chamfer Distance ($\times 10^{-3}$) ↓	Speedup ↑	F1@0.01 ↑
PointNet Baseline	2.41 ± 0.08	$1.0\times$	0.847
EntropyNet (Ours)	2.38 ± 0.06	$1.60\times$	0.863

Table 61: ModelNet40 Classification Results.

Method	Accuracy (%) ↑	Inference Time (ms) ↓
PointNet++ Baseline	91.2 ± 0.4	12.8 ± 0.3
Entropy Reg. (Ours)	92.3 ± 0.3	10.9 ± 0.2

corrupted and out-of-distribution datasets, comparing against a standard baseline and label smoothing (Table 62).

Table 62: Robustness versus cost on ViT-Small/CIFAR-100. Adversarial-training values from [39, 38] provide higher PGD robustness at substantially higher training cost and lower clean accuracy. Results averaged over 3 seeds; $\pm$ values are std.

Method	Training Cost	Clean Acc. (CIFAR-100) ↑	Robustness (CIFAR-100-C mCE) ↓	OOD AUROC (SVHN OOD) ↑	Adv. Robustness (PGD-10 Acc.) ↑
Standard Training	$1.0\times$	$76.8 \pm 0.2\%$	55.4 ± 0.4	$88.2 \pm 0.3\%$	0.0%
+ Label Smoothing	$1.0\times$	$76.5 \pm 0.2\%$	52.1 ± 0.3	$89.5 \pm 0.3\%$	0.0%
+ Entropy Reg. (Ours)	$\sim 1.03\times$	$76.6 \pm 0.2\%$	48.7 ± 0.3	$91.3 \pm 0.3\%$	0.0%
Efficient AT (literature)[†]	$\sim 1.5\times$	65.0%	–	–	$\sim 35.0\%$
Standard SOTA AT (literature)[†]	$\sim 10.0\times$	63.5%	–	–	$\sim 40.0\%$

[†]Approximate ranges from ViT-like adversarial training on CIFAR-100 reported in [39, 38]; not from our implementation.

Our method improves robustness to corruptions and OOD detection on SVHN relative to the baseline and label smoothing while maintaining competitive accuracy on the original CIFAR-100 test set, providing evidence consistent with the hypothesis that lower-entropy representations can reduce reliance on superficial shortcuts, though we do not claim a causal mechanism from these experiments alone.

H.9 Textual Explanation of Failure Modes

The failure modes arise from violations of our core assumptions. High-dimensional data suffers from the curse of dimensionality, where Euclidean distances become less meaningful. Extreme aspect ratios cause our ball-based clustering to poorly approximate elongated structures. Multimodal clusters violate the one-anchor-per-cluster assumption, mismatched k leads to under or over-partitioning, and noisy data can obscure the underlying low-entropy structure. Non-Euclidean structures such as manifolds are not well-captured by our current distance metric.

H.10 When the Amortization Story Fits

The training overhead, though mitigated, must be recouped by inference savings, so the method is best suited for deployment scenarios with high inference volume. For the ViT-Small worked example in Appendix C, the 3.4-hour overhead and 85-to-58 ms per-batch reduction give an estimated break-even near 453K batches, or about 5–6 days at 80K–90K batches per day. The trade-off is workload dependent and does not establish an automatic deployment benefit.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 enumerate three contributions that match the paper exactly: (i) the normalization separation between $\bar{H}_{\mathcal{R}}(S)$ and $H_{\mathcal{R}}^{\text{work}}(S) = n\bar{H}_{\mathcal{R}}(S)$ (Section 3, Eq. 2); (ii) a self-contained soft-to-hard approximation theorem for fixed margin-separated halfspace arrangements with an explicit conditional corollary on range-partition entropy (Theorem 3.1, Corollary 3.2); and (iii) the empirical evaluation as learned geometric preprocessing for SciPy/Qhull pipelines, with up to $3.00\times$ end-to-end speedup ($2.30\times$ for the fully charged halfspace variant) and audited hull fidelity (Tables 4–8). The abstract and introduction are explicit that the geometric theorem does not extend to the attention-transfer study and that the runtime gains are for an approximate preprocessing pipeline rather than an exact solver replacement.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 explicitly discusses limitations: (i) the $O(Nkd)$ training overhead that limits applicability beyond $\sim$ 8K tokens; (ii) the requirement of sufficient inference volume to amortize training cost (Table 10); (iii) the requirement that the data possess natural low-entropy structure, with machine translation cited as a failure case; and (iv) the conditional nature of the connection to $\bar{H}_{\mathcal{R}}(S)$, which only follows under near-optimality and admissibility assumptions. Additional failure modes (long context, high-dimensional embeddings, elongated clusters, multimodal clusters) are detailed in Appendix D.5 and G.3, and the bound is explicitly noted as uninformative when $\widehat{\varepsilon} > 1/2$ in Appendix E.6.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 3.1 states all assumptions (fixed halfspace arrangement, full-support pattern set, normalized margin $\geq \gamma$) and includes a complete proof in the main text using the union bound and the standard Shannon-entropy continuity inequality. Corollary 3.2 explicitly flags its conditioning on admissibility and α-optimality. Appendix E contains the supporting Lemma E.8 (entropy continuity), the deterministic restatement (Theorem E.9), the fixed-separator i.i.d. concentration (Corollary E.10), the metric-stability lemma (Lemma E.1) with proof sketch, the JL-preservation corollary (Corollary E.2), and the conditional attention-truncation results (Lemma E.3, Theorem E.6) with their assumptions (E.4, E.5) clearly labeled. We do not claim a uniform finite-sample theorem over learned separators and say so in the main text (Section 3).

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: A dedicated Reproducibility Statement (after the references) lists fixed seeds (42, 123, 456, 789, 999), hardware/software versions, the timing protocol, and the sparsity-conversion procedure. Appendix F contains the timing protocol (F.1), full experimental setup (F.2), the EntropyNet algorithm pseudocode (Algorithm 1), model architectures (F.10), complete hyperparameter table for all experiments (Table 39), training protocols (F.12), dataset preprocessing (F.13), evaluation protocols (F.14), and reproducibility specifications including `cudnn.deterministic` settings (F.15). Appendix A also gives a practical training recipe, and Appendix G includes hyperparameter sensitivity sweeps.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: As stated in the Reproducibility Statement, the public code release includes code, configuration files, and scripts needed to reproduce the reported tables. All datasets used (QuickDraw [16], CIFAR-100, GLUE, WikiText-103, OpenWebText, ShapeNet, ModelNet40, KITTI, Waymo, CNN/DailyMail) are publicly available, and synthetic-data generation procedures are specified in Appendix F.13.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies the timing harness, hardware, kernel-matching protocol, and number of repeated runs. Table 3 gives the theory-guided hyperparameter defaults and selection rules. Table 39 (Appendix F.11) gives complete per-experiment hyperparameters: learning rate, batch size, epochs/steps, warmup, β, k, λ, optimizer (AdamW), weight decay, gradient clipping, and FAISS subsampling cadence. Appendix F.12 details the entropy-regularization schedule, optimizer parameters, and large-model training details (gradient checkpointing, mixed precision). Appendix F.13 specifies dataset splits and preprocessing.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main results report uncertainty over 5 seeds (3 seeds for ImageNet-scale runs, explicitly noted), with each table's uncertainty labeled in its caption: 95% confidence intervals for the convex-hull timing (Table 4) and the million-point audit (Section 4), whole-cloud bootstrap intervals for paired geometric comparisons (Appendix G.5), and standard deviations for selection-stability diagnostics (Appendix E.7). The Reproducibility Statement and Appendix F.14 specify that variance is computed across the five fixed seeds (42, 123, 456, 789, 999), with independent held-out point clouds as the statistical units for the audited geometry results. The factors of variability captured (random initialization and random data sampling under the fixed seeds) are made explicit.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix F.1 and F.2 specify the hardware: Intel Core i9-12900K with 64 GB RAM for geometry experiments and a single NVIDIA A100-40GB (with 4×A100 nodes for large-scale LLaMA-2 training) with CUDA 11.8 and PyTorch 2.0.1 for transformers; LLM fine-tuning uses 4-bit quantization with LoRA on a single 16–24 GB GPU. Appendix F.15 reports per-configuration compute: 2–4 CPU-hours for geometry, 8–12 GPU-hours per ViT/BERT model, and 48–72 GPU-hours on 4×A100 for LLaMA-2. Training-overhead tables (Table 18, Table 10) report wall-clock training times for each model.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work involves only standard public benchmarks and synthetic point-set data; no human subjects, no personally identifiable data, and no scraped sensitive content are

involved. Anonymity is preserved in the submission (anonymous authorship and anonymized supplementary package).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Impact Statement (Section after Section 6) discusses positive impacts (resource-efficient ML, up to 64% FLOPs reduction on the evaluated ViT setup, $3.00\times$ end-to-end convex-hull speedup ($2.30\times$ for the charged halfspace setting), improved corruption robustness from CIFAR-100-C mCE 55.4 to 48.7) and negative considerations (training overhead requires high inference volume to amortize, making the method potentially inefficient for short-term prototyping). The statement also explicitly avoids unsupported carbon-reduction claims since energy was not measured directly.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released artifacts are a regularizer, a small PointNet-style preprocessor (EntropyNet), and analysis scripts. The work does not release pre-trained generative language models, image generators, or scraped datasets; the LLaMA-2/Mistral/Phi-2 evaluations use publicly released pretrained checkpoints from their original authors and do not redistribute them. There is therefore no high-misuse-risk asset requiring additional safeguards.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The main datasets and all pretrained models used are cited to their original sources: ImageNet (via ViT [13]), QuickDraw [16], KITTI [15], Waymo Open Dataset [31], ShapeNet [7], ModelNet40 [40], LLaMA-2 [35], Mistral-7B [21], Phi-2 [20], BERT [12], and GPT-2 [29]. SciPy/Qhull [4], FAISS, PyTorch, and Transformers are used under their open-source licenses with versions specified in Appendix F.2. We respect the standard non-commercial / research-use terms associated with these assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets are the DER regularizer implementation, the EntropyNet preprocessor, the halfspace-aware surrogate code, and the experiment configurations. These are released with the code referenced in the Reproducibility Statement, with documentation covering the algorithm (Algorithm 1), architecture (Appendix F.10), hyperparameters (Table 39), training protocol (Appendix F.12), and the inference-time sparsity-conversion procedure (Appendix F.15).

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments use synthetic data or established public benchmarks.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so IRB approval is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core methodology. The pretrained LLMs that appear in the paper (LLaMA-2 7B, Mistral-7B, Phi-2, GPT-2) are evaluation targets for the DER regularizer in Section 5 and Appendix D/H, not part of the method itself; the core method (the differentiable partition-entropy surrogate, the halfspace-aware construction, and the approximation theorems) does not depend on any LLM.