AAC: Admissible-by-Architecture Differentiable Landmark Compression for ALT

Anonymous authors
Paper under double-blind review

Abstract

We introduce **AAC** (Architecturally Admissible Compressor), a differentiable landmark-selection module for ALT (A*, Landmarks, and Triangle inequality) shortest-path heuristics whose outputs are admissible by construction: each forward pass is a row-stochastic mixture of triangle-inequality lower bounds, so the heuristic is admissible for *every* parameter setting without requiring convergence, calibration, or projection (Proposition 2). At deployment the module reduces to classical ALT on a learned subset (Proposition 4), composing end-to-end with neural encoders while preserving the classical toolchain. The construction is the first differentiable instance of the compress-while-preserving-admissibility tradition in classical heuristic search.

Under a matched per-vertex memory protocol we establish that ALT with farthest-point-sampling landmarks (FPS-ALT) has provably near-optimal coverage on metric graphs (Theorem 5, Corollary 6), leaving at most a few percentage points of headroom for *any* selector. AAC operates near this ceiling: the gap is 0.9–3.9 percentage points on 9 road networks and $\leq$1.3 percentage points on synthetic graphs, with zero admissibility violations across 1,500+ queries and all logged runs. At matched memory AAC is also 1.2–1.5$\times$ faster than FPS-ALT at the median query on DIMACS road networks, amortizing its offline cost within 170–1,924 queries. A controlled ablation isolates the binding constraint: training-objective drift under default initialization, not architectural capacity; identity-on-first-m initialization closes the expansion-count gap entirely. We release the module, a reusable matched-memory benchmarking protocol with paired two-one-sided-test (TOST) equivalence and pre-registration, and a reference compressed-differential-heuristics baseline.

1 Introduction

Point-to-point shortest-path search is fundamental to routing, robotics, and planning. A* search (Hart et al., 1968) with an admissible heuristic $h(u, t) \leq d(u, t)$ guarantees optimal paths. The quality of h directly determines search efficiency: a tighter lower bound means fewer node expansions.

The ALT framework (Goldberg & Harrelson, 2005) provides a natural admissible heuristic family: landmark distances yield lower bounds via the triangle inequality, achieving 82–95% expansion reduction on road networks (Goldberg & Harrelson, 2005; Goldberg et al., 2007; Bast et al., 2016). ALT's farthest-point sampling (FPS) landmark selection is fixed and does not adapt to the downstream search objective or query distribution. Bauer et al. (2010) proved that optimal ALT landmark selection is NP-hard; FPS, together with the avoid and maxcover heuristics of Goldberg & Harrelson (2005), are the established practical approximations, with a 2-approximation guarantee on covering radius via Gonzalez (1985).

Learned heuristics can adapt to graph structure, but exact admissibility is typically sacrificed. Neural methods such as PHIL (Pandy et al., 2022), Neural A* (Yonetani et al., 2021), TransPath (Kirilenko et al., 2023), UPath (Ananikian et al., 2026), and iA* (Chen et al., 2025) learn powerful heuristics yet provide no admissibility guarantee. Existing remedies tie the guarantee to training convergence: truncated-Gaussian likelihoods lower-bounded by a classical heuristic (Nunez-Molina et al., 2024) and CEA-loss training (Futuhi & Sturtevant, 2026) achieve near-admissibility but cannot guarantee zero violations under distribution shift,

early stopping, or out-of-distribution queries. Admissibility is thus typically enforced via loss regularization, post-hoc calibration, or abandoned entirely. This leaves a gap: classical admissible heuristics (ALT, hub labels, compressed differential heuristics (CDH)) that cannot be trained, and learned heuristics that cannot guarantee admissibility.

The selector is the missing differentiable piece. Two decades of work on ALT has improved the *representation* of landmark labels (Compressed Differential Heuristics, CH-Potentials, Pruned Landmark Labeling, Landmark Hub Labeling); the *selection* of which landmarks to precompute remains a non-differentiable combinatorial step dominated by FPS, whose 2-approximation to metric K-center (Gonzalez, 1985) is provably tight. Embedding selection in a differentiable pipeline is appealing, since it would let the selector adapt to query distribution, graph structure, or downstream task, but existing learned heuristics buy differentiability by abandoning admissibility. We ask a narrower question: can selection itself be differentiable and admissible *by architecture* rather than by loss or projection? What does that buy at matched landmark memory?

AAC: an admissible-by-architecture differentiable compressor. We construct **AAC** (Architecturally Admissible Compressor), a row-stochastic Gumbel-softmax compressor of an FPS-generated K_0-landmark teacher pool down to m deployed landmarks. Each compressed heuristic term is a convex combination of admissible teacher terms and therefore cannot exceed the teacher maximum (Proposition 2, a corollary of Pearl 1984's convex-combination argument applied to ALT). At deployment with hard argmax the heuristic reduces to ALT on the learned forward/backward landmark subset (Proposition 4); admissibility holds for every parameter setting, including under training divergence, early stopping, or hyperparameter drift (Table 11). The Gumbel-softmax relaxation makes the same compressor end-to-end differentiable, composing with neural encoders while preserving the deployment-time admissibility certificate (Appendix F).

What the matched-memory evidence shows. We evaluate AAC against FPS-ALT under a matched per-vertex memory protocol with paired two-one-sided-test (TOST) equivalence testing, false-discovery-rate (FDR) corrected Wilcoxon signed-rank tests, and a pre-registration audit (Section 5.2). The evidence reveals a structural finding: FPS coverage is provably near-optimal on the tested metric graphs (Theorem 5, Corollary 6), leaving limited headroom for any ALT-family selector; the binding constraint on closing the residual gap is training-objective drift under default initialization, not architectural capacity (Section 5.9). Per-regime numbers are in Table 2 and Sections 5.3–5.8.3.

Contributions. (C1) A differentiable admissible landmark selector. AAC is, to our knowledge, the first differentiable instance of the compress-while-preserving-admissibility tradition (Felner et al. 2007; Goldenberg et al. 2011; 2017) for the ALT family. Admissibility holds for every parameter setting (Proposition 2, after Pearl 1984); at deployment, hard argmax reduces the module to classical ALT on a learned subset (Proposition 4). The trained module is pool-agnostic and recovers ALT, CDH, and FPS as architectural special cases; the same compressor composes with differentiable encoder pipelines without losing the deployment-time admissibility certificate (Appendix F).

(C2) Training-objective drift inside an admissible family. The architecture trivially admits the FPS-on-pool subset (one-hot rows on the first m FPS indices recover FPS-ALT $K{=}m$ exactly; Proposition 4). Yet under default block-sparse initialization the gap-to-teacher gradient drifts the trained selector *away* from this subset (Section 5.9, Figure 7); identity-on-first-m initialization closes the gap entirely (Appendix Table 20), pinpointing initialization as the binding constraint. Even a query-adaptive greedy oracle beats FPS by at most 0.3–1.1 % (Section 5.7), confirming limited headroom for any selector.

(C3) A reusable matched-memory protocol. We release the val-split + FDR-corrected Wilcoxon + paired TOST protocol, a directionality-aware accounting convention (Section 5.2, Appendix E E.2), a reference CDH baseline (Goldenberg et al., 2011; 2017) with bound substitution and Bidirectional Pathmax (BPMX), and the full audit trail of our pre-registered OGB-arXiv prediction (Appendix E).

2 Background and Notation

2.1 Graphs, A* Search, and Admissibility

Let $G = (V, E, w)$ be a weighted graph with vertex set V, edge set E, and positive edge weights $w : E \to \mathbb{R}_{>0}$. Let $d(u, v)$ denote the shortest-path distance from u to v. The function d satisfies the triangle inequality: $d(u, v) \leq d(u, z) + d(z, v)$ for all $u, z, v \in V$ with finite distances. For undirected graphs, $d(u, v) = d(v, u)$; for directed graphs, $d(u, v) \neq d(v, u)$ in general.

Reachability assumption. All theorems assume finite shortest-path distances between the relevant vertex pairs; on directed graphs unreachable landmark distances are masked out before heuristic computation, and query pairs are restricted to a common strongly connected component. An automated strongly connected component (SCC) audit confirms all 100 standard-protocol queries lie in a common SCC on eight of the nine road-network graphs used in this paper (Netherlands omitted from the audit due to its 4.5M-node scale); the sentinel value and masking implementation are described in Appendix G.

A* search. A* (Hart et al., 1968) maintains a priority queue ordered by $f(v) = g(v) + h(v, t)$, where $g(v)$ is the cost of the best known path from source s to v, and $h(v, t)$ is a heuristic estimate of the remaining distance to target t. A heuristic is *admissible* if $h(v, t) \leq d(v, t)$ for all $v, t \in V$. Our implementation uses graph-search A* without node reopenings (closed-set). For optimality without reopenings, the heuristic must also be *consistent* ($h(u, t) \leq w(u, v) + h(v, t)$ for all edges (u, v)). ALT heuristics are consistent by the triangle inequality when all landmark distances are finite: $h_{\mathrm{ALT}}(u, t) \leq w(u, v) + h_{\mathrm{ALT}}(v, t)$ follows from the triangle inequality applied to each landmark term. Since AAC at inference selects a fixed subset of ALT landmarks (one-hot rows applied uniformly to all vertices), the deployed heuristic inherits consistency in the all-finite regime. Under the directed-graph masking of the reachability assumption above, admissibility is preserved (masking only removes terms from a max), but consistency is guaranteed only when the set of landmarks with finite distances does not vary between adjacent vertices; in practice this holds when the query and selected landmarks lie within a common strongly connected component.

2.2 ALT: Landmarks and Triangle Inequality

ALT heuristics. The ALT framework (Goldberg & Harrelson, 2005) selects K *landmark* vertices $\{l_1, \ldots, l_K\}$ and precomputes distances from each landmark to all vertices via Dijkstra's algorithm (Dijkstra, 1959). For directed graphs, both *forward* distances $d_{\mathrm{out}}(k, v) := d(l_k, v)$ and *backward* distances $d_{\mathrm{in}}(k, v) := d(v, l_k)$ are stored, requiring $2K$ floats per vertex. The ALT heuristic is:

$$h_{\mathrm{ALT}}(u, t) = \max\Big(\max_k \big(d_{\mathrm{out}}(k, t) - d_{\mathrm{out}}(k, u)\big),\ \max_k \big(d_{\mathrm{in}}(k, u) - d_{\mathrm{in}}(k, t)\big),\ 0 \Big). \tag{1}$$

Theorem 1 (ALT admissibility)**.** *$h_{\mathrm{ALT}}(u, t) \leq d(u, t)$ for all $u, t \in V$.*

Proof. We show each component is bounded by $d(u, t)$.

Forward bound. By the triangle inequality, $d(l_k, t) \leq d(l_k, u) + d(u, t)$. Rearranging: $d(l_k, t) - d(l_k, u) \leq d(u, t)$.

Backward bound. By the triangle inequality, $d(u, l_k) \leq d(u, t) + d(t, l_k)$. Rearranging: $d(u, l_k) - d(t, l_k) \leq d(u, t)$.

Since both bounds hold for every k, taking the maximum over k and over the two bound types preserves the inequality. The clamp at zero is trivially $\leq d(u, t)$ since distances are non-negative. □

For undirected graphs where $d_{\mathrm{out}} = d_{\mathrm{in}}$, the two bounds collapse to $h_{\mathrm{ALT}}(u, t) = \max_k |d(l_k, u) - d(l_k, t)|$.[1]

[1] Classical ALT takes a max over landmarks of admissible lower bounds (each per-landmark triangle-inequality bound is already $\leq d(u, t)$, so the tightest admissible value is the largest). Throughout this paper every max over landmarks or compressed dimensions follows Eq. (1), applied *after* row-stochastic mixing in the compressed case (Eq. (4)); readers familiar with min-based formulations of admissible upper bounds should read every max here in the standard ALT lower-bound sense.

Landmark selection. The quality of the ALT heuristic depends on the choice of landmarks. *Farthest-point sampling* (FPS) (Gonzalez, 1985) iteratively selects the vertex farthest from the already-chosen set, producing a K-element subset that is a 2-approximation for the K-center problem: the maximum distance from any vertex to its nearest landmark is at most twice the optimal value. Alternative strategies include random selection (fast but no coverage guarantee), planar partition landmarks (exploiting graph planarity when available), and avoid/maxcover heuristics tuned to specific query distributions (Goldberg & Harrelson, 2005). FPS is the default in our implementation due to its simplicity and guaranteed coverage quality.

Covering radius. Given a landmark subset $S = \{l_1, \ldots, l_m\} \subseteq L$, the **covering radius** of S is

$$r_m := \max_{v \in V} \min_{l \in S} d(v, l). \tag{2}$$

This is the worst-case distance from any vertex to its nearest landmark in S. A smaller r_m means S covers the graph more tightly. For FPS selection, the Gonzalez 2-approximation guarantee gives $r_m \leq 2\, r_m^*$, where r_m^* is the optimal m-center covering radius. The covering radius controls the gap between the ALT heuristic on the subset S and the true distance (Theorem 5, Section 3.3). Increasing m monotonically decreases r_m; the special case $m = K$ (all teacher landmarks selected) gives $r_K = 0$ only if $L = V$, which is impractical; in practice, r_m decreases with m but remains positive.

For directed graphs, the covering radius uses the symmetrized metric $r_m^{\mathrm{sym}} := \max_{v \in V_{\mathrm{reach}}} \min_{l \in S} \max\big(d(l, v),\, d(v, l)\big)$, where V_{reach} is the reachable vertex set (Section 2.1). Both forward and backward landmark distances must be bounded for the ALT heuristic to use landmark l for vertex v.

Memory model. ALT preprocessing stores shortest-path distances from each landmark to every vertex. For directed graphs, both forward distances $d(l_k, v)$ and backward distances $d(v, l_k)$ are required, totaling $2K$ floats per vertex ($2K|V|$ overall). For undirected graphs, $d(l_k, v) = d(v, l_k)$, so K floats per vertex suffice. At $K = 64$ landmarks and $|V| = 10^6$ vertices, directed ALT requires ≈ 488 MB in float32 or ≈ 977 MB in float64.

Complexity. *Preprocessing:* K runs of Dijkstra's single-source shortest-path (SSSP) algorithm, each costing $O(|V| \log |V| + |E|)$ with a binary heap, giving $O(K(|V| \log |V| + |E|))$ total. For directed graphs, an additional K reverse SSSPs on the transposed graph are needed (same asymptotic cost). *Query time:* Evaluating $h_{\mathrm{ALT}}(u, t)$ requires $O(K)$ lookups and comparisons. *Total A* cost:* The heuristic evaluation is $O(K)$ per node expansion; full A* query time additionally scales with the number of expanded nodes, which depends on graph structure and heuristic tightness.

3 Method: AAC

This section specifies the AAC module: a preprocessing pipeline, a row-stochastic compressor, and a training objective that together close *admissibility by architecture* (Section 1). The structural insight is that at deployment, with hard argmax, AAC reduces to ALT on the learned forward/backward landmark subset (Proposition 4): learning chooses which teacher landmarks survive, while the row-stochastic architecture guarantees admissibility for *every* parameter setting, even under early stopping, training divergence, or hyperparameter drift. This single property gives the four benefits stated in Section 1: no dependence on training convergence, end-to-end differentiability, a classical-ALT deployment target, and a clean substrate for matched-memory studies. The rest of the section specifies how the pipeline realizes those benefits.

For undirected graphs, the compressed representation uses $m < K_0$ dimensions. For directed graphs, full ALT uses $2K_0$ directional labels (forward and backward); AAC uses $m = m_{\mathrm{fwd}} + m_{\mathrm{bwd}}$ directional dimensions with $m_{\mathrm{fwd}}, m_{\mathrm{bwd}} \leq K_0$, and $m = 2K_0$ recovers ALT exactly. Each compressed dimension selects one teacher landmark; uniqueness is encouraged by a regularizer (R_{uniq}, Section 3.4) but not enforced by the architecture, so duplicate selections are possible.

Table 1: Core notation used throughout Section 3.

Symbol	Meaning
K_0	number of teacher (FPS) landmarks used at preprocessing
m	number of compressed dimensions retained at deployment (undirected)
$m_{\text{fwd}}, m_{\text{bwd}}$	forward / backward compressed dimensions (directed; $m = m_{\text{fwd}} + m_{\text{bwd}}$)
h_{ALT}	full-ALT heuristic over all K_0 teacher landmarks
h^S_{ALT}	ALT restricted to landmark subset $S \subseteq [K_0]$ (deployment-equivalent form)
h_{AAC}	AAC heuristic with hard argmax (deployment)
$\tilde{h}_{\text{AAC}}$	AAC heuristic with soft row-stochastic weights (training surrogate)
B	row-stochastic selection matrix of size $m \times K_0$
$R_{\text{uniq}}, R_{\text{ent}}$	uniqueness / entropy regularizers on B

3.1 Preprocessing Pipeline

1. **Anchor selection.** Select K_0 landmarks via farthest-point sampling on the largest weakly connected component.
2. **Teacher labels.** Compute K_0 forward SSSPs to obtain $d_{\text{out}}(k, v)$ for all $k \in [K_0]$, $v \in V$. For directed graphs, also compute K_0 reverse SSSPs (on the transposed graph) to obtain $d_{\text{in}}(k, v)$.
3. **Learned compression.** Train a row-stochastic selection matrix (Section 3.2) to select m landmarks from the K_0 teacher pool.
4. **Deploy.** Store m compressed values per vertex. At query time, evaluate $h(u, t)$ in $O(m)$.

Cost breakdown.

- **Preprocessing (offline):** $O(K_0(V \log V + E))$ for K_0 SSSP computations from teacher landmarks.
- **Compressor training (offline):** $O(E_{\text{train}} \cdot m \cdot K_0)$ for E_{train} training epochs over the $m \times K_0$ selection matrix (typically 200 epochs, negligible relative to SSSP).
- **Deployment:** Storage of $O(Vm)$ floats. Each heuristic lookup $h(u, t)$ in $O(m)$; full A* query time additionally scales with node expansions.

For the contextual variant, deployment additionally requires per-instance anchor SSSP on the predicted-cost graph before the $O(m)$-per-lookup heuristic evaluation. Figure 1 lays the three stages out side by side: **(01) Preprocessing** produces the K_0 teacher labels, **(02) Learning** trains the soft row-stochastic compressor A under the gap-closing loss with the gradient $\nabla_\theta \mathcal{L}$ flowing back into A, and **(03) Deployment** hardens A to one-hot rows so the inference heuristic reduces to classical ALT on a learned m-landmark subset (Proposition 4). The certificate ribbon at the top of the figure restates the architectural bound $h_A(u, t) \leq h_{\text{ALT}}(u, t) \leq d(u, t)$ that holds for every parameter θ (Proposition 2); the per-stage cost annotations under each panel give the offline / online resource budget at a glance.

3.2 Row-Stochastic Compression

Definition. The compressor maintains a learnable logit matrix $W \in \mathbb{R}^{m \times K_0}$. For directed graphs, the m compressed dimensions are split: $m_{\text{fwd}} = \lfloor m/2 \rfloor$ forward dimensions and $m_{\text{bwd}} = m - m_{\text{fwd}}$ backward dimensions, with separate matrices $W_{\text{fwd}} \in \mathbb{R}^{m_{\text{fwd}} \times K_0}$ and $W_{\text{bwd}} \in \mathbb{R}^{m_{\text{bwd}} \times K_0}$.

Training. The selection matrix A is sampled from a hard Gumbel-softmax over the logits W/τ with the straight-through estimator (Jang et al., 2017; Maddison et al., 2017; Bengio et al., 2013) (full sampling formula in Appendix A). Temperature τ is annealed exponentially from 1.0 to 0.1; the forward pass produces one-hot rows (hard argmax), the backward pass uses the soft Gumbel-softmax gradients. Since one-hot vectors are row-stochastic (non-negative entries summing to 1), A is row-stochastic in every forward pass.

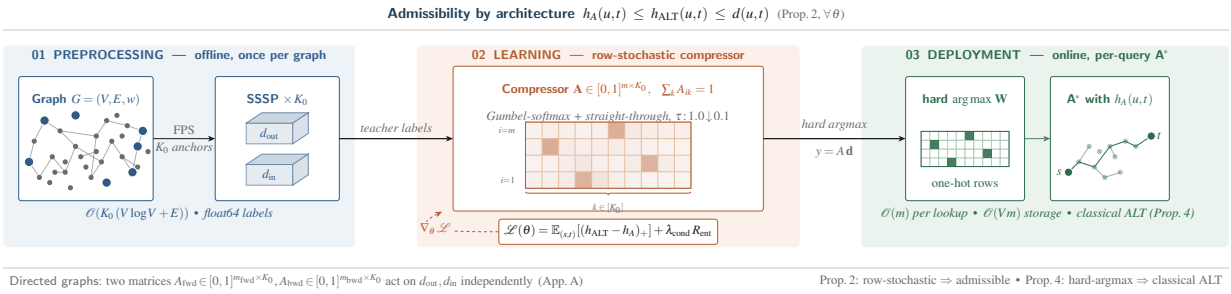

Figure 1: AAC method overview. The top ribbon states the architectural admissibility certificate $h_A(u,t) \le h_{\mathrm{ALT}}(u,t) \le d(u,t)$ that holds for every parameter θ (Proposition 2). **(01) Preprocessing** (offline, once per graph): FPS selects K_0 teacher anchors and K_0 SSSPs yield forward/backward distance labels $d_{\mathrm{out}}, d_{\mathrm{in}}$. **(02) Learning**: the row-stochastic compressor $A \in [0,1]^{m\times K_0}$ (each row sums to 1) is trained with Gumbel-softmax + straight-through (τ:1.0↓0.1) under the gap-closing loss $\mathcal{L}(\theta) = \mathbb{E}_{(s,t)}[(h_{\mathrm{ALT}} - h_A)_+] + \lambda_{\mathrm{cond}} R_{\mathrm{ent}}$; the dashed orange path is the SGD update $\nabla_\theta \mathcal{L}$. **(03) Deployment** (online, per-query A*): hard argmax on each row of W yields one-hot rows; the heuristic reduces to classical ALT on $\{l_{j^*(i)}\}_{i=1}^m$ (Proposition 4), with $O(m)$ per lookup and $O(Vm)$ storage. Directed graphs use independent forward/backward compressors $A_{\mathrm{fwd}}, A_{\mathrm{bwd}}$ acting on $d_{\mathrm{out}}, d_{\mathrm{in}}$ (Appendix A).

Design choices. AAC applies Gumbel-softmax independently per compressed dimension (per row of A), matching the L2X (Chen et al., 2018) selector primitive; the Concrete Autoencoder (Balın et al., 2019) and IP-CAE (Nilsson et al., 2024) are global-top-k predecessors. The novelty is not the selector primitive but its composition with the ALT structure: the row-stochastic constraint simultaneously enables differentiable selection *and* serves as the admissibility certificate via Pearl (1984)'s convex-combination argument (Section 4 places AAC in the four-lane taxonomy of admissibility-aware learning).

The compressed labels are computed as:

$$y(v) = A \cdot \mathbf{d}(v), \quad \text{where } \mathbf{d}(v) = \big(d(l_1, v), \ldots, d(l_{K_0}, v)\big)^\top. \tag{3}$$

For directed graphs, forward labels use $y_{\mathrm{fwd}}(v) = A_{\mathrm{fwd}} \cdot \mathbf{d}_{\mathrm{out}}(v)$ and backward labels use $y_{\mathrm{bwd}}(v) = A_{\mathrm{bwd}} \cdot \mathbf{d}_{\mathrm{in}}(v)$.

Inference. At deployment, hard argmax selection over W's rows reduces each compressed output to a single teacher landmark distance, $y_i(v) = d(l_{j^*(i)}, v)$ with $j^*(i) = \mathrm{argmax}_j\, W_{i,j}$ (Appendix A). Each row is one-hot, a special case of row-stochastic. Consequently, the deployed system performs standard ALT subset selection: only the method of *choosing* which subset is learned; the heuristic evaluation itself is identical to ALT on m landmarks.

Heuristic evaluation. The compressed heuristic is defined as:

$$h_A(u,t) = \max\Big(0, \underbrace{\max_{i\in[m_{\mathrm{bwd}}]} \big(y_i^{\mathrm{bwd}}(u) - y_i^{\mathrm{bwd}}(t)\big)}_{\text{backward bound}}, \underbrace{\max_{i\in[m_{\mathrm{fwd}}]} \big(y_i^{\mathrm{fwd}}(t) - y_i^{\mathrm{fwd}}(u)\big)}_{\text{forward bound}}\Big) \tag{4}$$

for directed graphs, and $h_A(u,t) = \max_i |y_i(u) - y_i(t)|$ for undirected graphs.

Extension to directed graphs. AAC extends to directed graphs using the directed-ALT formula $\max\big(d(u,L) - d(t,L),\, d(L,t) - d(L,u)\big)$ of Goldberg & Harrelson (2005): we instantiate two row-stochastic matrices $A_{\mathrm{fwd}}, A_{\mathrm{bwd}}$ (sizes $m_{\mathrm{fwd}} \times K_0$ and $m_{\mathrm{bwd}} \times K_0$ with $m_{\mathrm{fwd}} + m_{\mathrm{bwd}} = m$) acting independently on the forward and backward label tensors. Both inherit Proposition 2's convex-combination admissibility argument. The two settings differ in matched-memory accounting:

Setting	ALT memory	Matched-budget rule at B B/v
Undirected ($d_{\text{in}} \equiv d_{\text{out}}$)	K floats/vertex	AAC $m{=}B/4$ vs. ALT $K{=}B/4$
Directed ($d_{\text{in}} \neq d_{\text{out}}$)	$2K$ floats/vertex	AAC $m{=}B/4$ vs. ALT $K{=}B/8$

The covering-radius bound (Theorem 5) uses the symmetrized form $r_m^{\text{sym}} := \max_{v \in V_{\text{reach}}} \min_{l \in S} \max(d(l,v), d(v,l))$ on directed graphs and the standard r_m on undirected graphs; the full derivation and admissibility proof for the directed case are in Appendix A.

3.3 Theoretical Analysis: Capacity Ceiling and Covering Radius

We split the framework into two parts. Section 3.3 states the architectural admissibility guarantee (the capacity ceiling that bounds any row-stochastic compressor by its teacher), needed to interpret the matched-memory comparisons. Section 3.3 states the covering-radius and pool-size bounds that explain *why* the comparison comes out the way it does, needed to interpret the per-graph-type results.

Architectural admissibility. We now state the admissibility guarantee as a direct consequence of a classical observation from the heuristic-search literature (Pearl, 1984, Chap. 4): convex combinations of admissible heuristics are admissible. The logical chain is:

Triangle inequality ⇒ ALT admissibility (Thm. 1) ⇒ Compression preserves admissibility (Thm. 2, corollary of Pearl (1984)) ⇒ Covering radius controls gap (Thm. 5).

Proposition 2 (Row-stochastic compression preserves admissibility; corollary of Pearl 1984, Chap. 4)**.** *Let $A \in \mathbb{R}^{m \times K_0}$ be any row-stochastic matrix (non-negative entries, each row summing to 1). Then:*

(a) ***Undirected case:*** $h_A(u,t) = \max_i |y_i(u) - y_i(t)| \leq \max_k |d(l_k,u) - d(l_k,t)| = h_{\text{ALT}}(u,t) \leq d(u,t)$.

(b) ***Directed case:*** *Let $A_{\text{fwd}} \in \mathbb{R}^{m_{\text{fwd}} \times K_0}$ and $A_{\text{bwd}} \in \mathbb{R}^{m_{\text{bwd}} \times K_0}$ be row-stochastic. Then $h_A(u,t) \leq h_{\text{ALT}}(u,t) \leq d(u,t)$, where h_A is defined by Eq.* (4).

Sketch. Each compressed dimension is a convex combination of admissible teacher differences (each row of A, A_{fwd}, A_{bwd} is a probability vector), so the row-wise inner products are bounded by the max over teacher landmarks of each bound type; taking the max over compressed rows preserves the inequality, and ALT admissibility (Theorem 1) closes the chain to $d(u,t)$. Full proof in Appendix A; the inequality itself is Pearl (1984)'s pointwise-max observation, applied here to each ALT bound type. The empirical consequence is that the Gumbel-softmax training trajectory stays admissible without projection (Section 3.4). □

Corollary 3 (Inference admissibility)**.** *At inference, hard argmax selection produces one-hot rows $A_i = e_{j^*(i)}$, which are row-stochastic. Therefore Proposition 2 applies and the deployed heuristic is admissible.*

Proposition 4 (ALT as a special case of AAC)**.** ***Undirected case:*** *If $m = K_0$ and $A = I_{K_0}$, then $h_A \equiv h_{\text{ALT}}$.* ***Directed case:*** *If $m_{\text{fwd}} = m_{\text{bwd}} = K_0$ (total storage $m = 2K_0$) and $A_{\text{fwd}} = A_{\text{bwd}} = I_{K_0}$, then $h_A \equiv h_{\text{ALT}}$. The $m = 2K_0$ case is the non-compressed regime that recovers ALT exactly; actual compression requires $m < 2K_0$ for directed graphs (resp. $m < K_0$ for undirected). More generally, if each row of A (resp. A_{fwd}, A_{bwd}) is a standard basis vector $e_{j^*(i)}$, then h_A equals the ALT heuristic restricted to the selected landmark subset (see Figure 2).*

Proof. With $A = I$, $y_i(v) = \sum_k (I)_{i,k}\, d(l_k, v) = d(l_i, v)$, so the compressed labels equal the teacher labels. The heuristic definitions (Eq. 4 for directed, L_∞ for undirected) then coincide with Eq. (1). For directed graphs, setting $A_{\text{fwd}} = A_{\text{bwd}} = I_{K_0}$ recovers all K_0 forward and K_0 backward bounds. With one-hot rows $A_i = e_{j^*(i)}$, each compressed dimension selects exactly one teacher landmark, restricting the maximization to the selected subset. □

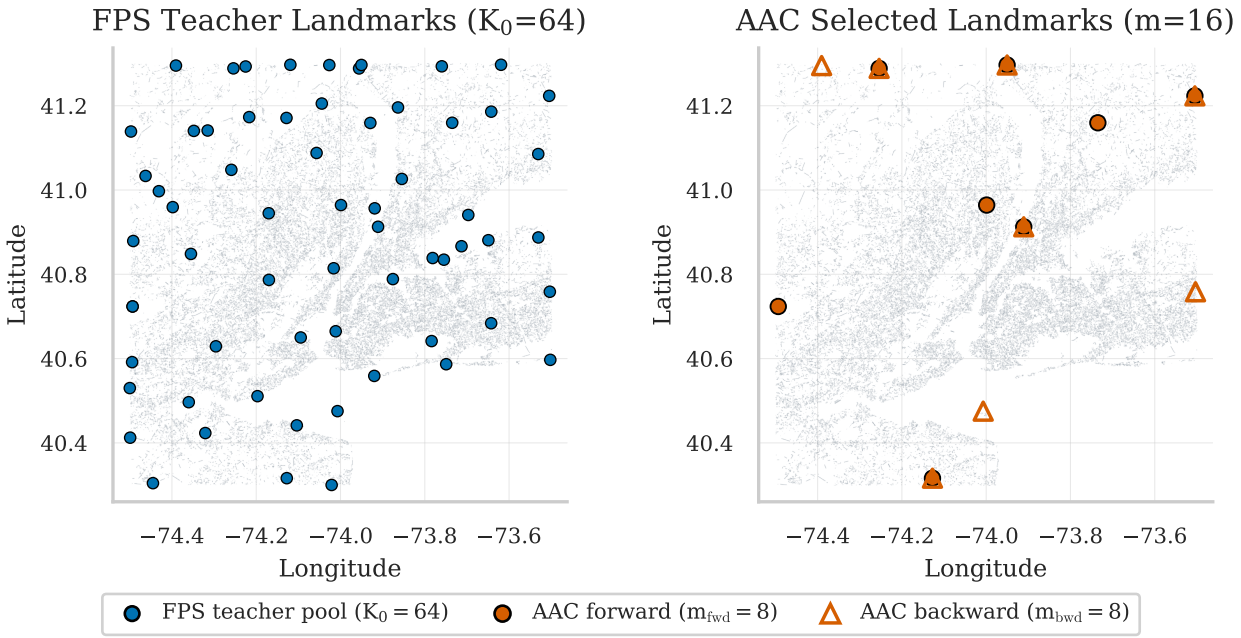

Figure 2: Landmark selection on NY (DIMACS). **Left**: K_0=64 FPS teacher landmarks, dispersed across the graph. **Right**: AAC's m=16 directional selection (8 forward + 8 backward; 11 distinct vertices, 5 shared across directions), boundary-concentrated rather than dispersed. Figure 3 shows the SBM analogue. Per-graph-type matched-memory results in Table 14.

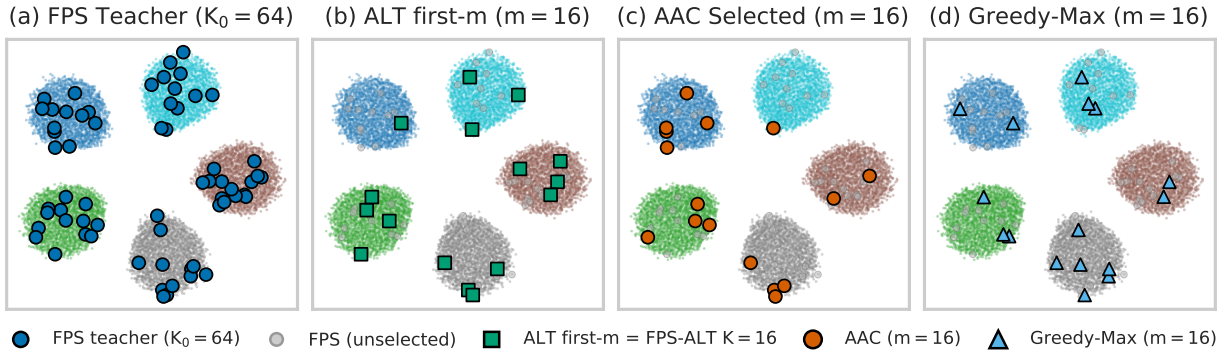

Figure 3: Landmark selection on SBM (5×2000, p_{in}=0.05, p_{out}=0.001; spring layout, color = block). Four selection rules from the same K_0=64 FPS teacher pool, m=16. **(a)**: FPS teacher (all 64). **(b)**: ALT first-m – the actual matched-memory baseline, algebraically equal to FPS-ALT K=m via the forced-first-m identity (Section 5.9). **(c)**: AAC learned selection. **(d)**: Greedy-Max coverage oracle on the same pool. Under per-graph-type matched-memory accounting (Section 5.2), FPS-ALT leads AAC by 0.12–1.34 % on SBM (Section 5.8.1); the figure documents a qualitative placement difference, not a downstream advantage.

Covering-radius and pool-size bounds. Section 3.3 answered "how loose can the compressor be relative to its teacher?" (capacity ceiling: it cannot exceed the teacher). This subsection answers "how loose can the teacher be relative to the true distance?" (covering-radius bound) and "how does the matched-memory question interact with K_0?" (pool-size bound). Together with the gradient-identity scope-note above (Section 3.4), they give the full theoretical picture: the architecture admits the FPS-ALT identity subset (Proposition 4), the teacher gap is $\leq 2r_m^{\text{sym}}$ (Theorem 5), and the gap-to-teacher gradient is not guaranteed to steer toward the covering-optimal subset; jointly, this explains the matched-memory parity finding without invoking any failure of the architecture.

The boundary placement visible in Figure 2 is consistent with a classical observation by Goldberg & Harrelson (2005): landmarks at the periphery produce tight ALT bounds for long-range queries. AAC's learned selection concentrates landmarks in a smaller extremal cluster, trading uniform spatial coverage for stronger bounds on a subset of query directions. On road networks this tradeoff is unfavorable: the covering radius increases (Theorem 5), degrading average-case performance. The contrast is clearest in Figure 3: the trained selector leaves one community with a single landmark and concentrates the budget on the other four, but this re-allocation does not translate into a measurable advantage over FPS at matched memory (Section 5.8.1).

Covering radius bound. The ALT heuristic on a landmark subset S cannot be tighter than the full set, but how much looser can it be? The covering radius (Section 2.2) quantifies this:

Theorem 5 (Covering radius bound)**.** *Let $S \subseteq L$ be any m-element landmark subset with covering radius $r_m := \max_{v \in V} \min_{l \in S} d(v, l)$. Then for all $u, t \in V$ with finite $d(u, t)$:*

$$d(u,t) - h^S_{\mathrm{ALT}}(u,t) \;\le\; 2\, r_m, \tag{5}$$

where h^S_{ALT} denotes the ALT heuristic restricted to landmarks S. Since AAC at inference selects a fixed subset S of teacher landmarks (Corollary 3), we have $d(u,t) - h_{\mathrm{AAC}}(u,t) \le 2\, r_m$.

For directed graphs under the reachability assumption (Section 2), the bound holds with the symmetrized covering radius $r^{\mathrm{sym}}_m := \max_{v \in V_{\mathrm{reach}}} \min_{l \in S} \max\big(d(l,v),\, d(v,l)\big)$, where V_{reach} is the set of vertices reachable from at least one landmark in both directions.

Proof sketch. Fix any u, t with finite $d(u,t)$. Let $l^* \in S$ be the landmark minimizing $d(u, l^*)$; by the covering radius definition, $d(u, l^*) \le r_m$. The forward ALT bound for l^* gives $d(l^*, t) - d(l^*, u) \le d(u,t)$ (Theorem 1), so:

$$d(u,t) - \big[d(l^*,t) - d(l^*,u)\big] = d(u,t) + d(l^*,u) - d(l^*,t) \le d(u,l^*) + d(l^*,u) \le 2\, d(u,l^*) \le 2\, r_m,$$

where the key step uses $d(l^*, t) \ge d(u,t) - d(u, l^*)$ (triangle inequality). Since $h^S_{\mathrm{ALT}} \ge d(l^*,t) - d(l^*,u)$, the result follows. Full proof including the directed case in Appendix A. □

Corollary 6 (FPS landmark selection)**.** *If S is selected by farthest-point sampling (Gonzalez, 1985), then $r_m \le 2\, r^*_m$, where r^*_m is the optimal m-center covering radius. Consequently, $d(u,t) - h^S_{\mathrm{ALT}}(u,t) \le 4\, r^*_m$.*

3.4 Training Objective

We minimize the expected gap between the teacher heuristic and the compressed heuristic, with one row-entropy regularizer R_{ent} encouraging sharp (low-temperature) selection:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,t)}\Big[\big(h_{\mathrm{ALT}}(s,t) - h_A(s,t;\theta)\big)_+\Big] + \lambda_{\mathrm{cond}} \cdot R_{\mathrm{ent}}(\theta), \tag{6}$$

where $(\cdot)_+ = \max(0, \cdot)$ and θ denotes the compressor parameters. We additionally considered a uniqueness regularizer $\lambda_{\mathrm{uniq}} \cdot R_{\mathrm{uniq}}(\theta)$ penalizing duplicate landmark selections; the $\lambda_{\mathrm{uniq}} \in \{0, 0.01, 0.1\}$ sweep in Appendix C shows the temperature schedule and R_{ent} alone yield effective unique-ratio 1.0 at every cell, so we do not include this term in the reported runs; it remains available in the released training code for tighter m/K_0 ratios where mode collapse may bind. The hard max in Eq. (4) admits an admissibility-preserving smooth surrogate M_T (log-sum-exp minus $\frac{\log m}{T}$); the static experiments do not use it (they train with straight-through Gumbel-softmax and the hard max), so we defer the full statement to Appendix A (Theorem 8), where it is needed for the contextual variant.

Proposition 7 (Gradient identity for the gap-to-teacher objective)**.** *Let θ denote the compressor parameters. Under exact admissibility (Proposition 2), the gap-to-distance and gap-to-teacher objectives are identical up to a θ-independent constant:*

$$\mathbb{E}\big[(d(s,t) - h_A(s,t;\theta))_+\big] = \mathbb{E}\big[(h_{\mathrm{ALT}}(s,t) - h_A(s,t;\theta))_+\big] + \mathbb{E}\big[d(s,t) - h_{\mathrm{ALT}}(s,t)\big].$$

In particular, $\nabla_\theta \mathbb{E}\big[(d(s,t) - h_A(s,t;\theta))_+\big] = \nabla_\theta \mathbb{E}\big[(h_{\mathrm{ALT}}(s,t) - h_A(s,t;\theta))_+\big]$.[2]

Proof. By Proposition 2, $h_A(s,t;\theta) \le h_{\mathrm{ALT}}(s,t) \le d(s,t)$ for every (s,t). Therefore both gaps are non-negative:

$$d(s,t) - h_A \ge 0 \quad \text{and} \quad h_{\mathrm{ALT}}(s,t) - h_A \ge 0,$$

so the positive-part operators are inactive: $(d - h_A)_+ = d - h_A$ and $(h_{\mathrm{ALT}} - h_A)_+ = h_{\mathrm{ALT}} - h_A$. Expanding: $d - h_A = (h_{\mathrm{ALT}} - h_A) + (d - h_{\mathrm{ALT}})$. Taking expectations, $\mathbb{E}[d - h_{\mathrm{ALT}}]$ is independent of θ (both are graph properties or frozen teacher labels). The gradient identity follows immediately. □

This allows training with h_{ALT} (available from teacher labels at cost $O(K_0 V)$ per direction) instead of d (requiring full $O(V^2)$ all-pairs supervision), over sampled source-target queries. Proposition 7 shows the two objectives share gradients but does not imply either minimum coincides with the covering-radius-optimal subset; Section 5.9 documents this gap empirically.

Toy example: 1D path P_7. A closed-form construction makes the gap-vs-coverage divergence concrete. Take $G = P_7$ (vertices $\{0,\dots,6\}$, unit edges), landmark budget m=2, and the uniform query distribution Q on $\{1,\dots,5\}^2 \setminus \{s{=}t\}$ (20 ordered pairs). Compare the symmetric k-center optimum S_{cov}=$\{2,4\}$ against the symmetric peripheral subset S_{gap}=$\{0,6\}$ (Figure 4). On a 1D path, $|d(s,l) - d(l,t)| = d(s,t)$ whenever l lies outside the interval $[\min(s,t), \max(s,t)]$, so S_{gap} has at least one landmark strictly outside every interior query interval and the ALT bound is exact for all 20 queries: $\mathbb{E}_Q[d - h^S_{\mathrm{ALT}}] = 0$. The covering optimum S_{cov} has $r_2 = 2$ (against $r_2 = 3$ for S_{gap}) but is straddled by the longest queries $(1,5)$ and $(5,1)$, contributing $\mathbb{E}_Q[d - h^S_{\mathrm{ALT}}] = 1/5 > 0$. The gap-to-teacher minimum on this graph is therefore the covering-radius worst case among the two candidates, and adding a soft covering-radius penalty λ_{cov} to the loss interpolates monotonically from S_{gap} (λ_{cov}=0) to S_{cov} (large λ_{cov}): exactly the tradeoff documented empirically in Appendix B.[3]

Contextual variant. For settings where edge costs depend on context, the compression module integrates into an end-to-end differentiable pipeline; deployment-time admissibility holds w.r.t. the predicted-cost graph (Proposition 2). The pipeline specification and a Warcraft compatibility check are in Appendix F; the rest of the paper concerns the static setting.

4 Related Work

We organize related work into four groups: exact route-planning engines, learned heuristics without formal guarantees, admissibility-aware learning, and theoretical foundations for feature-based heuristics.

Exact route-planning engines (CH, CRP, CCH, CH-Potentials, EHL*). Contraction hierarchies (CH) (Geisberger et al., 2008), hub labeling (Cohen et al., 2003; Abraham et al., 2012), CRP (Delling et al., 2011), and CCH (Dibbelt et al., 2016) achieve orders-of-magnitude speedup on static (and dynamic, via metric customization) road networks by exploiting hierarchical structure; the survey of Bast et al. (2016) reports continental-scale queries in milliseconds. CH-Potentials (Strasser & Zeitz, 2021) extracts an admissible A* potential from a contraction hierarchy and is arguably the most direct exact-A*-heuristic comparator on static road networks; we scope our claims outside the static-oracle regime it dominates (Section 5, scoping paragraph). We position against ALT rather than CH-Potentials because our target regime is learning from demonstrations, instance-dependent edge costs, or streaming and partially observed graphs, where full-graph static preprocessing is unavailable or must be repeated per instance, and where differentiability of

[2]The identity relies on the exact admissibility chain $h_A \le h_{\mathrm{ALT}} \le d$, which holds in the static setting (Section 3.1, exact Dijkstra teacher labels). It does *not* extend to the contextual variant: there the smooth-min Bellman–Ford proxy does not preserve a lower bound on true distances during training, and the contextual case relies on deployment-time admissibility via the hard-argmax reduction to ALT on the predicted-cost graph (Appendix F).

[3]The exactness $\mathbb{E}[\mathrm{gap}] = 0$ for S_{gap} is a 1D artifact (every interior query has a landmark on one side); on general graphs the analogous direction is a tendency rather than an equality, consistent with the classical observation by Goldberg & Harrelson (2005) that peripheral landmarks produce tight ALT bounds for long-range queries.

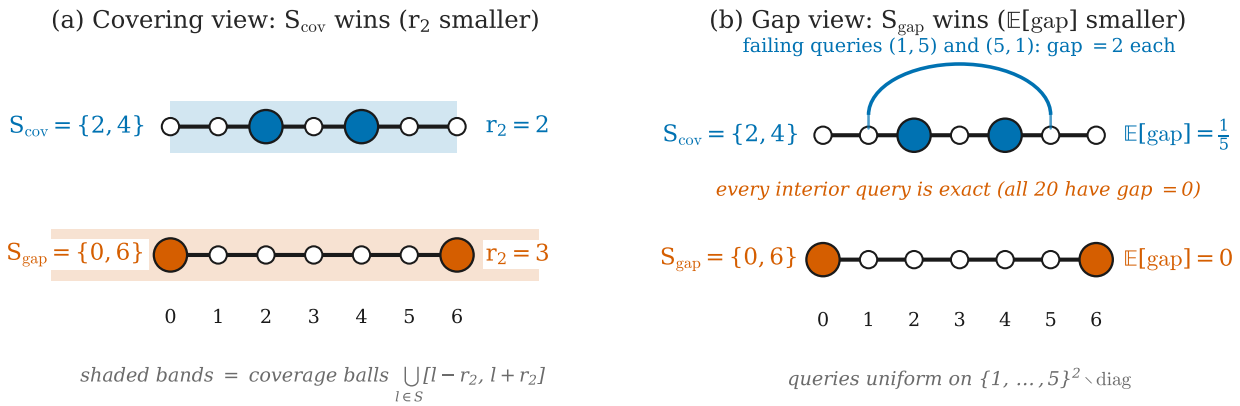

Figure 4: **Gap-to-teacher and covering radius diverge on the toy path P_7.** The same two candidate m=2 landmark selections appear in both panels (top row: S_{cov}=$\{2, 4\}$; bottom row: S_{gap}=$\{0, 6\}$); queries are drawn uniformly from $\{1, \ldots, 5\}^2 \setminus$ diag (20 ordered pairs). *(a) Covering view.* Shaded bands are coverage balls $\bigcup_{l \in S}[l - r_2,\, l + r_2]$. The symmetric k-center subset S_{cov} wins on r_2 (=2 vs. =3 for S_{gap}). *(b) Gap view.* The arc above S_{cov} traces the symmetric pair of queries $(1, 5)$ and $(5, 1)$ that the heuristic fails on (gap = 2 each, contributing $\mathbb{E}_Q[\text{gap}] = 4/20 = 1/5$); every other query is exact for both selections. S_{gap} has no failing queries because on a 1D path any landmark lying strictly outside $[\min(s, t), \max(s, t)]$ gives $|d(s, l) - d(l, t)| = d(s, t)$ exactly, and the peripheral pair $\{0, 6\}$ leaves at least one such landmark for every interior query. The two panels are perfectly contrasted: blue (covering-radius-aligned) wins (a); vermillion (gap-to-teacher-aligned) wins (b).

the heuristic through to predicted edge costs (Section 3.4) is required. On static-oracle benchmarks CH-Potentials remains the appropriate tool; the construction we study here is complementary, not a replacement. EHL* (Du et al., 2026) is a methodologically analogous *memory-budgeted* hub-labeling design that explicitly trades label memory against query runtime, the closest published precedent to our matched-memory diagnostic posture, although in the distance-oracle rather than ALT-heuristic setting.

2-hop and landmark hub labeling (PLL, LHL). The closest scalable predecessor to landmark hub labeling is the *pruned landmark labeling* (PLL) of Akiba et al. (2013): a per-vertex 2-hop labeling that answers exact distance queries by intersection of source and target label sets, with pruning during BFS to keep label sizes small. *Landmark hub labeling* (LHL) (Storandt, 2022; 2024; Cauvi et al., 2024) adds landmarks as anchors to the same 2-hop tradition; LHL achieves exact distance queries using ∼40–80 labels per vertex on road networks, with heavier combinatorial preprocessing (NP-hard for general graphs (Coste et al., 2025)). LHL invests heavier preprocessing for exact distance answers; AAC accepts a bounded heuristic gap (Theorem 5) for lightweight gradient-based training and differentiable-pipeline compatibility. We treat LHL as neighboring route-planning work, not a like-for-like matched-memory baseline, because the query model differs (oracle vs. ALT-family heuristic; see Figure 5 for the schematic placement of AAC relative to admissible A*-heuristic methods on the left panel and distance oracles on the right).

Learned heuristics without exact guarantees (PHIL, Neural A*, iA*, UPath, blackbox differentiation, DataSP). PHIL (Pandy et al., 2022) learns heuristics via imitation on expert trajectories; Neural A* (Yonetani et al., 2021) differentiates through A* on grid maps; recent grid/path-planning systems such as iA* (Chen et al., 2025) and UPath (Ananikian et al., 2026) improve search efficiency or out-of-distribution generalization. None guarantees admissibility. On the differentiable side, blackbox differentiation (Vlastelica et al., 2020) solves Dijkstra in the forward pass, DataSP (Lahoud et al., 2024) provides differentiable Floyd-Warshall at $O(V^3)$, and CombOptNet (Paulus et al., 2021) offers a generic differentiable integer-programming layer. AAC achieves $O(m)$ heuristic lookup with admissibility guaranteed by architecture rather than by solving the full combinatorial problem.

Differential heuristics, CDH, and FastMap (admissible compression baselines). Differential heuristics (Sturtevant et al., 2009) store distance differences from pivot vertices; on undirected graphs the max-of-$|d(a,p)-d(b,p)|$ formulation coincides with ALT and is admissible by the same triangle-inequality argument. Compressed differential heuristics (CDH) (Goldenberg et al., 2011; 2017) retain, per vertex, only the $r<P$ most informative pivot distances plus their indices from a P-pivot pool; CDH and AAC target the same object (an admissible landmark-based heuristic under a bytes-per-vertex budget) but differ in the compression operator: CDH selects a hard subset offline, AAC parameterizes it as a row-stochastic Gumbel-softmax layer trained by gradient descent. We do not claim AAC is uniquely admissible in this space (CDH is too) but only that it is the differentiable parameterization; a reference CDH implementation is included in the released artifact (Section 5.8.4). FastMap (Faloutsos & Lin, 1995; Cohen et al., 2018) computes Euclidean embeddings via iterative farthest-pair projection; we use L1 distance as the heuristic since the L2 variant is inadmissible (Cohen et al., 2018). FastMap's admissibility guarantee is undirected; our directed-graph experiments (Section 5.3.1) are a cautionary out-of-domain example.

Admissibility-aware learning. Futuhi & Sturtevant (2026) organize prior work on admissibility-aware learning into three lanes: *adapt learning to meet search requirements* (e.g., training-loss penalties or rejection sampling that approximately preserve admissibility), *adapt search to ML heuristics* (e.g., weighted-A* (wWA*) and anytime weighted-A*-style search that is robust to inadmissible h), and *abandon admissibility entirely*. We extend their three-way split with a fourth lane that the original taxonomy implicitly encompasses but does not name: *constrain the hypothesis class to be closed under admissibility*, so neither the loss nor the search has to compensate for a violation. In short, lanes 1–3 leave the model class unrestricted and use loss, search, or both to recover from violations; lane 4 picks a class that cannot violate. The four lanes, with AAC sitting in lane 4, are:

- *Loss-side adaptation (lane 1).* Nunez-Molina et al. (2024) model the learned heuristic as a truncated Gaussian lower-bounded by an admissible classical heuristic (a guarantee in expectation under the fitted likelihood, not pointwise per query), and Futuhi & Sturtevant (2026) propose a CEA loss that achieves $\sim 10^{-6}$ violation rates on puzzle domains with PAC generalization bounds for unrestricted hypothesis classes; Hadar et al. (2026) extend the line with multi-step Bellman updates whose guarantees remain optimization-dependent.
- *Search-side adaptation (lane 2).* Most of the learned-distance-oracle literature falls here implicitly: Ananikian et al. (2026)'s UPath, Chen et al. (2025)'s iA*, and TransPath (Kirilenko et al., 2023) learn powerful inadmissible heuristics and rely on the search wrapper (often weighted-A* style) to recover near-optimal cost ratios.
- *Abandoning admissibility entirely (lane 3).* A complementary line of work treats admissibility as a constraint to be dropped in exchange for raw search efficiency or out-of-distribution generalization, and reports cost-ratio rather than optimality (PHIL (Pandy et al., 2022), Neural A* (Yonetani et al., 2021), and the broader learned-distance-oracle literature surveyed by Choudhary et al. (2026) sit here when they are deployed as direct heuristic surrogates without an admissible safety wrapper).
- *Hypothesis-class restriction (lane 4, the lane AAC occupies).* Admissibility-preserving *compression* of lookup-table heuristics has a long classical history that motivates this lane: pattern databases (Culberson & Schaeffer, 1998) abstract away state variables and take a min over abstract distances, the *compressed* pattern databases of Felner et al. (2007) additionally compress the lookup table while preserving admissibility, and merge-and-shrink abstractions (Helmert et al., 2014) build factored abstractions whose composition preserves admissibility. AAC is the differentiable analog of this compress-while-preserving-admissibility tradition for the specific ALT triangle-inequality family; the structural guarantee in our paper is the same idea (stay inside a class that is closed under the compression operator) implemented via row-stochastic convex combination rather than variable abstraction or value-table compression. The mechanisms are complementary across lanes: CEA gives loss-enforced approximate admissibility over unrestricted hypothesis classes ($\sim 10^{-6}$ violation rate, PAC bound); AAC gives zero-violation architectural guarantees for a structurally restricted (ALT-family) class.

The closest prior work on *admissible neural heuristics* is NEAR (Shah et al., 2020), which learns neural relaxations of partial programs as admissible A* heuristics for differentiable program synthesis. NEAR

achieves ε-admissibility via a generalization argument over the learned relaxation, whereas AAC achieves *exact* admissibility by an architectural constraint (the row-stochastic max-of-admissibles construction of Proposition 2); the domains also differ (program-architecture graphs vs. ALT shortest-path heuristics on graphs). These lines of work address different domains with different hypothesis classes and different guarantee mechanisms. Within the admissible landmark-based space, CDH (Goldenberg et al., 2011; 2017) is architecturally admissible via offline per-vertex pivot-subset storage with bound substitution; AAC adds a *differentiable* parameterization of the same compress-while-preserving-admissibility step, which is what enables end-to-end composition with learned-cost pipelines (Section 3.4).

Learned distance-oracle estimators. A parallel line of work learns *distance estimators* for point-to-point shortest paths on road networks: ndist2vec and vdist2vec-family node-embedding methods, hierarchical-embedding approaches (Choudhary et al., 2026), and the broader "learned distance oracle" literature surveyed therein. These methods solve a related but different problem (approximate distance estimation, not admissible heuristic computation) and consequently return suboptimal paths in general, even at high expansion reduction. We focus on admissible heuristics and do not run a direct head-to-head on expansion reduction alone; a comparison is possible but requires the solution-quality caveat of Appendix D (FastMap), where we quantify the cost-ratio penalty of giving up admissibility.

Theory of learned heuristics. Eden et al. (2022) formalize feature-based heuristics for A* search, proving dimension and label-length tradeoffs for embeddings that preserve admissibility. Concretely, their main lower-bound result establishes that for any ℓ-label admissible scheme on a graph with n vertices the worst-case heuristic gap can be as large as $\Omega(\mathrm{diam}(G)/\ell)$ (informally), and the matching upper bound is a labeling whose per-vertex storage scales with ℓ. Our Theorem 5 is the constructive instantiation of this label-length-vs-gap tradeoff inside the ALT family: choosing m compressed dimensions (an $\ell{=}m$-label scheme that stores landmark-distance differences) yields a heuristic gap bounded by $2r_m^{\mathrm{sym}}$, where r_m depends on the graph through its m-center covering radius. The ALT-family instantiation makes both the upper bound (Gonzalez 2-approximation, Corollary 6) and the lower bound (covering radius cannot drop below the optimal r_m^*) explicit and computable per graph; the Eden, Indyk, and Xu framework gives the model-agnostic existence statement that this kind of tradeoff is unavoidable. Sakaue & Oki (2022) derive sample-complexity bounds for learning heuristic functions used by greedy best-first search (GBFS) and A*, establishing when finite data suffices to approximate a good heuristic. Rayner et al. (2013) formulate *learned heuristic-subset selection* as a submodular-coverage problem and give a greedy classical algorithm with approximation guarantees; AAC can be read as a continuous Gumbel-softmax relaxation of the same combinatorial selection problem, with the relaxation chosen so that gradients pass through to an outer pipeline (Section 3.4). The fact that our continuous relaxation matches but does not exceed FPS at matched memory on road networks (where Rayner et al. (2013)'s greedy classical selector also yields only marginal gains over uniform pivot choice) is consistent with the broader picture that on graphs with near-optimal FPS coverage (Corollary 6), the residual headroom for any selection algorithm, classical or learned, is small. These theoretical and algorithmic results frame our contribution as a constructive instantiation: AAC provides a concrete differentiable architecture within the "admissible-by-design" program that all three lines of work motivate.

5 Experiments

The experiments evaluate AAC against FPS-ALT under matched per-vertex deployed-label memory across four graph classes, addressing three questions: (1) How close does a differentiable admissible selector come to FPS's covering-radius ceiling on road networks? (2) Does graph structure (non-metric, community, power-law) open residual headroom? (3) What is the binding constraint on the gap? The consistent finding is that FPS-ALT is hard to beat at matched memory: the covering-radius bound (Theorem 5) is already tight on the tested graphs, and the residual gap is traced to training-objective drift, not architectural capacity (Section 5.9). A secondary compatibility question (can the compressor be inserted into a differentiable pipeline without surrendering admissibility?) is answered affirmatively in Appendix F.

Experimental roadmap. The empirical evaluation is organized as three complementary studies:

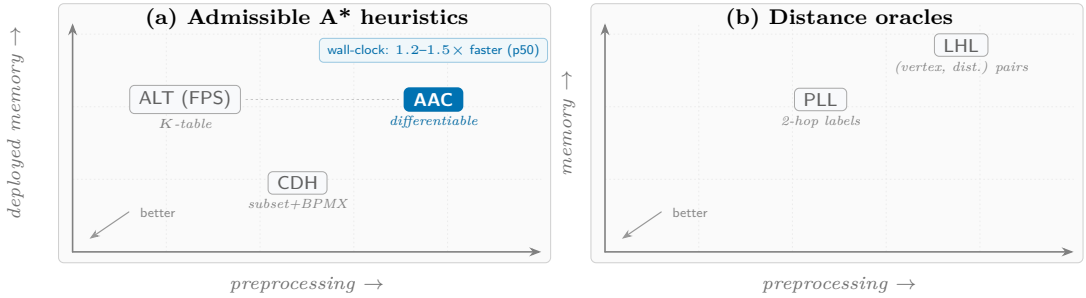

Figure 5: Preprocessing vs. deployed memory for admissible landmark-based methods; the *better* arrow marks the Pareto-favored corner *within* each panel. **(a)** A* heuristics: ALT and AAC sit at the same vertical level (matched B/v, dashed link); AAC's rightward offset is extra *offline* cost only (K_0 SSSPs + training). The boxed callout above AAC is not a plotted axis; it logs the orthogonal p50 wall-clock advantage at that matched configuration (1.24–1.51× on DIMACS, Table 3). **(b)** Distance oracles answer $d(u, t)$ directly; memory units are not comparable across panels.

1. **Static road networks** (Sections 5.3–5.4): matched-memory comparison of AAC vs FPS-ALT on 4 DIMACS and 5 OSMnx graphs (4.6K–4.5M nodes). This is the primary regime where FPS coverage is strong; the headline finding is a small but statistically robust FPS-ALT lead on road networks.

2. **Non-road graphs** (Section 5.8): synthetic stochastic block model (SBM, community structure) and Barabási–Albert (BA, power-law) graphs at 10K nodes plus the real Open Graph Benchmark arXiv (OGB-arXiv) citation network at ∼170K nodes. Here the story splits by graph family: ALT still leads on SBM/BA, while the only crossover appears on OGB-arXiv at the larger budgets; this is also where the pre-registered prediction (Appendix E) was evaluated.

3. **Contextual differentiable planning** (Section 5.10, Appendix F): compatibility check on Warcraft terrain that demonstrates the compressor can be dropped into a differentiable pipeline without surrendering deployment-time admissibility. This is a compatibility check, not a primary benchmark.

Sections 5.5–5.7 provide hybrid $\max(h_{\mathrm{AAC}}, h_{\mathrm{ALT}})$ and ablation evidence that cuts across studies 1 and 2. Significance testing uses FDR-corrected paired Wilcoxon signed-rank tests on per-query expansion counts.

5.1 Methodological Protocol

Comparator scope. Our matched-memory diagnostic covers four graph classes (DIMACS and OSMnx road networks; weighted SBM; Barabási–Albert with edge costs; OGB-arXiv citation network) and one compatibility-check setting (Warcraft 12×12 grid-world, Section 5.10). Within each class we vary graph size, query distribution, and landmark-pool construction; we do not claim coverage of dynamic or streaming graphs, graphs with negative-weight cycles, or non-metric settings beyond OGB-arXiv. We do not run head-to-head matched-memory comparisons against truncated-Gaussian likelihood models (Nunez-Molina et al., 2024) or CEA-loss training (Futuhi & Sturtevant, 2026); the architectural-vs.-loss-side admissibility distinction in Section 4 is a typological argument, not a head-to-head benchmark, and is named as future work in Section 6.2. We do not claim AAC beats state-of-the-art static-routing engines (CH, CH-Potentials, hub labeling); we quantify the ALT-family ceiling at matched memory and identify when a differentiable admissible selector earns its slot.

We pre-registered our primary hypotheses and TOST (Lakens, 2017) equivalence bounds before running the main experiments, following the ML pre-registration precedent of the NeurIPS 2020 Pre-registration Workshop (Bertinetto et al., 2021); the verbatim prediction and audit trail are in Appendix E, with the timestamped pre-registration archived alongside the released artifact (Appendix G). Our equivalence margin δ=1 % is intentionally permissive given typical observed effect sizes (0.12–3.9 % across the matched-memory cells we report); "TOST accepts equivalence within δ" therefore states that the gap is below 1 %, not that

it is statistically indistinguishable from zero. We also report the FDR-corrected per-cell Wilcoxon p-values (Table 16) so that readers can interpret tightness without relying on the equivalence margin alone.

One pre-registered prediction (AAC beats FPS-ALT on OGB-arXiv at matched memory by +2–+6 % at B=32 B/v narrowing to +1–+3 % at B=128 B/v) was falsified; we report the failure in Section 5.8.3 and interpret it in light of Theorem 5.

We use a directionality-aware query-accounting convention that reconciles directed and undirected ALT variants under a single cost metric (Section 5.2, "Matched deployed label memory"; full statement in Appendix E, E.2).

We use two visual badges to label tables: MAIN for headline tables that source the abstract and Section 1 numbers, and ABLATION for detail and supporting tables.

5.2 Experimental Setup

Hardware. All experiments ran on a single machine: Intel Core Ultra 9 285K CPU, 128 GB RAM, NVIDIA GeForce RTX 5090 (32 GB VRAM). Software: Python 3.11, PyTorch 2.11.0+cu130.

Seeds and queries. All experiments use 5 seeds $\{42, 123, 456, 789, 1024\}$ unless otherwise noted in individual table captions. Error bars report ± one standard deviation across seeds. Source-target queries are sampled once (seed 42) and shared across all methods within each experiment. Seeds control different sources of randomness per method:

- **AAC**: FPS starting vertex for teacher landmark selection, and Gumbel-softmax training initialization;
- **ALT**: FPS starting vertex for landmark selection. On all tested directed graphs (DIMACS and OSMnx), ALT shows zero variance across seeds (Tables 14, 7), indicating that FPS converges to the same landmark set regardless of starting vertex on these graphs;
- **FastMap**: fully deterministic (fixed starting vertex); standard deviations are zero.

Datasets.

- **DIMACS** road networks (Demetrescu et al., 2009): NY (∼264K nodes), BAY (∼321K), COL (∼436K), FLA (∼1.07M). Directed, weighted.
- **OSMnx** road networks (Boeing, 2017): Modena (30K nodes), Manhattan (4.6K), Berlin (28K), Los Angeles (50K), Netherlands (4.5M). Directed.
- **Warcraft** 12×12 grid maps (Vlastelica et al., 2020): 10,000 train / 1,000 test. 144 nodes, 8-connected. RGB terrain images (96×96).

AAC training. Static compression: $K_0 \in \{32, 64, 128\}$, $m \in \{8, 16, 32, 64\}$. Adam optimizer, lr=1e-3, 200 epochs, batch size 256, $\lambda_{\text{cond}} = 0.01$, $\lambda_{\text{uniq}} = 0$ (held out per Section 3.4; the $\lambda_{\text{uniq}} \in \{0, 0.01, 0.1\}$ ablation in Appendix C confirms the three values yield identical (to two decimals) unique-ratio 1.0 at every cell), Gumbel-softmax temperature τ annealed exponentially from 1.0 to 0.1 (distinct from the contextual smooth-min β schedule of Appendix F). All distance computations in float64.

Baselines.

- **ALT**: $K \in \{4, 8, 16, 32\}$ landmarks, farthest-point sampling. Memory: $2K \times 4$ bytes/vertex.
- **FastMap** (Cohen et al., 2018): $d \in \{8, 16, 32, 64\}$ dimensions. L1 (Manhattan) heuristic in Euclidean embedding. Memory: $d \times 4$ bytes/vertex.
- **Dijkstra**: no heuristic (expansion baseline).

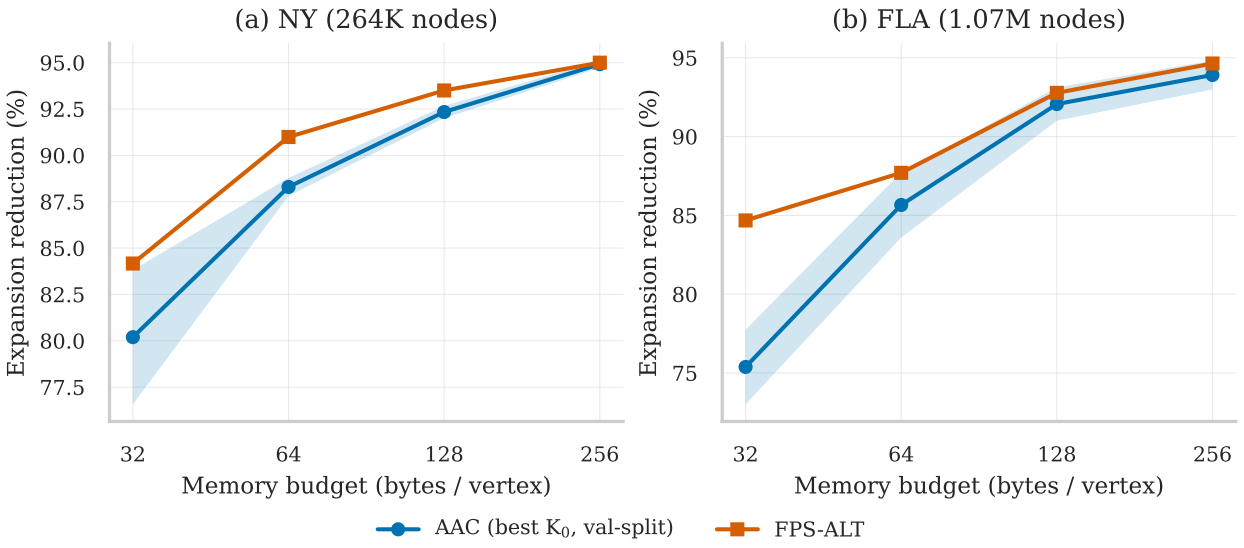

Figure 6: Pareto frontier of memory budget vs. expansion reduction on DIMACS road networks (admissible methods only). **(a)** NY (264K nodes); **(b)** FLA (1.07M). AAC tracks FPS-ALT within a few percentage points at matched memory on these highway-dominated graphs while remaining admissible by construction (Table 2, Proposition 2). Val-split means over 5 seeds with ±1 s.d. bands (MAIN protocol; canonical numbers in Table 6, retrospective upper envelope in Table 14). FastMap excluded as inadmissible (Section 5.3.1).

Matched deployed label memory. Throughout the paper, "matched memory" denotes the per-vertex bytes consumed by deployed landmark labels at query time. AAC stores m floats per vertex ($4m$ bytes in float32); for directed graphs the budget is split $\lfloor m/2 \rfloor$ forward and $\lceil m/2 \rceil$ backward. ALT stores $2K$ floats per vertex on directed graphs ($8K$ bytes; both forward and backward distances) and K floats on undirected graphs ($4K$ bytes). The matched-memory rule is therefore graph-dependent: at budget B bytes/vertex, **directed** graphs match AAC $m{=}B/4$ against ALT $K{=}B/8$, while **undirected** graphs match AAC $m{=}B/4$ against ALT $K{=}B/4$. Offline preprocessing time, training cost, and query latency are *not* matched; Table 13 reports those axes separately. All distance computations use float64; deployment labels are stored in float32 for the byte-per-vertex accounting. We observed no admissibility violations in any experiment.

Cross-table protocol note. The *val-split* convention uses 200 queries per seed, reserving the first 100 for choosing AAC's K_0 before reporting on the disjoint 100-query test split (Table 6). The *retrospective best-K_0* variant (Tables 14, 7) picks the best K_0 per cell after seeing all test queries; the ALT–AAC ordering is preserved under both conventions. Ablation tables retrain the compressor independently, so values for the same (K_0, m) can differ across tables by up to ∼3 % due to Gumbel-softmax stochasticity.

Seeds and query counts. Unless stated otherwise, every reported cell aggregates 5 seeds × 100 queries per seed (500 queries total per cell). Two graphs are reported with reduced replication where preprocessing cost makes 5 seeds prohibitive: **Netherlands** (OSMnx, ∼2.4M nodes, Table 7) uses 3 seeds × 100 queries, and the **Warcraft** compatibility study (Appendix F, 12×12 grids) uses 3 seeds with the standard differentiable-planning evaluation protocol of Vlastelica et al. (2020). All other tables in the paper use the 5-seed default.

5.3 DIMACS Road Networks: AAC vs. ALT vs. FastMap

Table 6 (Appendix B) is the per-cell DIMACS detail behind Table 2 under the MAIN val-split protocol. For each seed, 200 queries are split 100/100 for validation (AAC K_0 selection) and test reporting. At every matched budget, val-selected AAC generalizes with a mean val–test gap of +0.3 to +2.5 % and still underperforms ALT on test. The retrospective best-K_0 variant (Table 14) preserves the ordering. AAC incurs higher offline preprocessing (14–52 s vs. ALT 1.4–5.0 s); see Table 13 for the full cost breakdown.

Table 2: **MAIN** Headline matched-memory comparison on *expansion count* at B=64 B/v across all four graph classes: 9 road networks (4 DIMACS, 5 OSMnx), 2 synthetic non-road graphs (SBM, BA), and 1 real citation network (OGB-arXiv). Cells are mean expansion-reduction % over Dijkstra ($\pm$ std across 5 seeds; 3 seeds for Netherlands). FPS-ALT and AAC at matched landmark memory under the corrected per-graph-type accounting (Section 5.2); CDH cell is the best of `CDH+sub` and `CDH+sub+BPMX` where the reference benchmark exists (Table 4). TOST equivalence at δ=1 %: ✓ = accepted within δ, × = rejected (boundary), – = not applicable (DIMACS / OSMnx road graphs use the FDR-corrected per-cell Wilcoxon test of Table 16, Section 5.3, instead of TOST); see footnotes for details on the rejected SBM and OGB-arXiv cells. *Wall-clock query latency at the same matched-memory configuration goes the other way:* AAC is 1.24–1.51× faster than FPS-ALT at p50 on every DIMACS graph and at p95 on three of four (Table 3); see Section 5.3 for the cache-locality mechanism. Per-budget detail: roads in Appendix Tables 6, 7, 14; non-road in Table 9; full CDH benchmark in Table 4. "ALT std 0.0" reflects FPS determinism given a fixed LCC seed vertex.

Graph class	Graph (size)	AAC	FPS-ALT	CDH best	TOST δ=1 %
DIMACS road	NY (264K)	88.9 ± 1.2	**91.1**	47.1 %	–
	BAY (321K)	85.3 ± 4.3	**89.2**	—	–
	COL (436K)	85.6 ± 2.2	**86.5**	—	–
	FLA (1.07M)	86.2 ± 1.3	**87.6**	—	–
OSMnx road	Modena (30K)	87.7 ± 1.2	**91.2**	52.8 %	–
	Manhattan (4.6K)	88.1 ± 0.6	**90.4**	—	–
	Berlin (28K)	91.3 ± 0.7	**92.8**	—	–
	Los Angeles (50K)	87.5 ± 2.1	**91.1**	—	–
	Netherlands (4.5M)	87.3 ± 0.4	**91.2**	—	–
Synthetic	SBM 10K	94.02 ± 0.48	**94.52**	30.0 %	× [a]
	BA 10K	93.54 ± 0.19	**94.25**	—	✓
Citation	OGB-arXiv (169K)	**92.54 ± 0.54**	91.33	—	× [b, †]

[a] SBM TOST is rejected at the boundary (p_{lower}=0.052). [b] OGB-arXiv TOST is rejected; the pre-registered +2 to +6 % magnitude is falsified. [†] AAC cell is bolded only because AAC is numerically higher; it does *not* pass TOST equivalence at δ=1 % (Section 5.8.3).

Wilcoxon analysis. Per-cell paired Wilcoxon signed-rank tests (two-sided, 100 queries × 5 seeds per cell; Fisher combination across seeds, Benjamini–Hochberg FDR correction at q=0.05) show ALT achieves significantly fewer expansions than AAC at every one of 12 (graph, budget) cells, with 10/12 at combined $p < 10^{-3}$ and the two exceptions (COL-128, FLA-64) still significant after FDR. Full per-cell p-values and the storage-asymmetric companion (AAC at 64 vs. ALT at 128 B/v) are in Appendix Table 16. The mechanism (FPS captures highway-junction structure near-optimally, leaving little headroom for any selector) is explained by the covering-radius analysis (Theorem 5, Section 6).

Query latency. Table 3 reports p50/p95 query latency at matched deployed label memory (B=64 B/v: AAC K_0=64, m=16; ALT K=8, both giving the same $O(16)$-per-lookup heuristic-evaluation budget). At matched memory AAC is faster than ALT at both percentiles on every graph except BAY-p95, where ALT is 1.16× faster; the p50 advantage is 1.24–1.51× in AAC's favor and the p95 advantage is up to 1.75× on NY. Despite ALT's lower expansion count under matched memory (Table 16), AAC's compressed-label heuristic enjoys better cache locality (16 contiguous floats per direction vs. ALT's 8 floats from each of two directional tables), narrowing or reversing the per-query latency picture. Numbers are seed-42, 100 queries each, generated by our timing harness (described in Appendix G).

Full Pareto sweep. Table 15 in Appendix B shows all $K_0 \times m$ configurations. The best AAC configurations approach but do not surpass ALT at matched budgets on these graphs.

5.3.1 FastMap on Directed Graphs: A Methodological Note

FastMap (Cohen et al., 2018) achieves 93–97% expansion reduction on directed DIMACS graphs, but since its admissibility guarantee applies only to undirected graphs, **100% of FastMap-guided A* paths are**

Table 3: MAIN Median query latency (p50/p95 in ms) on DIMACS road networks at *matched deployed label memory* (B=64 B/v: AAC K_0=64, m=16; ALT K=8). Bold marks the per-(graph,percentile) winner. At matched memory AAC is faster than ALT on every cell except BAY-p95. Both use the same A* implementation.

Graph	Method	p50 (ms)	p95 (ms)
NY	AAC	**67.0**	**166.5**
	ALT	88.1	290.9
BAY	AAC	**105.6**	571.4
	ALT	136.0	**492.0**
COL	AAC	**132.4**	**549.9**
	ALT	200.0	814.8
FLA	AAC	**377.7**	**1210.5**
	ALT	468.4	1362.1

suboptimal on these benchmarks (mean cost ratio 1.15–1.22, max 1.95). This illustrates a methodological point: *expansion reduction alone is misleading when comparing across admissibility classes.* Full details in Appendix D.

5.4 OSMnx Directed Road Networks

On directed city-scale networks (Modena, Manhattan, Berlin, Los Angeles, Netherlands), ALT outperforms or matches AAC at nearly all tested memory budgets. Table 7 (Appendix B) summarizes the results across three budgets (32, 64, 128 B/v) and five graphs; the per-configuration sweep is in Table 8. At 64 B/v, ALT leads by 1.5–3.9 percentage points. The gap is wider at 32 B/v (2.7–7.0%) and narrows at 128 B/v (0.9–2.3%), with Manhattan at 128 B/v being the sole crossover (92.2% vs. 92.1%; a pool-size effect at $m \approx K_0/2$, see Section 3.3). The NY-DIMACS forced-first-m extension (Section 5.9, Appendix Table 19) confirms the training-drift mechanism on a directed road graph.

5.5 Hybrid $\max(h_{\mathsf{AAC}}, h_{\mathsf{ALT}})$ Evaluation

Hybrid terminology. We use two consistent labels throughout the paper, defined once here. **Matched-memory (primary):** a hybrid arm that consumes the same B bytes/vertex as either pure arm; for the half/half hybrid this means AAC at $B/2$ B/v and ALT at $B/2$ B/v combined by pointwise max. **Doubled-memory ($2B$, robustness check):** a hybrid arm that runs AAC at B B/v *and* ALT at B B/v in parallel and takes the pointwise max, consuming $2B$ bytes/vertex total. The matched-memory comparison is the scientifically primary question; the doubled-memory comparison speaks only to complementarity under a relaxed budget and lives in robustness tables.

Since both AAC and ALT produce admissible heuristics, their pointwise maximum $\max(h_{\mathrm{AAC}}, h_{\mathrm{ALT}})$ is also admissible and at least as informative as either component alone. Table 10 evaluates this hybrid on Modena, Manhattan, and NY at four budget tiers (32, 64, 96, 128 B/v), with a pure-ALT row at every tier to expose matched-budget comparisons.

ALT wins at matched total budget. At every matched budget level on all three graphs, pure ALT outperforms the AAC+ALT hybrid. For example, at 96 B/v: Modena ALT 92.5% vs. hybrid 89.0% (−3.5 %); Manhattan ALT 91.6% vs. hybrid 90.7% (−0.9 %); NY ALT 93.1% vs. hybrid 91.3% (−1.8 %). Adding AAC to ALT at matched budget does *not* beat spending the same bytes on more ALT landmarks. This is consistent with the covering-radius analysis (Theorem 5): on road networks FPS already captures most of the available landmark value, so the marginal bytes added as AAC yield less improvement than the same bytes added as additional FPS landmarks.

Complementarity is genuine at doubled memory. On non-road graphs (Section 5.8, Appendix E) the doubled-memory ($2B$) pointwise-max ensemble is the top performer, recovering +1 to +4 percentage points over pure ALT (Section 5.8.3). The matched-memory half/half hybrid is, however, dominated by both pure arms

on every non-road cell, so the complementarity does not translate into a free win under a strict memory constraint. The full per-budget road-graph hybrid sweep is in Table 10 (Appendix B).

5.6 Admissibility Verification Under Early Stopping

The core claim of AAC is that admissibility holds architecturally, given exact teacher labels, regardless of training quality. Table 11 tests this directly: AAC is trained to epoch checkpoints $\{1, 5, 10, 50, 200\}$ on Modena (K_0=64, m=16) and evaluated against Dijkstra-computed shortest distances. Zero admissibility violations are observed at *every* checkpoint across all 5 seeds, including the 1-epoch model, which already achieves 87.2% expansion reduction while remaining provably safe. Performance is stable from epoch 1 onward, confirming that the row-stochastic architecture provides immediate utility without extensive training.

Path-cost optimality. The admissibility check above bounds the pointwise heuristic $h(u, t) \leq d(u, t)$. On benchmarks whose per-run records log an "all-optimal" flag (the selection-strategy ablation, admissibility robustness sweep, coverage-aware regularization, and query-distribution experiments, totaling 213 runs across multiple graphs and seeds), a path-optimality audit confirms that A*-without-reopenings returned the Dijkstra-optimal path cost on every query of every run (zero path suboptimalities). In addition, the same audit checks 400 per-query path-cost records retained on all four DIMACS graphs (NY, FLA, BAY, COL; 100 queries each), and finds zero suboptimal AAC paths across all 400 queries. This addresses the directed-graph consistency caveat of Section 2.1: the sentinel-masked landmark set never excluded reachable terms in a way that compromised optimality. For the remaining DIMACS/OSMnx main tables, multi-seed Wilcoxon, and hybrid runs, per-run path costs were checked at run time but not persisted alongside the expansion numbers; we log zero admissibility violations on these and rely on Proposition 2's architectural chain rather than a post-hoc audit of persisted records.

Sensitivity to query distribution and amortized cost. The matched-memory ordering on *expansion count* is robust to query distribution (uniform / hotspot / powerlaw on NY and Manhattan; Table 12). On wall-clock cost the picture is more nuanced: AAC pays 7–31× in offline preprocessing (Table 13), but at matched memory its per-query latency is *lower* than ALT's at p50 on every DIMACS graph (Table 3), so the offline cost amortizes at modest workloads ($N_{\text{break}} \in \{170, 522, 1{,}276, 1{,}924\}$ queries on COL/FLA/BAY/NY; Figure 8). Above these breakevens AAC is the cheaper total-wall-clock choice; below them FPS-ALT is. The full distribution sweep (Table 12), the multi-axis cost breakdown (Table 13), and the amortized-cost curve (Figure 8) live in Appendix B.

5.7 Selection Strategy Ablation

To understand *why* ALT leads, we compare five selection strategies at matched memory (64 B/v, m=16) on Modena and Manhattan: AAC (default), Random-Subset, FPS-Subset (canonical ALT), FPS-RR$_{10}$ (random-restart FPS, validation-best of 10 seed vertices), and Greedy-Max (query-adaptive oracle on the K_0 pool); full numbers in Appendix Table 21. The ordering is Greedy-Max $\geq$ FPS-Subset $\gg$ AAC (default) $>$ Random-Subset on both graphs; random-restart FPS scores 1–2 % *below* canonical FPS, ruling out a "FPS is just lucky" explanation. Even Greedy-Max, a query-adaptive oracle with knowledge of the evaluation queries, beats FPS by only 0.3–1.1 %. The room for any selection algorithm to improve on FPS landmark quality is small. Duplicate collapse is not the bottleneck either: every AAC (default) model achieves effective unique-ratio 1.0 across all tested configurations, including tight budgets at m=4 from the compression-curve sweep (Table 22).

Covering radius interpretation. The FPS covering radii on road networks (symmetrized r_m^{sym} as defined in Appendix A, reported separately for forward and backward when they differ) are: Modena (K=16) fwd 26,386 m, bwd 27,524 m; NY (K=16) fwd 352,682 m. To contextualize these values: random query pairs on Modena have a mean shortest-path distance of approximately 160,000 m, so r_m^{sym} at K=16 is ∼16% of a typical query distance, already tight. On NY, r_m^{sym} at K=16 is ∼5% of a typical cross-city distance (∼7,000,000 m). FPS coverage at K=64 is tighter still: 13,113 m on Modena (fwd), 164,385 m on NY. These

values are within the Gonzalez 2-approximation guarantee (Corollary 6), confirming that FPS provides near-optimal spatial coverage on road networks. Full per-strategy numbers are in Appendix Table 21.

5.8 Non-Road Graphs: Matched-Memory Parity

Under the per-graph-type accounting of Section 5.2 (undirected graphs match m AAC floats against $K{=}m$ ALT floats, since ALT stores only one direction; directed graphs match m vs. $K{=}m/2$), pure FPS-ALT shows a small but consistent lead of 0.12–1.34 % in expansion reduction on weighted SBM and Barabási–Albert, and the two arms split on the pre-registered OGB-arXiv benchmark at magnitudes $\sim$5× smaller than the pre-registered bands. The four non-road subsections that follow cover one graph each (§5.8.1 SBM, §5.8.2 BA, §5.8.3 OGB-arXiv) plus the reference CDH baseline (§5.8.4); Section 5.9 adds a controlled training-objective drift diagnostic that explains *why* the learned compressor fails to improve on FPS at matched memory.

5.8.1 Stochastic Block Model (SBM, Weighted)

Setup. 10,000 nodes, 5 communities of 2,000 nodes each, $p_{\text{in}}{=}0.05$, $p_{\text{out}}{=}0.001$, edge weights uniform in $[1, 10]$ (undirected). Community structure creates clusters where FPS may waste landmarks on boundary vertices rather than distributing coverage within clusters. Under the matched-memory rule of Section 5.2, we evaluate at three budget levels (32, 64, and 128 B/v) with matched configurations $K_0{=}32, m{=}8$ vs. ALT $K{=}8$; $K_0{=}64, m{=}16$ vs. ALT $K{=}16$; $K_0{=}128, m{=}32$ vs. ALT $K{=}32$ (5 seeds, 100 queries per seed).

Result. Per-seed paired differences (AAC − ALT in expansion-reduction %) yield mean −1.26 % at $B{=}32$, −0.50 % at $B{=}64$, −0.12 % at $B{=}128$ (5 seeds). The paired TOST at $\delta{=}1$ %, $\alpha{=}0.05$ accepts equivalence within δ at $B{=}128$ only; at $B{=}32$ and $B{=}64$ the ALT lead exceeds the margin and the test rejects equivalence. The half/half hybrid arm is dominated at every budget; the doubled-memory pointwise-max ensemble $\max(h_{\text{AAC}}, h_{\text{ALT}})$ beats pure ALT by $\sim$0.95 % on SBM at $B{=}64$ B/v (and by $\sim$1.4 % over pure AAC) at 2× memory cost (Table 9). A query-adaptive Greedy-Max covering oracle (greedy selection from the $K_0{=}4m$ FPS pool that maximizes the average ALT heuristic over the evaluation queries) beats pure ALT by only +0.85 % at $B{=}32$, +0.52 % at $B{=}64$, +0.41 % at $B{=}128$: the headroom above FPS that any selection-based strategy could exploit is small.

5.8.2 Barabási–Albert (BA, with Edge Costs)

Setup. 10,000 nodes, preferential attachment with $m_{\text{attach}}{=}5$, edge weights uniform in $[1, 10]$ (undirected). Skewed degree distribution creates high-degree hubs where FPS gravitates toward the periphery while optimal selection should cover the hub-dominated shortest paths. Same matched-memory configurations as SBM.

Result. Per-seed mean differences −1.34 % at $B{=}32$, −0.71 % at $B{=}64$, −0.37 % at $B{=}128$. TOST accepts equivalence within $\delta{=}1$ % at $B{=}64$ and $B{=}128$ and rejects at $B{=}32$. The pattern mirrors SBM: ALT marginally ahead at the tightest budget, equivalent within 1 % at the larger budgets, with no cell where AAC leads. The half/half hybrid is dominated at every budget; the doubled-memory pointwise-max ensemble beats pure ALT by +0.6 % at $B{=}64$ at 2× memory cost. Greedy-Max sits *below* pure ALT at every budget on BA (−0.43, −0.04, −0.08 %); even a query-adaptive oracle from a 4×-larger FPS pool cannot beat the matched-memory FPS landmarks directly.

5.8.3 OGB-arXiv (Citation Network)

We pre-registered (before any OGB-arXiv evaluation; the verbatim prediction is reproduced in Appendix E): AAC beats FPS-ALT on the symmetrized 169,343-node OGB-arXiv citation network (Hu et al., 2020) at matched memory, with bands +2 to +6 % at $B{=}32$ B/v narrowing to +1 to +3 % at $B{=}128$ B/v. The observed matched-memory differences (5 seeds, 100 queries/seed) are −1.13 % at $B{=}32$ (direction inverted), $+1.21 \pm 0.54$ % at $B{=}64$, and $+0.87 \pm 0.20$ % at $B{=}128$: magnitudes $\sim$5× below the pre-registered bands at every budget, no cell achieving TOST equivalence within $\delta{=}1$ % at $\alpha{=}0.05$; the pre-registration is falsified. The AAC lead at $B{\geq}64$ is descriptive consistency-of-sign across seeds rather than a positive equivalence claim; effect magnitudes are within an order of magnitude of the seed-to-seed standard deviation, and we do not interpret OGB-arXiv at $B{\geq}64$ as a graph where AAC is demonstrably superior to FPS-ALT. This

is consistent with Theorem 5: OGB-arXiv's non-metric citation structure does place it in the regime where FPS's covering-radius bound is loose and a learned selector might gain ground, but the headroom turns out to be at most ∼1 % at the tested budgets, contradicting our initial expectation that the loose-bound regime would translate into a several-percentage-point matched-memory gap; the convention-independent zero-admissibility-violation prediction holds in 15/15 cells. On OGB-arXiv specifically the half/half hybrid is again dominated by both pure arms at matched memory, but Greedy-Max retains a modest +1.5 to +2.5 % advantage over pure ALT (Table 9), making OGB-arXiv the only non-road cell where pool-plus-query-adaptive selection demonstrably helps. The full audit trail is in Appendix E.

Cross-graph matched-memory hybrid summary. The matched-memory hybrid pointwise-max comparison (Table 9) compares pure AAC, pure ALT, half/half hybrid, doubled-memory $\max(h_{\mathrm{AAC}}, h_{\mathrm{ALT}})$, and Greedy-Max oracle across all three non-road graphs at three budget tiers. The half/half hybrid is dominated on *every* non-road cell: splitting the budget across two under-provisioned arms does not recover either arm's full-budget performance. The doubled-memory pointwise-max oracle exceeds each pure arm by 1.5–2.9 % on the cells where either arm alone is weakest, but requires running both methods at deployment so it is not a matched-memory arm.

5.8.4 Reference CDH Baseline at Matched Memory

The closest classical admissible neighbor of AAC in the landmark-compression family is the Compressed Differential Heuristic (CDH) of Goldenberg et al. (2011; 2017): a per-vertex *subset-storage* scheme on a P-pivot table that retains, per vertex, only the $r{<}P$ most informative entries plus the indices naming them. CDH and AAC are therefore the two natural "compress-while-staying-admissible-via-triangle-inequality" arms of the design space; we release a reference CDH implementation (top-r-farthest selection rule, $P{=}64$ pool) and benchmark it under exactly the matched-memory protocol of Sections 5.3–5.8 (5 seeds × 100 queries; matched bytes per vertex; same FPS pool as ALT). Three CDH arms are reported, in increasing tightness: (a) **CDH**, the strict intersection variant; (b) **CDH+sub**, the Goldenberg upper/lower bound-substitution mode (Section 4; uses a $P \times P$ pivot–pivot side-table that costs a fixed P^2 floats off-heap and is not charged to the per-vertex budget); (c) **CDH+sub+BPMX**, additionally enabling Felner-style one-step Bidirectional Pathmax during A* expansion (sound under closed-set A* without reopenings; this matches the original CDH evaluation protocol of Goldenberg et al. (2017)). Admissibility for all three arms on directed and undirected fixtures is covered by the released unit tests.

Results. Table 4 reports the CDH benchmark at matched memory across SBM (undirected), OSMnx Modena (directed), and DIMACS NY (directed); per-seed numbers are deterministic given the LCC seed vertex. Across all three graphs the entire CDH family is dominated at every matched-memory cell tested by both direct FPS-ALT and by AAC: the gap is 44–77 % in favor of FPS-ALT on SBM, 30–38 % on Modena, and 31–56 % on NY. On Modena, bound substitution does not move CDH because the per-vertex stored subsets at $r{\in}\{3, 7, 14\}$ overlap too sparsely for the side-table to fire often; BPMX (applied on top of bound substitution) adds a small ∼0.5–2 % expansion-count overhead at this graph scale while preserving admissibility (verified by the released unit tests). This is the matched-memory diagnostic's third complementary signal: the parity story extends *outside* the AAC-vs-FPS axis to the closest classical-compressor axis as well.

Scope of the CDH benchmark. Two factors contextualize the wider-than-expected gap. *(a) Regime.* Goldenberg et al. (2011; 2017) emphasize cell-grid game maps at tighter budgets (∼8–32 B/v); we test road and synthetic graphs at 32–128 B/v, where the FPS-ALT teacher is already strong (Theorem 5). *(b) CDH parameter search.* We use the canonical top-r-farthest rule with $P{=}64$; a sweep over (P, r) pairs and alternative selection rules is future work. The reference implementation and BPMX-enabled A* path are released so that further CDH tuning runs against an audited baseline.

5.9 Training-Objective Drift: A Controlled Ablation

Headline finding. The first m landmarks of AAC's K_0-landmark FPS pool *are* the FPS-ALT $K{=}m$ landmarks (same FPS seed vertex), so identity one-hot selection on the first m pool indices recovers FPS-

Table 4: **MAIN** Reference Compressed Differential Heuristics (CDH) benchmark at matched memory. CDH variants from Goldenberg et al. (2011; 2017): `CDH` (strict intersection), `CDH+sub` (Goldenberg upper/lower bound substitution; off-heap P^2 pivot side-table not charged to per-vertex budget), `CDH+sub+BPMX` (additionally enables Felner-style one-step Bidirectional Pathmax during A* expansion). Pool size P=64 in all configurations; r is the per-vertex retained subset size adjusted to match B B/v. Mean expansion-reduction % over Dijkstra; 5 seeds × 100 queries per cell; CDH std is 0.00 across seeds because the FPS-pivoted reference CDH is fully determined by the LCC seed vertex (admissibility verified by the released unit tests). AAC and FPS-ALT rows are reproduced from the canonical headline sources under the same val-split test protocol: SBM from Table 9, Modena from Table 7, NY from Table 6 (test column). CDH is freshly run under the same matched-memory protocol. Discussion in Section 5.8.4.

Graph	Method	32 B/v	64 B/v	128 B/v
SBM 10K (undir.)	AAC	88.7 %	94.0 %	96.9 %
	FPS-ALT	**90.0 %**	**94.5 %**	**97.0 %**
	CDH	12.9 %	30.0 %	52.7 %
OSMnx Modena (30K, dir.)	AAC	80.7 %	87.7 %	91.9 %
	FPS-ALT	**83.4 %**	**91.2 %**	**92.8 %**
	CDH	45.3 %	52.8 %	62.8 %
	CDH+sub	45.3 %	52.8 %	62.8 %
	CDH+sub+BPMX	43.5 %	50.7 %	61.2 %
DIMACS NY (264K, dir.)	AAC	81.3 %	88.9 %	92.6 %
	FPS-ALT	**85.7 %**	**91.1 %**	**93.7 %**
	CDH	29.9 %	46.7 %	63.0 %
	CDH+sub	29.9 %	46.7 %	63.0 %
	CDH+sub+BPMX	30.1 %	47.1 %	63.1 %

Table 5: **ABLATION** Forced-first-m diagnostic on SBM at matched memory (32 B/v, K_0=32, m=8, seed 42, 100 queries). The first 8 landmarks of the K_0=32 FPS pool coincide with ALT's K=8 landmarks; forcing AAC to select them recovers ALT exactly. Training moves away from this subset and gets worse with more epochs.

Configuration	Mean expansions	Reduction
FPS-ALT, K=8	535.7	89.95 %
AAC K_0=32, m=8, forced one-hot on first 8 (no training)	535.7	89.95 %
AAC K_0=8, m=8 trained 200 epochs (identity case)	535.7	89.95 %
AAC K_0=32, m=8 trained 200 epochs (default schedule)	573.4	89.24 %
AAC K_0=32, m=8 trained 500 epochs	612.7	88.50 %

ALT K=m *exactly* (Table 5, second row: 535.7 expansions, identical to FPS-ALT to the last digit). Yet the default-trained AAC selector drifts away from this subset and underperforms it by 0.71 % at 200 epochs and 1.45 % at 500 epochs. We read this as a clear diagnostic that the matched-memory parity reported throughout Sections 5.8.1–5.8.3 need not reflect an architectural ceiling: the architecture admits FPS-ALT trivially. Below we replicate the diagnostic across two graph families, two budgets, three seeds, and five training-epoch checkpoints (Figure 7) and a directed road graph (Appendix Table 19); identity-on-first-m initialization closes the gap entirely (Appendix Table 20), pinpointing initialization as the binding constraint and motivating it as a top-level contribution (§1, C2).

The matched-memory parity in Sections 5.8.1–5.8.3 therefore raises a sharp question: *why* does the trained compressor underperform identity selection at matched memory? To isolate the mechanism, we run a controlled five-variant diagnostic experiment on SBM (K_0=32, m=8, seed 42):

Three observations follow. (i) The forced-first-8 baseline matches ALT *exactly* (535.7 expansions, 89.95%), confirming the heuristic plumbing is correct: there is no implementation bug separating AAC from ALT when the selection is forced to the ALT landmarks. (ii) With 200 epochs of Gumbel-softmax training the compressor moves *away* from the ALT subset, scoring 89.24% (−0.71 %). (iii) With 500 epochs it scores

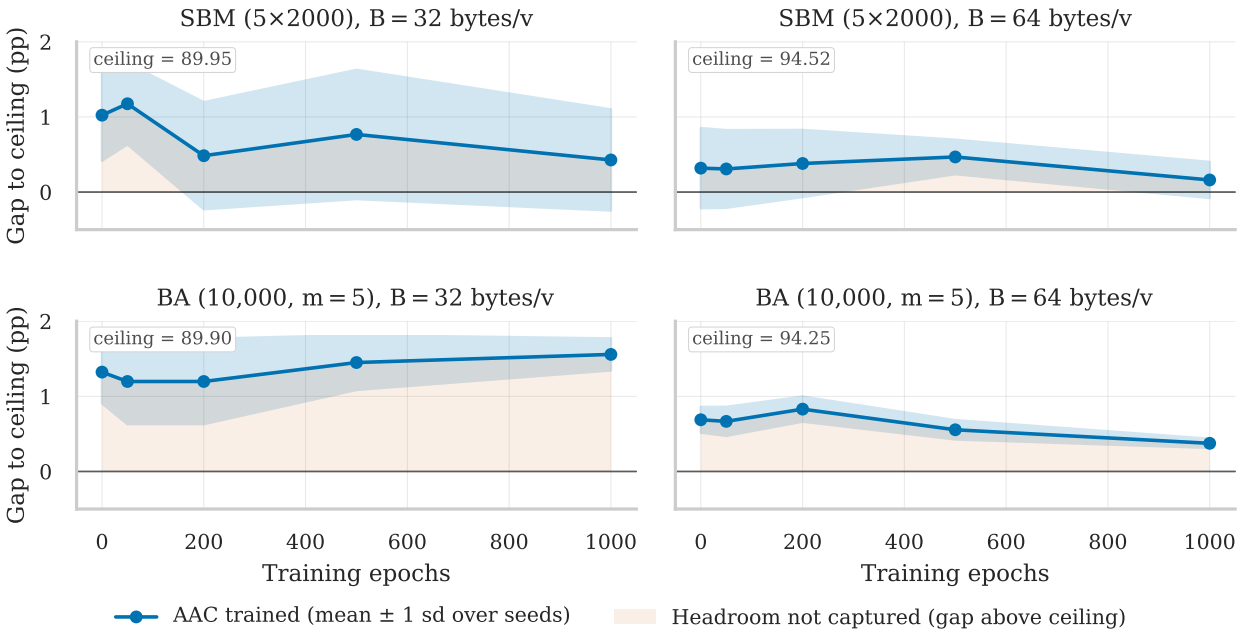

Figure 7: Training-objective drift across two graph families (rows: SBM, BA) and two memory budgets (columns: 32, 64 B/v). **Solid**: AAC expansion reduction (%) vs. training epochs (mean over 3 seeds, shaded band ±1 s.d.). **Dashed green**: forced-first-m baseline (W set to identity on the first m FPS pool indices), the architectural ceiling. **Dotted red**: FPS-ALT $K{=}m$ reference, algebraically equal to forced-first-m on these undirected cells. The trained selector stays below the ceiling across all four cells and the gap widens or oscillates rather than closing. Road-graph extension in Table 19.

88.50%, *worse* than the 200-epoch checkpoint. More training does not converge toward the ALT subset; it drifts further from it.

The mechanism is that block-sparse initialization in the compressor structurally distributes the m rows across the K_0-landmark pool rather than concentrating them on the low-index FPS-optimal subset, and the gap-to-teacher gradient (Proposition 7) minimizes an objective whose global optimum on the teacher pool is not the same subset that minimizes covering radius (cf. Figure 4 for a closed-form 1D illustration). Proposition 7 asserts that gap-to-teacher and gap-to-distance gradients are identical, not that either objective's minimum coincides with the minimum of covering radius: precisely the gap diagnosed here. A direct causal test (adding a soft covering-radius penalty $\lambda_{\text{cov}} \cdot \tilde{r}$ to the distillation loss) narrows the road-network gap by 1.3–1.4 % on Modena/Manhattan while preserving admissibility, but is neutral-to-slightly-negative on SBM and BA, consistent with the covering-radius reading that road-network coverage slack is what the regularizer exploits (full sweep in Appendix B). Our conclusion is narrower than a covering-radius-only reading would predict: architectural admissibility is cheap, but the training objective is the binding constraint on whether learned selection can match FPS at matched memory.

Multi-graph and multi-budget replication. To check that the seed-42 SBM result above is not idiosyncratic, we re-ran the forced-first-m vs. trained comparison on two graph families (SBM 5×2000, BA 10000, m=5) at two budgets (32 and 64 B/v) across three seeds (42, 43, 44) and five training-epoch checkpoints (0, 50, 200, 500, 1000). Figure 7 plots the resulting expansion-reduction curves against the forced-first-m and FPS-ALT $K{=}m$ horizontal references.

The pattern is consistent across all four (graph, budget) cells: the trained selector stays 0.3–1.4 % *below* forced-first-m throughout the 0–1000-epoch window, and additional training oscillates rather than closing the gap. Per-seed worst-case residuals reach ∼0.8 % on SBM B=64 during the training window (with the seed-mean residual smaller at any single checkpoint due to variance across the three seeds); BA B=32 is

the tightest cell, with seed-mean residual within ∼0.5 % throughout. No cell shows monotone convergence toward the ALT identity subset under the default training schedule. The hyperparameter-and-initialization sweep below (Appendix Table 20) shows that an identity-on-first-m initializer closes the gap entirely; the binding constraint in the regimes we test is initialization rather than architectural capacity.

ALT-pool (first m) arm. A companion diagnostic (Appendix Table 17) runs FPS-ALT using the first m landmarks of the same K_0-landmark pool AAC is trained on, across seven graphs (SBM, BA, NY-DIMACS, Modena, Manhattan, Berlin, LA). On every graph and budget the arm is numerically equal to canonical FPS-ALT at matched memory, confirming the algebraic identity underlying the forced-first-m construction: "AAC with identity selection on the first-m FPS indices" is ALT-at-matched-memory. The residual AAC-vs-ALT gap on matched-memory non-road comparisons is training-objective drift off this subset, not architectural capacity.

Road-graph extension and initialization robustness. The same forced-first-m vs. trained comparison on DIMACS NY (264K nodes, directed, K=8, B=64 B/v, three seeds; Appendix Table 19) reproduces the qualitative pattern: the trained selector stays 1.7–3.3 % below the FPS-ALT ceiling and the gap widens monotonically from −1.67 % at epoch 50 to −3.28 % at epoch 1000, ruling out the alternative explanation that the drift is specific to synthetic graphs. A hyperparameter sweep on the SBM B=32 cell over learning rate, batch size, and K_0 leaves the drift at −0.6 to −1.7 % under the default block-sparse initialization but *eliminates* it (+0.00 % at every cell) under an identity-on-first-m init that already sits at the ceiling (Appendix Table 20). The binding constraint is the joint (init, gradient) trajectory, not the optimization landscape itself.

5.10 Compatibility Case Study: Warcraft 12×12 Grids

Appendix F evaluates AAC in a differentiable planning pipeline on Warcraft 12×12 grid maps (Vlastelica et al., 2020) as a compatibility check, not a primary benchmark. The frozen compressor composes with an end-to-end encoder and the full pipeline trains to completion while preserving deployment-time admissibility (zero violations on every evaluated instance). AAC's deployed heuristic carries an admissibility certificate w.r.t. the predicted-cost graph at every checkpoint (the encoder learns; the compressor stays a structural max-of-admissibles by Proposition 2). Per-method numbers are in Table 23; closing the imitation-fidelity gap to blackbox differentiation under this constraint is left to future work.

6 Discussion

The FPS ceiling and where it breaks. The matched-memory evidence (Table 2) reveals a structural fact about ALT-family landmark selection: FPS's Gonzalez 2-approximation on covering radius (Corollary 6) is already tight on the graphs we tested, so no selector, classical or learned, can substantially improve expansion count at matched memory. The classical CDH compressor is dominated by both AAC and FPS-ALT at every matched-memory cell tested (Section 5.8.4), confirming that the compression operator is not the bottleneck. On expansion count, FPS-ALT leads AAC on every metric regime under the val-split protocol (Table 16); the architectural admissibility constraint itself is free, and the cost lives in the gap-to-teacher gradient under the default training schedule (Section 5.9). On wall-clock latency at our chosen storage layout, the picture flips: AAC is faster on every DIMACS graph at p50 (Table 3), with the interleaved-ALT layout as the open control for this claim. The only expansion-count cell where AAC leads under val-split is OGB-arXiv at $B \geq 64$, precisely the regime where FPS coverage is *not* near-optimal. The closest classical admissible compressor, CDH, is dominated by both AAC and FPS-ALT at every matched-memory cell tested (Section 5.8.4), confirming that the compression operator itself is not the bottleneck in the ALT family. The residual headroom for learned selection lives in three directions (Section 6.2): non-FPS candidate pools, non-Euclidean or instance-cost graphs, and composition with classical admissible toolchains.

What architectural admissibility makes possible. Architectural admissibility (Proposition 2) is what made the matched-memory diagnostic possible: it removes optimization-dependent confounds from the comparison (the deployed heuristic is provably safe at every checkpoint, Table 11), it lets the same compressor

be inserted into a differentiable encoder pipeline as an existence proof (Appendix F), and at deployment it reduces to classical ALT on a learned subset (Proposition 4) so the classical toolchain (bi-directional search, BPMX, CDH-style bound substitution) remains available.

A methodological by-product. The accounting convention (whether ALT's K counts forward landmarks only or both directions) can shift matched-memory comparisons by an order of magnitude even when implementation, training, and evaluation are held fixed. We therefore lock the convention and state it explicitly (Section 5.2, Appendix E E.2) before pre-registering effect sizes. The OSMnx Manhattan crossover at $m{=}32, 64$ on Table 22 is the matching pool-size effect at $m{\approx}K_0/2$ explained in Section 3.3.

Budget–performance tradeoff. The compression curve (Table 22) reveals the three-way relationship between the full teacher (FPS-ALT at $K_0{=}64$, 512 B/v), AAC at compression ratio m/K_0, and matched-budget ALT at $K{=}m/2$. On Modena: FPS-Full achieves 95.5%, AAC at $m{=}16$ (64 B/v) 84.9%, and ALT at $K{=}8$ (64 B/v) 91.2%, confirming that at 8× compression, lossy selection costs 10.6 % relative to the uncompressed teacher while direct FPS costs only 4.3 %. As m approaches K_0 the crossover surfaces (Manhattan at $m{=}32$, AAC 92.2% vs. ALT 92.1%): this reflects access to a K_0-landmark pool versus a fresh $K_0/2$-landmark FPS run, not a violation of the capacity ceiling. At $m{=}K_0$ AAC effectively recovers the teacher (Proposition 4), and the asymmetry vanishes on undirected graphs where the matched-memory rule is ALT $K{=}m$. The matched-memory question for memory-constrained deployment remains decided by FPS.

6.1 Limitations

- *We do not claim AAC replaces specialized routing engines.* Contraction hierarchies (Geisberger et al., 2008), hub labeling (Abraham et al., 2012), CRP (Delling et al., 2011), CCH (Dibbelt et al., 2016), and CH-Potentials (Strasser & Zeitz, 2021) achieve orders-of-magnitude speedup on static and dynamic road networks; we target graphs without strong hierarchy and settings where differentiability is required.
- *We find ALT ahead of AAC on expansion count on every road network we tested under the headline* 64 *B/v val-split protocol; the only retrospective best-K_0 exception we see is OSMnx Manhattan at* 128 *B/v, which we attribute to a pool-size effect (Section 3.3).* On DIMACS at 128 B/v, FLA: ALT 92.8 ± 0.0% vs. AAC 92.1 ± 1.1% under the retrospective best-K_0 protocol of Table 14 (val-split Table 6 test cells are tied at 91.8%/91.8%); the median per-seed Wilcoxon p-value at COL-128 B/v is $p \approx 0.23$ but the Stouffer-combined $p \approx 1.3{\times}10^{-3}$ remains FDR-significant (Table 16), consistent with FPS near-optimality on highway structure. On directed OSMnx city graphs at 64 B/v, ALT leads by 1.5–3.9 % (Berlin, Manhattan, Modena, LA, Netherlands). *The expansion-count limitation does not extend to wall-clock latency:* at the same 64 B/v matched-memory configuration AAC is faster than ALT at p50 on every DIMACS graph and at p95 on three of four (Table 3), and amortizes its higher offline preprocessing within 170–1,924 queries (Figure 8).
- *Mechanism: covering radius binds on roads; training-objective drift binds elsewhere.* FPS's Gonzalez 2-approximation on r_m ensures near-optimal spatial coverage on road networks; the gap-to-teacher gradient (Proposition 7) is correlated with but not identical to covering-radius minimization (cf. Figure 4), and we observe the trained compressor drift *away* from the FPS-from-pool optimum with more training on non-road graphs (Section 5.9).
- *We treat the contextual variant as a compatibility check, not a performance result.* Blackbox differentiation achieves 54.7% path match vs. AAC's 20.3 ± 2.4% on Warcraft 12×12 (Appendix F); the appendix establishes that architectural admissibility survives insertion into the differentiable pipeline. We leave closing the imitation-fidelity gap to future work.
- *Initialization is the binding constraint on the expansion-count gap.* Under default block-sparse initialization the trained selector drifts away from the FPS-from-pool optimum (Section 5.9); identity-on-first-m initialization closes the gap entirely (Appendix Table 20).
- *Seed and query-distribution scope.* Our static experiments use 5 seeds (3 for the 4.5M-node Netherlands graph and Warcraft); query-distribution sensitivity covers uniform/hotspot/powerlaw on NY and Manhattan (Appendix B). Production query logs may exhibit richer structure not captured here.

6.2 Future Directions

Three falsifiable follow-ups open from the architectural guarantee and the matched-memory result. **(1) Richer candidate pools.** Theorem 5's covering ceiling is taken against the FPS pool. A pool drawn from non-FPS sources (planar partitioning, avoid/maxcover landmarks (Goldberg & Harrelson, 2005), boundary landmarks (Efentakis & Pfoser, 2013), or learned pool generators trained on instance features) could provide headroom that no FPS-pool compressor can recover. AAC is pool-agnostic, so this is a drop-in swap. **(2) Non-Euclidean and instance-cost regimes.** The OGB-arXiv and Pareto-detail evidence both hint that residual headroom for learned selection lives in graphs whose metric is not well-approximated by FPS coverage (citation hubs, small-world structure, and predicted-cost grids; Appendix F). A targeted study on these regimes paired with a non-FPS pool from (1) is the natural follow-up; the subset-selection formulation of Rayner et al. (2013) also suggests re-asking the question with non-landmark hypothesis classes (learned Euclidean embeddings (Rayner & Bowling, 2011), submodular-coverage-aware selectors). **(3) Composition with classical admissible toolchains.** Because AAC deploys as ALT-on-a-subset, it inherits BPMX propagation, CDH with bound substitution (Section 5.8.4), CH-Potentials (Strasser & Zeitz, 2021) as an admissible upper bound, and bi-directional landmark tightening; a matched-memory benchmark of AAC $\circ$ CDH-with-BPMX would close the third side of the admissible landmark-compression triangle.

7 Conclusion

Summary. AAC establishes that differentiability and deployment-time admissibility can coexist in a single ALT-family layer (Proposition 2); the cost is in optimization, not in architecture: the binding constraint is training-objective drift under default initialization, not the hypothesis class (Section 5.9). Under matched per-vertex memory, the differentiable construction trades a sub-4-percentage-point expansion deficit on metric regimes for end-to-end differentiability and a structural admissibility certificate; the headline numbers live in Table 2.

The matched-memory protocol (Section 5.2; paired TOST + FDR-corrected Wilcoxon + forced-first-m diagnostics + pre-registration audit in Appendix E) is the methodological contribution the community can re-use to compare any future learned admissible heuristic on common terms. We trace the parity to a covering-radius fact (Theorem 5): FPS's Gonzalez 2-approximation on r_m is already tight on the tested graphs, and the one regime where the learned selector leads (OGB-arXiv at $B \geq 64$) is precisely the one where FPS coverage is not near-optimal, motivating the pool-agnostic follow-up directions of Section 6.2. Artifact: `https://anonymous.4open.science/r/aac` (full source tree, run logs, and the verbatim pre-registration; see Appendix G).

References

Ittai Abraham, Daniel Delling, Andrew V. Goldberg, and Renato F. Werneck. Hierarchical hub labelings for shortest paths. In *Proceedings of the 20th European Symposium on Algorithms (ESA)*, volume 7501 of *LNCS*, pp. 24–35. Springer, 2012.

Takuya Akiba, Yoichi Iwata, and Yuichi Yoshida. Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 349–360, 2013.

Aleksandr Ananikian, Daniil Drozdov, and Konstantin Yakovlev. UPath: Universal planner across topological heterogeneity for grid-based pathfinding. *arXiv preprint arXiv:2602.23789*, 2026. doi: 10.48550/arXiv.2602.23789.

Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *PMLR*, pp. 444–453, 2019.

Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F. Werneck. Route planning in transportation networks. *Algorithm Engineering*, pp. 19–80, 2016. arXiv:1504.05140.

Reinhard Bauer, Daniel Delling, Peter Sanders, Dennis Schieferdecker, Dominik Schultes, and Dorothea Wagner. Combining hierarchical and goal-directed speed-up techniques for Dijkstra's algorithm. In *ACM Journal of Experimental Algorithmics*, volume 15, pp. 2.3:1–2.3:31, 2010. Discusses NP-hardness of optimal ALT landmark selection.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Luca Bertinetto, João F. Henriques, Samuel Albanie, Michela Paganini, and Gül Varol (eds.). *Proceedings of the NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, volume 148 of *Proceedings of Machine Learning Research*, 2021. PMLR. Pilot study of pre-registration review in machine learning; held virtually 11 December 2020.

Geoff Boeing. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

Justine Cauvi, Ruoying Li, and Sabine Storandt. Landmark hub labeling: Improved bounds and faster query answering. In *24th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*, volume 123 of *OASIcs*, pp. 1:1–1:17. Schloss Dagstuhl, 2024. doi: 10.4230/OASIcs.ATMOS.2024.1.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *PMLR*, pp. 883–892, 2018.

Xiangyu Chen, Fan Yang, and Chen Wang. ia*: Imperative learning-based A* search for path planning. *IEEE Robotics and Automation Letters*, 10(12):12987–12994, 2025. doi: 10.1109/LRA.2025.3625500.

Gautam Choudhary, Libin Zhou, Yeasir Rayhan, and Walid G. Aref. An empirical survey and benchmark of learned distance indexes for road networks. *arXiv preprint arXiv:2602.04068*, 2026.

Edith Cohen, Eyal Halperin, Haim Kaplan, and Uri Zwick. Reachability and distance queries via 2-hop labels. *SIAM Journal on Computing*, 32(5):1338–1355, 2003.

Liron Cohen, Tansel Uras, Shiva Jahangiri, Aliyah Arunasalam, Sven Koenig, and T. K. Satish Kumar. The FastMap algorithm for shortest path computations. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1427–1433, 2018.

Louann Coste, Ruoying Li, Sabine Storandt, and Tobias Töpfer. The complexity of landmark hub labeling. In *Proceedings of the 24th International Conference on Algorithms and Complexity (CIAC)*, LNCS. Springer, 2025.

Joseph C. Culberson and Jonathan Schaeffer. Pattern databases. *Computational Intelligence*, 14(3):318–334, 1998.

Daniel Delling, Andrew V. Goldberg, Thomas Pajor, and Renato F. Werneck. Customizable route planning. In *10th International Symposium on Experimental Algorithms (SEA)*, volume 6630 of *LNCS*, pp. 376–387. Springer, 2011.

Camil Demetrescu, Andrew V. Goldberg, and David S. Johnson. The shortest path problem: Ninth DIMACS implementation challenge. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 74. AMS, 2009.

Julian Dibbelt, Ben Strasser, and Dorothea Wagner. Customizable contraction hierarchies. *ACM Journal of Experimental Algorithmics*, 21(1):1–49, 2016.

Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1): 269–271, 1959.

Jinchun Du, Bojie Shen, and Muhammad Aamir Cheema. EHL*: Memory-budgeted indexing for ultrafast optimal euclidean pathfinding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. arXiv:2408.11341.

Talya Eden, Piotr Indyk, and Haike Xu. Embeddings and labeling schemes for A*. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 215 of *LIPIcs*, pp. 62:1–62:19, 2022. doi: 10.4230/LIPIcs.ITCS.2022.62.

Alexandros Efentakis and Dieter Pfoser. Optimizing landmark-based routing and preprocessing. *Proceedings of the 6th ACM SIGSPATIAL Workshop on Computational Transportation Science*, 2013.

Christos Faloutsos and King-Ip Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pp. 163–174. ACM, 1995.

Ariel Felner, Richard E. Korf, Ram Meshulam, and Robert C. Holte. Compressed pattern databases. *Journal of Artificial Intelligence Research (JAIR)*, 30:213–247, 2007. doi: 10.1613/jair.2241.

Ehsan Futuhi and Nathan R. Sturtevant. Learning admissible heuristics for A*: Theory and practice. In *International Conference on Learning Representations (ICLR)*, 2026. doi: 10.48550/arXiv.2509.22626. Poster; OpenReview ID `WAQIxi7ifb`; preprint arXiv:2509.22626.

Robert Geisberger, Peter Sanders, Dominik Schultes, and Daniel Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *Proceedings of the 7th International Workshop on Experimental Algorithms (WEA)*, volume 5038 of *LNCS*, pp. 319–333. Springer, 2008.

Andrew V. Goldberg and Chris Harrelson. Computing the shortest path: A* search meets graph theory. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 156–165, 2005.

Andrew V. Goldberg, Haim Kaplan, and Renato F. Werneck. Better landmarks within reach. In *Proceedings of the 6th International Workshop on Experimental Algorithms (WEA)*, LNCS, pp. 38–51. Springer, 2007.

Meir Goldenberg, Nathan R. Sturtevant, Ariel Felner, and Jonathan Schaeffer. The compressed differential heuristic. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 24–29, 2011.

Meir Goldenberg, Ariel Felner, Amir Palombo, Neil Burch, and Nathan R. Sturtevant. The compressed differential heuristic. *AI Communications*, 30(6):393–418, 2017. doi: 10.3233/AIC-170743.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi: 10.1016/0304-3975(85)90224-5.

Gal Hadar, Forest Agostinelli, and Shahaf S. Shperberg. Beyond single-step updates: Reinforcement learning of heuristics with limited-horizon search. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 36955–36963, 2026. doi: 10.1609/aaai.v40i43.41023. Vol. 40, No. 43; preprint arXiv:2511.10264.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Malte Helmert, Patrik Haslum, Jörg Hoffmann, and Raz Nissim. Merge-and-shrink abstraction: A method for generating lower bounds in factored state spaces. *Journal of the ACM*, 61(3):1–63, 2014.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 22118–22133, 2020.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

Daniil Kirilenko, Anton Andreychuk, Aleksandr Panov, and Konstantin Yakovlev. TransPath: Learning heuristics for grid-based pathfinding via transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 12436–12443, 2023.

Alan Lahoud, Erik Schaffernicht, and Johannes Stork. DataSP: A differential all-to-all shortest path algorithm for learning costs and predicting paths with context. In *Proceedings of the 40th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 244 of *PMLR*, pp. 2094–2112, 2024.

Daniël Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

Alfred Nilsson, Klas Wijk, Sai Bharath Chandra Gutha, Erik Englesson, Alexandra Hotti, Carlo Saccardi, Oskar Kviman, Jens Lagergren, Ricardo Vinuesa Motilva, and Hossein Azizpour. Indirectly parameterized concrete autoencoders. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 38237–38252, 2024. arXiv:2403.00563. IP-CAE addresses Concrete Autoencoder duplicate-selection instability by learning an embedding and a mapping to Gumbel-Softmax distribution parameters.

Carlos Nunez-Molina, Masataro Asai, Pablo Mesejo, and Juan Fernandez-Olivares. On using admissible bounds for learning forward search heuristics. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6761–6769, 2024. doi: 10.24963/ijcai.2024/747.

Michal Pandy, Weikang Qiu, Gabriele Corso, Petar Velickovic, Zhitao Ying, Jure Leskovec, and Pietro Lio. Learning graph search heuristics. In *Proceedings of the First Learning on Graphs Conference (LoG)*, volume 198 of *PMLR*, pp. 10:1–10:13, 2022.

Anselm Paulus, Michal Rolinek, Vit Musil, Brandon Amos, and Georg Martius. CombOptNet: Fit the right NP-hard problem by learning integer programming constraints. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pp. 8443–8453, 2021.

Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Addison-Wesley, Reading, MA, 1984.

D. Chris Rayner and Michael Bowling. Euclidean heuristic optimization. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 81–86, 2011.

D. Chris Rayner, Nathan Sturtevant, and Michael Bowling. Subset selection of search heuristics. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 637–643, 2013.

Shinsaku Sakaue and Taihei Oki. Sample complexity of learning heuristic functions for greedy-best-first and A* search. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 4713–4726, 2022.

Ameesh Shah, Eric Zhan, Jennifer J. Sun, Abhinav Verma, Yisong Yue, and Swarat Chaudhuri. Learning differentiable programs with admissible neural heuristics. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. Neural relaxations of partial programs as admissible A* heuristics for differentiable program synthesis.

Sabine Storandt. Algorithms for landmark hub labeling. In *33rd International Symposium on Algorithms and Computation (ISAAC)*, volume 248 of *LIPIcs*, pp. 5:1–5:17. Schloss Dagstuhl, 2022. doi: 10.4230/LIPIcs.ISAAC.2022.5.

Sabine Storandt. Scalable landmark hub labeling for optimal and bounded suboptimal pathfinding. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6788–6795, 2024.

Ben Strasser and Tim Zeitz. A fast and tight heuristic for A* in road networks. In *19th International Symposium on Experimental Algorithms (SEA)*, volume 190 of *LIPIcs*, pp. 6:1–6:16. Schloss Dagstuhl, 2021. Known as CH-Potentials; full version in ACM JEA 2023.

Nathan R. Sturtevant, Ariel Felner, Max Barer, Jonathan Schaeffer, and Neil Burch. Memory-based heuristics for explicit state spaces. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 609–614, 2009.

Marin Vlastelica, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. Differentiation of blackbox combinatorial solvers. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

Ryo Yonetani, Tatsunori Taniai, Mohammadamin Barekatain, Mai Nishimura, and Asako Kanezaki. Path planning using neural A* search. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pp. 12029–12039, 2021.

A Extended Proofs

Full Proof of Proposition 2 (Row-Stochastic Compression Preserves Admissibility, after Pearl 1984)

Proof. We prove both cases by showing that row-stochastic linear maps cannot amplify coordinate-wise differences.

Part (a): undirected case. Define $\delta_k := d(l_k, u) - d(l_k, t)$ for $k \in [K_0]$. From Eq. (3):

$$y_i(u) - y_i(t) = \sum_{k=1}^{K_0} A_{i,k}\, \delta_k.$$

Since $A_{i,k} \geq 0$ and $\sum_k A_{i,k} = 1$, this is a convex combination of $\{\delta_k\}_{k=1}^{K_0}$. For any convex combination:

$$\min_k \delta_k \;\leq\; \sum_k A_{i,k}\, \delta_k \;\leq\; \max_k \delta_k.$$

The upper bound follows because $\sum_k A_{i,k}\, \delta_k \leq \sum_k A_{i,k} \max_j \delta_j = \max_j \delta_j$. The lower bound is symmetric. Taking absolute values:

$$\Big|\sum_k A_{i,k}\, \delta_k\Big| \leq \max\big(|\max_k \delta_k|,\; |\min_k \delta_k|\big) = \max_k |\delta_k|.$$

To see this: if $\sum_k A_{i,k}\delta_k \geq 0$, then $|\sum_k A_{i,k}\delta_k| \leq \max_k \delta_k \leq |\max_k \delta_k| \leq \max_k |\delta_k|$. If $\sum_k A_{i,k}\delta_k < 0$, then $|\sum_k A_{i,k}\delta_k| = -\sum_k A_{i,k}\delta_k \leq -\min_k \delta_k = |\min_k \delta_k| \leq \max_k |\delta_k|$.

Therefore $|y_i(u) - y_i(t)| \leq \max_k |\delta_k|$ for every i, and taking the max over i:

$$h_A(u,t) = \max_i |y_i(u) - y_i(t)| \leq \max_k |d(l_k, u) - d(l_k, t)| = h_{\mathrm{ALT}}(u,t).$$

The final inequality $h_{\mathrm{ALT}}(u,t) \leq d(u,t)$ follows from Theorem 1.

Part (b): directed case. For the backward bound: define $\delta_k^{\mathrm{bwd}} := d_{\mathrm{in}}(k,u) - d_{\mathrm{in}}(k,t) = d(u,l_k) - d(t,l_k)$. Then:

$$y_i^{\mathrm{bwd}}(u) - y_i^{\mathrm{bwd}}(t) = \sum_k A_{i,k}^{\mathrm{bwd}}\, \delta_k^{\mathrm{bwd}} \;\leq\; \max_k \delta_k^{\mathrm{bwd}} = \max_k \big(d(u,l_k) - d(t,l_k)\big),$$

by the convex combination bound. For the forward bound: define $\delta_k^{\mathrm{fwd}} := d_{\mathrm{out}}(k,t) - d_{\mathrm{out}}(k,u) = d(l_k,t) - d(l_k,u)$. Then:

$$y_i^{\mathrm{fwd}}(t) - y_i^{\mathrm{fwd}}(u) = \sum_k A_{i,k}^{\mathrm{fwd}}\, \delta_k^{\mathrm{fwd}} \;\leq\; \max_k \delta_k^{\mathrm{fwd}} = \max_k \big(d(l_k,t) - d(l_k,u)\big).$$

Taking the max over i for each bound and then the max over both bounds:

$$h_A(u,t) \leq \max\Big(\max_k \big(d(u,l_k) - d(t,l_k)\big),\; \max_k \big(d(l_k,t) - d(l_k,u)\big),\; 0\Big) = h_{\mathrm{ALT}}(u,t).$$

Admissibility follows from Theorem 1. □

We also provide the complete proof of Theorem 5.

Undirected Case

Let $G = (V, E, w)$ be an undirected weighted graph with positive edge weights. Let $S = \{l_1, \ldots, l_m\} \subseteq L$ be an m-element landmark subset with covering radius $r_m = \max_{v \in V} \min_{l \in S} d(v,l)$.

Claim. For all $u, t \in V$ with finite $d(u,t)$: $d(u,t) - h_{\mathrm{ALT}}^S(u,t) \leq 2\, r_m$.

Proof. The ALT heuristic on the subset S is $h^S_{\text{ALT}}(u,t) = \max_{l\in S} |d(l,u) - d(l,t)|$.

Choose $l^* = \arg\min_{l\in S} d(u,l)$. By the covering radius definition, $d(u,l^*) \le r_m$.

Consider the forward bound for l^*:

$$d(u,t) - \big[d(l^*,t) - d(l^*,u)\big] = d(u,t) + d(l^*,u) - d(l^*,t).$$

By the triangle inequality, $d(u,t) \le d(u,l^*) + d(l^*,t)$, so $d(l^*,t) \ge d(u,t) - d(u,l^*)$. Substituting:

$$d(u,t) + d(l^*,u) - d(l^*,t) \le d(u,t) + d(l^*,u) - \big[d(u,t) - d(u,l^*)\big] = d(l^*,u) + d(u,l^*).$$

Since G is undirected, $d(l^*,u) = d(u,l^*) \le r_m$, so:

$$d(u,t) + d(l^*,u) - d(l^*,t) \le 2\, d(u,l^*) \le 2\, r_m.$$

Since $h^S_{\text{ALT}}(u,t) \ge d(l^*,t) - d(l^*,u)$ (the max over all landmarks and bound types is at least the forward bound for l^*):

$$d(u,t) - h^S_{\text{ALT}}(u,t) \le d(u,t) - \big[d(l^*,t) - d(l^*,u)\big] \le 2\, r_m. \qquad \square$$

Directed Case

Let $G = (V,E,w)$ be a directed weighted graph. The ALT heuristic uses both forward bounds $d(l,t) - d(l,u)$ and backward bounds $d(u,l) - d(t,l)$.

Define the symmetrized covering radius:

$$r_m^{\text{sym}} := \max_{v\in V_{\text{reach}}} \min_{l\in S} \max\big(d(l,v),\, d(v,l)\big),$$

where $V_{\text{reach}} = \{v \in V : \exists\, l \in S \text{ with } d(l,v) < \infty \text{ and } d(v,l) < \infty\}$.

Proof. We bound the gap for forward and backward bounds separately.

Forward bound. Choose $l^* \in S$ minimizing $\max(d(l^*,u), d(u,l^*))$ over reachable landmarks. Then $d(l^*,u) \le r_m^{\text{sym}}$ and $d(u,l^*) \le r_m^{\text{sym}}$.

The forward gap for l^* is:

$$d(u,t) - \big[d(l^*,t) - d(l^*,u)\big] = d(u,t) + d(l^*,u) - d(l^*,t).$$

By the triangle inequality (directed): $d(l^*,t) \ge d(u,t) - d(u,l^*)$, since $d(u,t) \le d(u,l^*) + d(l^*,t)$ (using the path $u \to l^* \to t$; finiteness of $d(u,l^*)$ is guaranteed by the symmetrized covering radius definition which ensures $\max(d(l^*,u), d(u,l^*)) \le r_m^{\text{sym}} < \infty$). Proceeding:

$$\begin{aligned} d(u,t) + d(l^*,u) - d(l^*,t) &\le d(u,t) + d(l^*,u) - \big[d(u,t) - d(u,l^*)\big] \\ &= d(l^*,u) + d(u,l^*) \\ &\le 2\, r_m^{\text{sym}}. \end{aligned}$$

Backward bound. Choose $l^\dagger \in S$ minimizing $\max(d(l^\dagger,t), d(t,l^\dagger))$. Then $d(l^\dagger,t) \le r_m^{\text{sym}}$ and $d(t,l^\dagger) \le r_m^{\text{sym}}$.

The backward gap for $l^\dagger$ is:

$$d(u,t) - \big[d(u,l^\dagger) - d(t,l^\dagger)\big] = d(u,t) - d(u,l^\dagger) + d(t,l^\dagger).$$

By the directed triangle inequality, $d(u,t) \le d(u,l^\dagger) + d(l^\dagger,t)$, so $d(u,t) - d(u,l^\dagger) \le d(l^\dagger,t)$:

$$\begin{aligned} d(u,t) - d(u,l^\dagger) + d(t,l^\dagger) &\le d(l^\dagger,t) + d(t,l^\dagger) \\ &\le 2\, r_m^{\text{sym}}. \end{aligned}$$

Taking the max over all landmarks and both bound types for the ALT heuristic:

$$d(u,t) - h^S_{\text{ALT}}(u,t) \le 2\, r_m^{\text{sym}}. \qquad \square$$

Proof of Corollary 6

Proof. If S is selected by farthest-point sampling (Gonzalez, 1985), the Gonzalez 2-approximation theorem for the K-center problem guarantees $r_m \le 2\, r_m^*$, where r_m^* is the minimum possible covering radius over all m-element subsets. Substituting into the bound of Theorem 5:

$$d(u,t) - h^S_{\mathrm{ALT}}(u,t) \le 2\, r_m \le 2 \cdot 2\, r_m^* = 4\, r_m^*. \qquad \square$$

Compressor Sampling and Inference Details

The training-time selection matrix is sampled from a hard Gumbel-softmax with straight-through estimator (Jang et al., 2017; Maddison et al., 2017; Bengio et al., 2013):

$$A = \mathrm{GumbelSoftmax}(W/\tau,\ \mathrm{hard} = \mathrm{True}). \tag{7}$$

The forward pass returns one-hot rows; the backward pass uses the soft Gumbel-softmax Jacobian. At deployment, hard argmax over W yields one-hot rows directly:

$$y_i(v) = d(l_{j^*(i)}, v), \qquad j^*(i) = \mathrm{argmax}_j\, W_{i,j}. \tag{8}$$

Both forms are special cases of the row-stochastic compressor of Section 3.2 (Eq. 3), so Proposition 2 applies pointwise.

Smooth-Max Differentiable Surrogate

This section establishes an exact-label observation: the hard max in the heuristic evaluation can be smoothed without breaking admissibility. This result is *not* the source of the contextual deployment guarantee (that comes from recomputing exact distances at deployment, Section 3.4); it completes the end-to-end differentiable chain for training. The static experiments in Section 5.3–5.4 do *not* use this surrogate: they train with the straight-through Gumbel-softmax estimator of Section 3.2 and the hard max. The smooth-max analysis below is used only in the contextual variant (Section 3.4), where it completes the end-to-end differentiable chain.

Theorem 8 (Smooth lower bound)**.** *For* $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{R}^m$ *and temperature* $T > 0$*, define:*

$$M_T(\mathbf{x}) := \frac{1}{T} \log \sum_{i=1}^{m} \exp(T x_i) - \frac{\log m}{T}. \tag{9}$$

Then $M_T(\mathbf{x}) \le \max(\mathbf{x})$ *for all* $\mathbf{x}$*, with* $M_T(\mathbf{x}) \to \max(\mathbf{x})$ *as* $T \to \infty$*.*

Proof. Let $x^* = \max_i x_i$. The standard log-sum-exp bound gives:

$$T x^* \ \le\ \log \sum_{i=1}^{m} \exp(T x_i) \ \le\ T x^* + \log m. \tag{10}$$

Left inequality: $\sum_i \exp(T x_i) \ge \exp(T x^*)$, so $\log \sum \ge T x^*$. *Right inequality:* $\exp(T x_i) \le \exp(T x^*)$ for all i, so $\sum_i \exp(T x_i) \le m \exp(T x^*)$, giving $\log \sum \le T x^* + \log m$. Dividing Eq. (10) by T and subtracting $\frac{\log m}{T}$:

$$x^* - \frac{\log m}{T} \ \le\ M_T(\mathbf{x}) \ \le\ x^*.$$

The right inequality gives $M_T(\mathbf{x}) \le \max(\mathbf{x})$. The left inequality gives $M_T(\mathbf{x}) \ge \max(\mathbf{x}) - \frac{\log m}{T} \to \max(\mathbf{x})$ as $T \to \infty$. $\square$

Corollary 9 (End-to-end admissibility chain)**.** *Replacing* max *with the smooth log-sum-exp* M_T *in the compressed heuristic yields a smooth heuristic* $\tilde{h}_A$ *satisfying:*

$$\tilde{h}_A(u,t) \ \le\ h_A(u,t) \ \le\ h_{\mathrm{ALT}}(u,t) \ \le\ d(u,t). \tag{11}$$

The first inequality is Theorem 8, the second is Proposition 2, and the third is Theorem 1. In the static exact-label setting, this chain preserves a lower bound throughout training; it does not apply to the contextual smooth-min proxy of Section 3.4, where deployment-time admissibility on the predicted-cost graph is handled by re-running exact Dijkstra from the selected landmarks.

Directed-Graph Extension: Derivation, Memory Accounting, and Covering Radius

This subsection collects the directed-graph extensions of the AAC architecture, the matched-memory accounting convention, and the covering-radius bound; the main-text statement is the compressed paragraph at the end of Section 3.

(a) Compressor. For undirected graphs ($d_{\text{out}} = d_{\text{in}}$), a single matrix $A \in \mathbb{R}^{m \times K_0}$ acts on a single label tensor. For directed graphs, we instantiate two independent matrices $A_{\text{fwd}}, A_{\text{bwd}}$ of sizes $m_{\text{fwd}} \times K_0$ and $m_{\text{bwd}} \times K_0$ with $m_{\text{fwd}} + m_{\text{bwd}} = m$ (default $m_{\text{fwd}} = \lfloor m/2 \rfloor$). Both matrices are row-stochastic; Proposition 2(b) handles the directed case with the same convex-combination argument applied separately to the forward and backward bound types of Eq. (4).

(b) Memory accounting. On directed graphs ALT stores both d_{out} and d_{in} ($2K$ floats/vertex); on undirected graphs $d_{\text{in}} \equiv d_{\text{out}}$ and ALT stores only K floats/vertex. Consequently the matched-deployed-label-memory rule at B B/v is AAC $m{=}B/4$ vs. ALT $K{=}B/8$ (directed) and AAC $m{=}B/4$ vs. ALT $K{=}B/4$ (undirected). AAC stores m floats per vertex in both cases; for directed graphs the budget is split $\lfloor m/2 \rfloor$ forward and $\lceil m/2 \rceil$ backward.

(c) Covering radius. The covering-radius bound (Theorem 5) uses the symmetrized form $r_m^{\text{sym}} := \max_{v \in V_{\text{reach}}} \min_{l \in S} \max(d(l, v), d(v, l))$ on directed graphs and the standard r_m on undirected graphs. The Gonzalez 2-approximation guarantee on r_m is preserved under symmetrization because a 2-approximation of an upper-bound metric is still a 2-approximation of any quantity bounded above by that metric.

(d) Sentinel masking. Sentinel masking (Section 2) removes landmarks with infinite distances from the max computation. This can only reduce the heuristic value (removing terms from a max cannot increase it), so the bound $d(u, t) - h^S_{\text{ALT}}(u, t) \leq 2\, r_m^{\text{sym}}$ is preserved over the reachable subset: the actual gap may exceed $2\, r_m^{\text{sym}}$ only if u or t has no reachable landmark, in which case the heuristic falls back to $h = 0$ (Dijkstra behavior, documented in the reachability assumption of Section 2.1).

B Extended Experiment Tables

This appendix collects per-budget detail tables that source the main-text headline (Table 2) and the additional robustness checks. The material is organized into three thematic groups: (B.1) per-graph headline detail tables that source the matched-memory comparison; (B.2) hybrid and admissibility-stress robustness checks; (B.3) the cost decomposition (query-distribution sensitivity, multi-axis cost, and amortized wall-clock cost).

B.1 Per-graph headline detail

The next three tables source the main-text matched-memory result (Table 2) at a finer per-graph granularity. Table 6 reports the validation-split protocol on DIMACS NY/FLA and OSMnx Modena/Manhattan with a held-out test set. Table 7 is the per-budget summary across all five OSMnx city graphs. Table 8 is the full per-$(K_0, m, \text{B/v})$ configuration sweep on the same five graphs.

B.2 Hybrid and admissibility-stress robustness

This subsection documents the robustness checks that test what happens at the boundaries of the matched-memory protocol. Table 9 reports the matched-memory hybrid $\max(h_{\text{AAC}}, h_{\text{ALT}})$ on the non-road graphs (SBM, BA, OGB-arXiv) at the per-graph-type fair accounting of Section 5.1. Table 10 reports the doubled-memory pointwise-max hybrid on the directed road networks: a negative result at the same total budget, where pure ALT outperforms the AAC+ALT combination. Table 11 documents the admissibility verification

Table 6: MAIN **Validation-split protocol** addressing retrospective K_0-selection concerns. For each seed, 200 queries are split into a 100-query *validation* set (used to pick the best AAC K_0) and a disjoint 100-query *test* set (used for reporting). ALT has no hyperparameter to tune so val selection is vacuous. The **Test** columns are the reported numbers. The generalization gap (Val−Test) is small ($\leq 2.5\,\%$ in the mean), confirming the protocol is not query-selecting. At every matched budget, ALT equals or exceeds val-selected AAC on test, consistent with the retrospective Tables 14 and 7. 5 seeds per row; ALT std is zero because FPS is deterministic given the LCC-anchor start vertex.

Graph	Method	B/v	Val (%)	Test (%)	Gap (%)	K_0 picks
NY	AAC (val-sel.)	32	83.2 ± 2.2	81.3 ± 2.7	1.9	{32, 128}
	ALT	32	85.1	**85.7**	−0.6	—
	AAC (val-sel.)	64	89.6 ± 0.5	88.9 ± 1.2	0.7	{64, 128}
	ALT	64	91.2	**91.1**	0.1	—
	AAC (val-sel.)	128	92.9 ± 0.3	92.6 ± 0.3	0.3	{32, 128}
	ALT	128	93.7	**93.7**	0.0	—
FLA	AAC (val-sel.)	32	78.8 ± 2.1	76.3 ± 3.7	2.5	{32, 64, 128}
	ALT	32	85.7	**84.5**	1.1	—
	AAC (val-sel.)	64	86.7 ± 1.6	86.2 ± 1.3	0.5	{32, 64}
	ALT	64	88.5	**87.6**	0.8	—
	AAC (val-sel.)	128	92.4 ± 0.8	91.8 ± 0.7	0.6	{32, 64}
	ALT	128	92.8	**91.8**	1.1	—
Modena	AAC (val-sel.)	32	82.5 ± 1.5	81.3 ± 1.6	1.2	{32, 64, 128}
	ALT	32	83.4	**82.8**	0.5	—
	AAC (val-sel.)	64	88.5 ± 1.3	87.7 ± 1.5	0.8	{32, 64, 128}
	ALT	64	91.2	**89.3**	1.9	—
	AAC (val-sel.)	128	93.0 ± 0.7	$\mathbf{91.7 \pm 0.6}$	1.3	{32, 64}
	ALT	128	92.8	91.5	1.4	—
Manhattan	AAC (val-sel.)	32	81.2 ± 2.9	80.1 ± 3.5	1.1	{32, 64}
	ALT	32	83.9	**84.8**	−0.9	—
	AAC (val-sel.)	64	88.4 ± 1.1	87.6 ± 1.0	0.8	{32, 64, 128}
	ALT	64	90.4	**90.3**	0.2	—
	AAC (val-sel.)	128	92.3 ± 0.4	91.3 ± 0.4	1.0	{32, 64, 128}
	ALT	128	92.1	**91.7**	0.4	—

under early stopping, confirming that the architectural admissibility guarantee survives every checkpoint we tested.

B.3 Query-Distribution Sensitivity, Multi-Axis Cost, and Amortized Cost

This subsection consolidates the three robustness checks on cost: how the matched-memory comparison behaves under non-uniform query distributions (Table 12), how the three cost axes (online memory, offline preprocessing, query latency) decompose at the canonical B=64 B/v configuration (Table 13), and at what query workload the larger AAC offline cost is amortized by its lower per-query latency (Figure 8).

Query-distribution sensitivity. A natural hypothesis is that learned selection should outperform FPS when queries cluster in specific regions, concentrating landmarks where they are most needed. We test this with three query modes: *uniform* (i.i.d. from the largest connected component), *hotspot* (90% of endpoints drawn from a small high-degree cluster), and *powerlaw* (endpoints drawn proportional to degree raised to exponent 1.5). Table 12 reports results on NY (264K nodes, DIMACS) and Manhattan (4.6K nodes, OSMnx) at 64 B/v, 5 seeds, 100 queries per seed. ALT maintains its advantage under all three query distributions on both graphs. Under hotspot queries, AAC's seed variance increases substantially (NY: ±6.1% hotspot vs. ±1.8% uniform), indicating sensitivity to the interaction between learned landmark placement and query concentration. The Hybrid $\max(h_{\mathrm{AAC}}, h_{\mathrm{ALT}})$ is the top performer across all six distribution-graph combinations in Table 12, but at doubled memory (128 B/v vs. 64 B/v for the pure AAC and ALT columns); at matched memory, pure ALT at 128 B/v still leads (see Table 10). The cross-distribution pattern suggests

Table 7: ABLATION OSMnx road network results across scales: expansion reduction (%) at matched memory budgets. AAC rows show the best K_0 per budget (descriptive Pareto, same protocol as Table 14). Bold: best admissible method per budget per graph. All configurations verified at 0 admissibility violations. Mean ± std over 5 seeds (3 for Netherlands). *Cross-table note:* AAC cells use a per-budget best-K_0 retrospective protocol; values for the same nominal (K_0, m) may differ from Tables 21 and 22 by training stochasticity within the reported standard deviation (see Section 5.2, "Cross-table protocol").

Scale / Graph	Method	Nodes	32 B/v	64 B/v	128 B/v
Small-scale					
Modena	AAC	30K	80.7 ± 2.9	87.7 ± 1.2	91.9 ± 0.7
	ALT	30K	**83.4**	**91.2**	**92.8**
Manhattan	AAC	4.6K	79.2 ± 4.7	88.1 ± 0.6	$\mathbf{92.2 \pm 0.2}$
	ALT	4.6K	**83.9**	**90.4**	92.1
City-scale					
Berlin	AAC	28K	83.7 ± 2.6	91.3 ± 0.7	94.2 ± 0.4
	ALT	28K	**86.5**	**92.8**	**95.1**
Los Angeles	AAC	50K	79.3 ± 3.5	87.5 ± 2.1	92.1 ± 1.4
	ALT	50K	**85.4**	**91.1**	**94.2**
Country-scale					
Netherlands	AAC	4.5M	79.9 ± 1.7	87.3 ± 0.4	91.9 ± 1.4
	ALT	4.5M	**86.9**	**91.2**	**94.2**

Manhattan-128 AAC (92.2 %) is bolded only because AAC numerically exceeds ALT (92.1 %); this is a pool-size effect at $m \approx K_0/2$ (Section 3.3). Under the val-split protocol of Table 6 ALT leads at this cell (91.7 % vs. 91.3 %), so the crossover is retrospective best-K_0 only.

that learned and FPS-based landmarks capture complementary spatial structure, with the hybrid's stability advantage most useful as a variance-reduction wrapper around AAC rather than as a budget-for-budget replacement for ALT.

Multi-axis cost breakdown. "Matched memory" in this paper controls the deployed per-vertex *label* memory; the offline preprocessing cost and the per-query latency are not matched. Table 13 consolidates the three axes (online label memory, offline preprocessing time, p50/p95 query latency) on the four DIMACS graphs at the canonical B=64 B/v configuration, following the multi-axis cost-presentation pattern of recent learned-distance-index surveys (Choudhary et al., 2026). Online label memory is matched by construction; AAC pays 7–31× in additional offline preprocessing time (∼13–52 s for AAC vs. ∼1.4–5.0 s for ALT, the difference being K_0 SSSPs plus compressor training versus K SSSPs alone); but contrary to the naive "ALT is also faster online" story, median A* query latency at *matched memory* is in AAC's favor at p50 on every graph and at p95 on three of four (Table 3; the BAY-p95 cell is the sole exception where ALT is 1.16× faster). The compressed-label heuristic's contiguous 16-float-per-direction layout has better cache behavior than ALT's two 8-landmark tables, so AAC's lower expansion-count efficiency is partially compensated by lower per-expansion overhead. We use the descriptor "matched deployed label memory" rather than "matched memory" in the abstract and the captions of Tables 16 and 9 to keep this scope explicit; in body text we use "matched memory" as the short-hand once the convention is established.

Amortized wall-clock cost. A practitioner choosing between the two methods at a fixed memory budget must amortize the offline cost over the deployment's query workload N. Under the matched-memory timing of Table 13 (AAC K_0=64, m=16 vs. ALT K=8, both at 64 B/v), AAC's larger offline cost is amortized by its lower p50 query latency at *finite, modest* workloads on every DIMACS graph: breakeven $N \approx 1{,}924$ on NY, 1,276 on BAY, 170 on COL, and 522 on FLA (formula $N_{\text{break}} = (T_{\text{AAC,offline}} - T_{\text{ALT,offline}})/(T_{\text{ALT,p50}} - T_{\text{AAC,p50}})$). Under the matched protocol AAC is the cheaper total-wall-clock choice on every DIMACS graph above $\sim 10^3$ queries. Cost regimes below that (single-shot ad-hoc queries, small unit tests, and benchmark probes) still favor FPS-ALT.

Table 8: ABLATION Full per-configuration OSMnx results. Mean ± std over 5 seeds (3 for Netherlands). Bold: best reduction per (graph, memory budget). All configurations verified at 0 admissibility violations.

Graph	Method	K_0	m	B/v	Reduction (%)	Latency (ms)
Manhattan (4.6K)	AAC	16	8	32	79.2 ± 4.7	4
	AAC	32	8	32	77.2 ± 6.6	5
	AAC	32	16	64	88.1 ± 0.6	3
	AAC	64	16	64	87.4 ± 1.3	3
	AAC	64	32	128	$\mathbf{92.2 \pm 0.2}$	2
	ALT	–	–	32	**83.9**	3
	ALT	–	–	64	**90.4**	2
	ALT	–	–	128	92.1	2
	ALT	–	–	256	**93.5**	2
Modena (30K)	AAC	16	8	32	80.7 ± 2.9	26
	AAC	32	8	32	80.1 ± 4.7	27
	AAC	32	16	64	87.7 ± 1.2	18
	AAC	64	16	64	84.9 ± 1.8	22
	AAC	64	32	128	91.9 ± 0.7	13
	ALT	–	–	32	**83.4**	22
	ALT	–	–	64	**91.2**	13
	ALT	–	–	128	**92.8**	11
	ALT	–	–	256	**94.4**	9
Berlin (28K)	AAC	16	8	32	81.3 ± 5.2	33
	AAC	32	8	32	83.7 ± 2.6	36
	AAC	32	16	64	91.3 ± 0.7	21
	AAC	64	16	64	89.8 ± 0.9	22
	AAC	64	32	128	94.2 ± 0.4	17
	ALT	–	–	32	**86.5**	22
	ALT	–	–	64	**92.8**	17
	ALT	–	–	128	**95.1**	14
	ALT	–	–	256	**96.3**	11
Los Angeles (50K)	AAC	16	8	32	79.1 ± 4.7	77
	AAC	32	8	32	79.3 ± 3.5	75
	AAC	32	16	64	87.5 ± 2.1	46
	AAC	64	16	64	86.9 ± 0.9	47
	AAC	64	32	128	92.1 ± 1.4	38
	ALT	–	–	32	**85.4**	54
	ALT	–	–	64	**91.1**	39
	ALT	–	–	128	**94.2**	31
	ALT	–	–	256	**95.3**	27
Netherlands (4.5M)	AAC	16	8	32	79.9 ± 1.7	3652
	AAC	32	8	32	71.4 ± 3.9	5527
	AAC	32	16	64	87.3 ± 0.4	2393
	AAC	64	16	64	86.8 ± 0.7	2488
	AAC	64	32	128	91.9 ± 1.4	1560
	ALT	–	–	32	**86.9**	1324
	ALT	–	–	64	**91.2**	919
	ALT	–	–	128	**94.2**	680
	ALT	–	–	256	**95.9**	495

B.4 DIMACS retrospective and Pareto sweep

This subsection collects the retrospective best-K_0 tables for DIMACS that complement the val-split protocol of Table 6. Table 14 is the retrospective companion to the validation-split table: it shows the best-performing K_0 per budget for AAC, selected after looking at all 100 query results per cell. The unblinded upper envelope of the AAC curve serves two purposes: it exposes the maximum performance any test-time K_0 choice could deliver, and it sources the preprocessing numbers used by Table 13 and the latency table in the main text.

Table 9: **MAIN** Matched-memory hybrid comparison on non-road graphs under the corrected per-graph-type memory accounting of Section 5.2. All graphs are undirected: pure AAC uses $m{=}B/4$ with $K_0{=}4m$; pure ALT uses $K{=}B/4$; hybrid-half splits the budget equally ($m{=}B/8$ AAC + $K{=}B/8$ ALT, pointwise max). Greedy-Max is a query-adaptive covering oracle: greedily picks m landmarks from the same $K_0{=}4m$ FPS pool to maximize the average ALT bound over the test queries — not deployable (peeks at queries), serves as an upper bound on any selection-based non-learned FPS-pool strategy. The doubled-memory $\max(h_{\text{AAC}}, h_{\text{ALT}})$ runs both at deployment ($2B$ B/v total, 2× compute) and is therefore not a matched-budget arm. Mean expansion reduction vs. Dijkstra (%); 5 seeds × 100 queries per cell; ALT std is 0.00 (deterministic FPS). Discussion of the per-cell ordering and Greedy-Max headroom in Section 5.8.3.

Graph	B (B/v)	Dij. exp.	Pure AAC	Greedy-Max	Pure ALT	Hybrid-half	$\max(h_{\text{AAC}}, h_{\text{ALT}})$
SBM 10k	32	5,330	88.69 ± 0.54	90.80	**89.95**	88.64 ± 0.20	91.45 ± 0.19
	64	5,330	94.02 ± 0.48	95.04	**94.52**	93.57 ± 0.17	95.47 ± 0.16
	128	5,330	96.92 ± 0.12	97.45	**97.04**	96.69 ± 0.15	97.53 ± 0.09
BA 10k	32	4,764	88.56 ± 0.78	89.47	**89.90**	89.20 ± 0.18	91.10 ± 0.26
	64	4,764	93.54 ± 0.19	94.21	**94.25**	93.66 ± 0.17	94.90 ± 0.08
	128	4,764	96.56 ± 0.12	96.85	**96.93**	96.35 ± 0.07	97.28 ± 0.04
OGB-arXiv 169k	32	78,617	85.02 ± 1.42	88.69	**86.15**	84.42 ± 0.82	88.23 ± 0.50
	64	78,617	$\mathbf{92.54 \pm 0.54}$	93.17	91.33	90.92 ± 0.53	93.37 ± 0.37
	128	78,617	$\mathbf{96.03 \pm 0.20}$	96.66	95.16	95.03 ± 0.22	96.30 ± 0.14

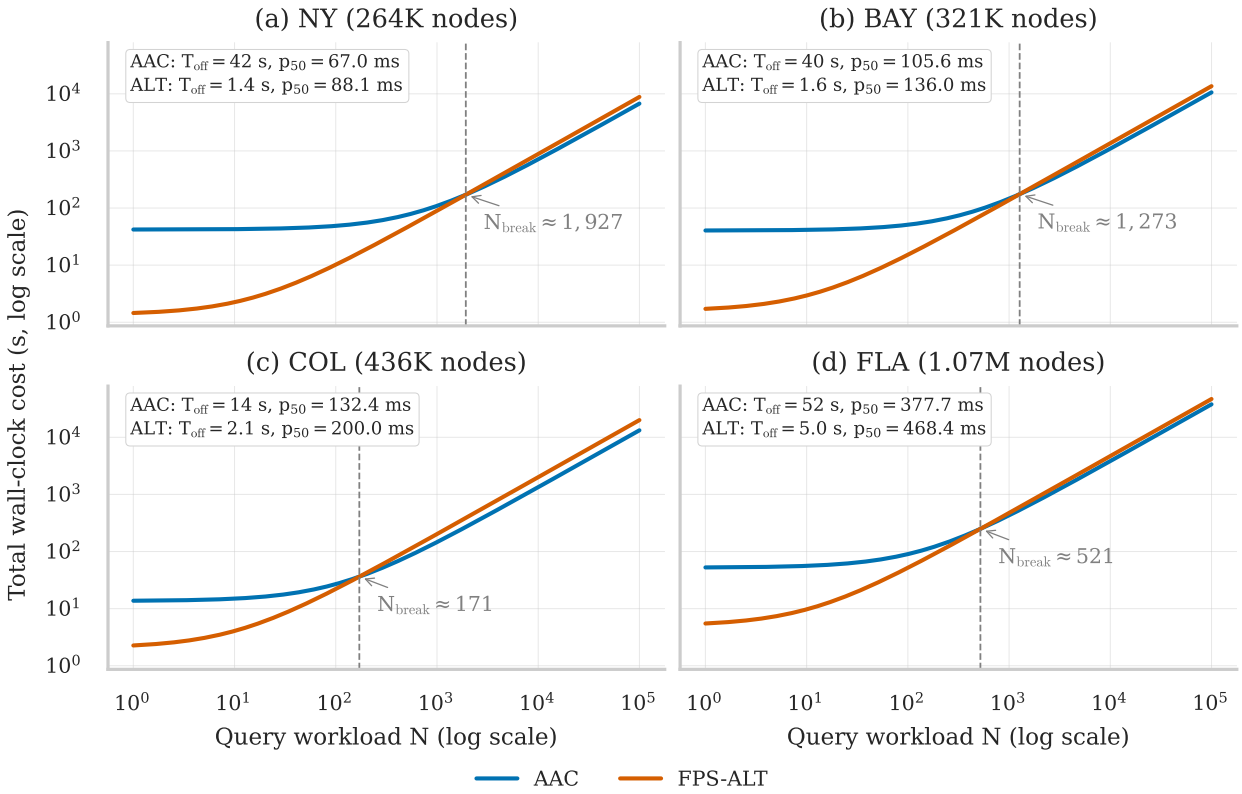

Figure 8: Total wall-clock cost vs. query workload N on DIMACS at $B{=}64$ B/v: $T_{\text{total}}(N) = T_{\text{off}} + N\,T_{\text{p50}}$ (both from Table 13). Vertical dashed line: breakeven N where AAC's offline cost is amortized by its lower p50 latency (≈ 1,924 NY, 1,276 BAY, 170 COL, 522 FLA). AAC dominates above breakeven; FPS-ALT below. Per-panel insets report the T_{off} and p_{50} values driving the curves.

Table 15 extends the same retrospective protocol to the full $K_0 \times m$ Pareto sweep on NY and FLA, exposing the trade-off curve at finer granularity. Table 16 reports per-cell combined Wilcoxon p-values across 5

Table 10: ABLATION Hybrid $\max(h_{\mathrm{AAC}}, h_{\mathrm{ALT}})$ evaluation: expansion reduction (%) on directed road networks. AAC is trained from a K_0 teacher pool (K_0=16 for m=8, K_0=32 for m=16, K_0=64 for m=32). The hybrid is admissible by construction (pointwise max of two admissible heuristics). Budget is the sum of AAC and ALT storage. Rows within each graph are sorted by total budget (B/v) to expose matched-budget comparisons. **Bold marks the best Exp. Red. per matched-budget group** (Std is reported for variance context only and is never bolded). At every matched budget, pure ALT outperforms the hybrid AAC+ALT combination: a clean negative result, where adding AAC to ALT at matched budget does *not* beat spending the same bytes on more ALT landmarks. Mean ± std over 5 seeds.

Graph	Method	B/v	Exp. Red. (%)	Std
Modena	AAC (m=8)	32	80.7	2.9
	ALT (K=4)	32	**83.4**	0.0
	AAC (m=16)	64	86.8	1.9
	AAC+ALT (m=8, K=4)	64	87.6	1.0
	ALT (K=8)	64	**91.2**	0.0
	AAC+ALT (m=16, K=4)	96	89.0	1.3
	ALT (K=12)	96	**92.5**	0.0
	AAC (m=32)	128	91.4	1.0
	ALT (K=16)	128	**92.8**	0.0
Manhattan	AAC (m=8)	32	79.2	4.7
	ALT (K=4)	32	**83.9**	0.0
	AAC (m=16)	64	86.8	1.7
	AAC+ALT (m=8, K=4)	64	87.8	1.4
	ALT (K=8)	64	**90.4**	0.0
	AAC+ALT (m=16, K=4)	96	90.7	0.8
	ALT (K=12)	96	**91.6**	0.0
	AAC (m=32)	128	91.6	1.2
	ALT (K=16)	128	**92.1**	0.0
NY	AAC (m=8)	32	78.4	5.5
	ALT (K=4)	32	**85.1**	0.0
	AAC (m=16)	64	89.2	0.6
	AAC+ALT (m=8, K=4)	64	89.2	1.0
	ALT (K=8)	64	**91.2**	0.0
	AAC+ALT (m=16, K=4)	96	91.3	0.6
	ALT (K=12)	96	**93.1**	0.0
	AAC (m=32)	128	92.8	0.5
	ALT (K=16)	128	**93.7**	0.0

Table 11: ABLATION **Admissibility under early stopping** (Modena, K_0=64, m=16). AAC is trained to epoch checkpoints $\{1, 5, 10, 50, 200\}$ and evaluated on 100 queries (5 seeds). Admissibility violations are counted by comparing $h(u, t)$ against Dijkstra-computed $d(u, t)$ for all query pairs. Zero violations are observed at *every* checkpoint, confirming the architectural guarantee empirically: even a 1-epoch model is provably safe. Performance is strong from epoch 1 (87.2% reduction), peaks at epoch 10 (88.0%), and remains stable through epoch 200 (86.2%).

Epochs	Exp. Red. (%) ↑	Violations	Status
1	87.2 ± 1.8	0	✓
5	84.8 ± 3.0	0	✓
10	88.0 ± 1.2	0	✓
50	85.5 ± 3.1	0	✓
200	86.2 ± 2.4	0	✓

seeds for the AAC vs. ALT matched-budget comparison on all four DIMACS road graphs, referenced from Section 5.3.

Table 12: ABLATION Expansion reduction (%) across query distribution modes on NY (264K nodes, DIMACS) and Manhattan (4.6K nodes, OSMnx) at 64 B/v. Hybrid uses 128 B/v (sum of AAC + ALT budgets). Mean ± std over 5 seeds, 100 queries per seed. ALT leads AAC on all modes; the Hybrid $\max(h_{\mathrm{AAC}}, h_{\mathrm{ALT}})$ is the top performer across all distributions *but at doubled memory (2B B/v)* — at matched memory (Hybrid 64 B/v vs. pure ALT 128 B/v), pure ALT still leads on the road graphs (see Table 10 for the matched-memory road-graph hybrid comparison; Table 9 covers the non-road analogue on SBM/BA/OGB-arXiv).

Graph	Mode	AAC	ALT	Hybrid
NY	Uniform	87.4 ± 1.8	91.6 ± 0.3	$\mathbf{93.4 \pm 0.3}$
	Hotspot	85.8 ± 6.1	90.8 ± 2.2	$\mathbf{92.8 \pm 1.8}$
	Powerlaw	88.5 ± 2.0	91.4 ± 0.4	$\mathbf{93.7 \pm 0.7}$
Manhattan	Uniform	87.1 ± 0.8	90.1 ± 0.5	$\mathbf{91.8 \pm 0.7}$
	Hotspot	87.5 ± 1.8	89.1 ± 2.6	$\mathbf{91.4 \pm 1.7}$
	Powerlaw	87.6 ± 2.2	90.5 ± 0.7	$\mathbf{92.1 \pm 0.9}$

Table 13: MAIN Multi-axis cost summary on DIMACS road networks, matched-deployed-label-memory configuration (B=64 B/v: AAC K_0=64, m=16; ALT K=8). Online memory is the deployed per-vertex label memory matched across methods. Offline preprocessing is K_0 Dijkstras plus compressor training for AAC, K Dijkstras for ALT. Query latency is from Table 3 (median over 100 queries, seed 42). Bold per (graph, axis) cell marks the cheaper method.

Graph	Method	Online (B/v)	Offline (s)	p50 query (ms)	p95 query (ms)
NY	AAC	64	42.0	**67.0**	**166.5**
	FPS-ALT	64	**1.4**	88.1	290.9
BAY	AAC	64	40.4	**105.6**	571.4
	FPS-ALT	64	**1.6**	136.0	**492.0**
COL	AAC	64	13.6	**132.4**	**549.9**
	FPS-ALT	64	**2.1**	200.0	814.8
FLA	AAC	64	52.3	**377.7**	**1210.5**
	FPS-ALT	64	**5.0**	468.4	1362.1

B.5 Mechanism diagnostics: ALT-pool, coverage-aware regularizer, training drift

This subsection collects the controlled diagnostics that isolate which mechanism inside the AAC training loop binds the matched-memory gap. The ALT-pool (first-m) arm tests whether the gap is architectural (it is not). The coverage-aware regularizer sweep tests whether the gap is closed by directly penalizing covering radius (partially, on roads only). The road-graph and hyperparameter robustness checks confirm that the training-drift pattern is not a synthetic-graph artifact and is stable under (learning rate, batch size, K_0) perturbations.

ALT-pool (first m) diagnostic. Table 17 reports the ALT-pool (first m) arm described in Section 5.9: FPS-ALT evaluated on the first m landmarks of the same K_0-landmark pool AAC is trained on, for seven graphs across both directed and undirected regimes. The arm equals pure FPS-ALT at matched memory on every graph and budget tested, confirming that the architecture admits the FPS-ALT subset and that any residual AAC gap is training-objective drift.

Coverage-aware regularizer sweep. If FPS's near-optimal coverage radius (Corollary 6) is the binding constraint on road networks, then directly penalizing the learner's covering radius during training is the cleanest causal test available. We add a soft covering-radius penalty $\lambda_{\mathrm{cov}} \cdot \tilde{r}$ to the standard distillation loss, where $\tilde{r}$ is a differentiable log-sum-exp approximation to the covering radius. Across two OSMnx graphs (Modena, Manhattan), five seeds, and $\lambda_{\mathrm{cov}} \in \{0, 10^{-3}, 10^{-2}, 10^{-1}\}$ with K_0=64, m=16 (64 B/v), the regularizer narrows the ALT gap by 1.3 ± 1.2 % on Modena (from 5.1 to 3.8 %) and 1.4 ± 2.2 % on Manhattan (from 4.4 to 3.1 %), while preserving admissibility in every run; the improvement plateaus at $\lambda_{\mathrm{cov}} \geq 10^{-3}$. This is a partial yes (directly optimizing covering radius does help), but the residual 3–4 % gap after coverage-aware regularization suggests the FPS candidate pool is the binding constraint on road

Table 14: ABLATION **Retrospective best-K_0 summary** (appendix companion to the validation-split protocol in Table 6). AAC vs. ALT at equal matched-deployed-label-memory budgets on DIMACS NY and FLA (5-seed means, 100 queries per graph). Best expansion reduction per (graph, memory) group in bold. AAC rows show the best-performing K_0 configuration per budget, selected *retrospectively* from the full Pareto sweep below; the validation-split version in the main text (Table 6) recovers the same ordering with a val–test gap of at most 2.5 %. A separate controlled Wilcoxon comparison (Section 5.3, Table 16) confirms ALT's advantage on all four DIMACS graphs.

Graph	Method	Bytes/V	Exp. Red. (%)	Mean Exp.	Preproc. (s)
NY	AAC	32	80.2 ± 3.6	24834	89.2
	ALT	32	**84.2**	19855	1.5
	AAC	64	88.3 ± 0.5	14677	108.4
	ALT	64	**91.0**	11299	2.2
	AAC	128	92.3 ± 0.3	9609	106.6
	ALT	128	**93.5**	8152	3.1
FLA	AAC	32	75.4 ± 2.4	129176	95.2
	ALT	32	**84.7**	80400	3.2
	AAC	64	85.7 ± 2.1	75242	64.6
	ALT	64	**87.7**	64543	4.9
	AAC	128	92.1 ± 1.1	41656	53.9
	ALT	128	**92.8**	37920	8.2

networks. Table 18 reports the same regularizer sweep on SBM and BA. On both synthetic graphs the regularizer is neutral to slightly negative: on SBM every non-zero λ_{cov} is statistically indistinguishable from the baseline ($p \gtrsim 0.4$), on BA the regularizer drifts the learner $\sim 0.6\,\%$ *below* the λ_{cov}=0 baseline. This is the pattern predicted by the covering-radius reading: small-diameter synthetic graphs leave little covering slack for the regularizer to close, and the road-network 1.3–1.4 % gain is the regime where the extra signal matters. We report it as *partial evidence for the covering-radius mechanism* rather than a general fix.

Training-drift diagnostic on a road graph. To check that the training-objective drift is not specific to non-road synthetic graphs, we re-run the forced-first-m vs. trained comparison on DIMACS NY (264K nodes, directed, K_{ALT}=8, matched memory B=64 B/v) with three seeds (42, 123, 456) and the same five training-epoch checkpoints used in the SBM/BA experiment of Section 5.9. Table 19 reports the resulting drift: at the architectural ceiling (91.20%, equal to FPS-ALT K=8 on this cell), the trained selector stays 1.7–3.3 % below across the full 0–1000-epoch window, and the gap *widens monotonically with more training* from −1.67 % at epoch 50 to −3.28 % at epoch 1000.

Hyperparameter and initialization robustness for training drift. Table 20 reports a hyperparameter-and-initialization sweep on the SBM B=32 cell described at the end of Section 5.9. The take-away is that the training drift documented above is robust across the (learning rate, batch size, K_0) hyperparameter grid we test, but vanishes when the compressor is initialized at the architectural ceiling (*identity-on-first-m* init), so the binding constraint is the default block-sparse initialization rather than a fundamental optimization barrier on this cell.

C Ablation Studies

We conduct three targeted ablation experiments on the OSMnx city graphs (Modena and Manhattan) to isolate the contributions of AAC's key design choices. All experiments use K_0=64 teacher landmarks, seeds $\{42, 123, 456, 789, 1024\}$, and 100 random query pairs per configuration.

Selection Strategy

To understand *why* ALT leads, Table 21 compares five selection strategies at matched memory (64 B/v, m=16): (i) AAC (default) (trained row-stochastic selection from K_0=64 teachers), (ii) Random-Subset

Table 15: ABLATION Full Pareto sweep detail for admissible methods (5-seed means ± std, 100 queries per graph). Best expansion reduction per (graph, memory budget) in bold. FastMap is excluded (inadmissible; see Section 5.3.1 for its 93–97% expansion reduction with 100% suboptimal paths).

Graph	Method	Config	Bytes/V	Reduction (%)
NY	AAC	K_0=32, m=8	32	80.2 ± 3.6
	AAC	K_0=32, m=16	64	88.3 ± 0.5
	AAC	K_0=32, m=32	128	92.3 ± 0.3
	AAC	K_0=64, m=8	32	77.3 ± 4.5
	AAC	K_0=64, m=16	64	87.8 ± 1.9
	AAC	K_0=64, m=32	128	91.8 ± 0.5
	AAC	K_0=64, m=64	256	94.9 ± 0.2
	AAC	K_0=128, m=8	32	73.9 ± 7.9
	AAC	K_0=128, m=16	64	87.6 ± 0.7
	AAC	K_0=128, m=32	128	91.6 ± 0.9
	AAC	K_0=128, m=64	256	94.4 ± 0.5
	ALT	K=4	32	**84.2**
	ALT	K=8	64	**91.0**
	ALT	K=16	128	**93.5**
	ALT	K=32	256	**95.0**
FLA	AAC	K_0=32, m=8	32	74.5 ± 3.9
	AAC	K_0=32, m=16	64	85.7 ± 2.1
	AAC	K_0=32, m=32	128	92.1 ± 1.1
	AAC	K_0=64, m=8	32	73.8 ± 2.0
	AAC	K_0=64, m=16	64	85.3 ± 2.1
	AAC	K_0=64, m=32	128	90.7 ± 0.7
	AAC	K_0=64, m=64	256	93.9 ± 0.9
	AAC	K_0=128, m=8	32	75.4 ± 2.4
	AAC	K_0=128, m=16	64	81.9 ± 3.4
	AAC	K_0=128, m=32	128	90.6 ± 1.0
	AAC	K_0=128, m=64	256	92.9 ± 0.6
	ALT	K=4	32	**84.7**
	ALT	K=8	64	**87.7**
	ALT	K=16	128	**92.8**
	ALT	K=32	256	**94.6**

(random m landmarks from the same K_0 pool), (iii) FPS-Subset (standard ALT with K=8 FPS-selected landmarks, deterministic seed vertex from the largest SCC), (iv) FPS-RR$_{10}$ (random-restart FPS: 10 random seed vertices, best chosen on a held-out 100-query validation split, reported on a disjoint 100-query test split), and (v) Greedy-Max (greedily selects landmarks from the K_0 pool maximising average heuristic over the evaluation queries; a query-adaptive oracle upper bound). FPS-Subset outperforms both pool-based learning methods on both graphs (Modena: 91.2% vs. 88.2% and 86.7%; Manhattan: 90.4% vs. 87.6% and 86.7%). Random-restart FPS does not improve on the canonical FPS-Subset: on both graphs, the validation-best of 10 random seed vertices scores 89.2% on test, 1–2 % *below* the canonical seed-vertex FPS, ruling out a "FPS is just lucky" explanation. Greedy-Max slightly outperforms FPS on both graphs (91.5% vs. 91.2% on Modena; 91.5% vs. 90.4% on Manhattan), showing that query-adaptive selection from the teacher pool *can* beat FPS, but even with oracle access the gain is modest (+0.3–1.1 %). Main-text discussion is in Section 5.7.

Admissibility Under Early Stopping

See Section 5.6 in the main text (Table 11).

Compression Efficiency Curve

Table 22 sweeps $m \in \{4, 8, 16, 32, 64\}$ at fixed K_0=64, comparing AAC vs. ALT at matched memory ($2m$ floats per direction). ALT outperforms AAC at high compression ratios; the gap narrows and reverses at

Table 16: MAIN Per-cell paired Wilcoxon p-values on DIMACS at matched deployed label memory (AAC vs. FPS-ALT, two-sided signed-rank, 100 queries × 5 seeds). We report both Fisher's method and Stouffer's z-score combination across seeds, plus the median per-seed p for reference. Every cell is significant under Benjamini–Hochberg FDR correction at q=0.05 under both combinators; the worst Stouffer-combined cell is COL-128 at p=1.3×10^{-3}, and the worst Fisher-combined cell is FLA-64 at p=1.6×10^{-2} (both still FDR-significant). Caveats in Section 5.3.

Graph	Budget (B/v)	Fisher p	Stouffer p	Median per-seed p	FDR q=0.05
NY	32	2.2×10^{-17}	5.9×10^{-20}	1.3×10^{-5}	✓
	64	7.8×10^{-13}	4.6×10^{-14}	2.0×10^{-3}	✓
	128	8.3×10^{-15}	8.1×10^{-17}	7.3×10^{-5}	✓
BAY	32	2.0×10^{-25}	1.7×10^{-26}	1.4×10^{-7}	✓
	64	3.0×10^{-9}	1.2×10^{-7}	8.2×10^{-2}	✓
	128	6.2×10^{-5}	2.7×10^{-4}	4.0×10^{-1}	✓
COL	32	8.6×10^{-12}	1.2×10^{-9}	1.0×10^{-1}	✓
	64	4.8×10^{-5}	2.6×10^{-5}	1.9×10^{-1}	✓
	128	1.1×10^{-2}	1.3×10^{-3}	2.3×10^{-1}	✓ (worst Stouffer)
FLA	32	6.0×10^{-24}	3.9×10^{-25}	1.8×10^{-6}	✓
	64	1.6×10^{-2}	5.5×10^{-4}	1.4×10^{-1}	✓ (worst Fisher)
	128	6.0×10^{-8}	5.3×10^{-9}	2.4×10^{-2}	✓

low compression. The gap is largest at high compression (m=4–8) where AAC is 12.7 % behind ALT on Modena, and narrows as m approaches K_0: at m=32 and m=64 on Manhattan, AAC slightly surpasses ALT (92.2% vs. 92.1% at m=32; 93.7% vs. 93.5% at m=64). This suggests that AAC's learned compression recovers ALT-level performance given sufficient capacity, and can marginally outperform FPS at large m.

Reconciling with the capacity ceiling. These crossover results do not contradict Proposition 2. The capacity ceiling says the learner cannot exceed the *teacher with K_0 landmarks* (FPS-ALT at K=K_0). It does *not* say AAC with m dimensions cannot surpass FPS-ALT at K=$m/2$, which is the matched-memory baseline in the table. At m=64 from K_0=64, AAC selects 64 landmarks from the 64-teacher pool (effectively identity, modulo selection order), matching the teacher. But Table 22 compares against FPS-ALT at K=32 (matched memory: $2 \times 32 = 64$ floats/direction, same as m=64 for AAC). Since FPS-at-32 and best-subset-of-64-of-size-32 are different selection procedures on different-sized pools, the crossover reflects the advantage of pool access rather than a violation of the capacity ceiling. At m=K_0, AAC recovers the teacher exactly (Proposition 4), while the matched-budget ALT baseline uses K=$K_0/2$, strictly fewer landmarks from fresh FPS. The proper three-way decomposition (FPS-at-K_0 full teacher vs. AAC-at-m vs. FPS-at-K=$m/2$ matched budget) is presented in Section 5.7.

D FastMap: Extended Out-of-Domain Analysis

FastMap (Cohen et al., 2018) was designed and its admissibility guarantee proven for undirected graphs. We include it on the directed DIMACS benchmarks as a cautionary out-of-domain example, not as a critique of the method itself. FastMap's L1 embedding heuristic achieves the highest raw expansion reduction (93–97%) on DIMACS graphs. However, our empirical verification across all four tested dimensions $d \in \{8, 16, 32, 64\}$ reveals that FastMap is inadmissible on these directed benchmarks: on all 800 queries across NY and FLA (100 queries × 2 graphs × 4 dimensions), **100% of FastMap-guided A* paths are suboptimal**. The mean cost ratio (FastMap path cost / optimal cost) ranges from 1.15 to 1.22, with a maximum of 1.95.

This finding has a methodological implication: *expansion reduction alone is a misleading metric when comparing admissible and inadmissible methods.* An inadmissible heuristic can achieve arbitrarily high expansion reduction by aggressively overestimating, pruning the search tree at the cost of solution quality. Expansion reduction alone should not be compared across admissibility classes without also reporting solution quality.

Table 17: ABLATION **ALT-pool (first m) diagnostic**: ALT run on the first m landmarks of the *same* K_0-landmark FPS teacher pool that AAC is trained on, vs. canonical FPS-ALT at matched memory (ALT $K{=}m$ on undirected graphs, $K{=}m/2$ on directed graphs). The two arms agree exactly to two decimal places on all seven graphs and three budgets tested, because FPS is greedy-incremental: the first K landmarks of a K_0-step FPS run *are* the K-step FPS landmarks. The arm therefore makes explicit that the FPS-ALT matched-memory subset is inside the row-stochastic admissibility set AAC searches: any residual AAC-vs-ALT gap is training-objective drift (Section 5.9), not architectural capacity. 5 seeds × 200 queries per cell aggregated full-CSV; ALT has zero variance across seeds (deterministic FPS given the LCC seed). Numbers may differ from Table 6's val-split TEST column (5 seeds × 100 disjoint test queries) by sampling variance within the val-split standard deviations — the protocols agree on the ALT–AAC ordering at every cell.

Graph	**Type**	B **(B/v)**	**ALT-pool first-m red. (%)**	**FPS-ALT matched red. (%)**
		32	89.95	89.95
SBM (5×2000)	undirected	64	94.52	94.52
		128	97.04	97.04
		32	89.90	89.90
BA (10,000, m=5)	undirected	64	94.25	94.25
		128	96.93	96.93
		32	85.11	85.11
NY (DIMACS)	directed	64	91.20	91.20
		128	93.73	93.73
		32	83.38	83.38
Modena (OSMnx)	directed	64	91.20	91.20
		128	92.85	92.85
		32	83.86	83.86
Manhattan (OSMnx)	directed	64	90.44	90.44
		128	92.13	92.13
		32	86.46	86.46
Berlin (OSMnx)	directed	64	92.77	92.77
		128	95.14	95.14
		32	85.41	85.41
Los Angeles (OSMnx)	directed	64	91.11	91.11
		128	94.20	94.20

E Pre-Registration Protocol and Accounting Conventions

Outcome. We falsified the main claim of the pre-registration: the predicted bands of +2 to +6 % are not survived by the observed +0.87 to +1.21 % effect at $B \geq 64$, and the predicted direction is reversed at $B = 32$ where ALT leads by 1.13 %. The convention-independent sub-prediction (zero admissibility violations) holds in 15/15 cells, both by architectural guarantee (Proposition 2) and by empirical verification across the full 5-seed × 3-budget audit (1,500 queries total: 0 violations and $\max d_{\mathrm{AAC}}/d_{\mathrm{Dijkstra}} = 1.000000000$ on every per-cell record). The directionality-aware accounting convention used throughout this paper (Section 5.2, restated in E.2 below) is the methodological by-product that motivated the apparatus around the pre-registration.

E.1 Pre-Registration Protocol and Timestamps

B (B/v)	predicted gap (AAC−ALT)	observed gap (AAC−ALT)	TOST δ=1 %	verdict
32	+2 to +6 %	−1.13 % (ALT ahead)	rejected	**direction inverted**
64	+1.5 to +4.5 %	$+1.21 \pm 0.54$ %	rejected	magnitude 4× low
128	+1 to +3 %	$+0.87 \pm 0.20$ %	rejected	magnitude 2–3× low

Pre-registration protocol. Before running any evaluation on OGB-arXiv (Hu et al., 2020), we filed a timestamped prediction (archived in the released artifact at `results/PREREG.md`; see Appendix G). The prediction was written based on the covering-radius theory (Theorem 5) and the synthetic-graph findings

Table 18: ABLATION **Coverage-aware regularizer on non-road graphs.** AAC trained with the differentiable covering-radius regularizer R_{cov} at four strengths $\lambda_{\text{cov}} \in \{0, 10^{-3}, 10^{-2}, 10^{-1}\}$, evaluated against the matched-memory FPS-ALT reference at the same deployed label memory. Setup: K_0=32, m=8, 200 training epochs, batch size 256, learning rate 10^{-3}, 100 queries per seed. SBM results from 5 seeds; BA results from 3 completed seeds. The Wilcoxon p-value column is the paired test of AAC expansions at that λ_{cov} against the same-seed λ_{cov}=0 baseline, with p-values combined across seeds (Fisher). On non-road synthetic graphs the regularizer fails to close the ALT gap: on SBM it is statistically indistinguishable from λ_{cov}=0 ($p \gtrsim 0.4$); on BA it is neutral-to-slightly-negative. This is consistent with the covering-radius reading that, on small-diameter synthetic graphs, FPS is already near-optimal and there is little covering slack for the regularizer to close. The road-network coverage-aware evaluation in Appendix B (Modena/Manhattan) closed a 1.3–1.4 % gap, so the non-road null finding here bounds the regularizer's effective scope rather than overturning it.

Graph	λ_{cov}	AAC red. (%)	ALT red. (%)	Wilcoxon p vs. baseline
SBM (5×2000)	0	88.69	89.95	1.00 (baseline)
	10^{-3}	88.78	89.95	≈ 0.41
	10^{-2}	88.68	89.95	≈ 0.45
	10^{-1}	88.65	89.95	≈ 0.47
BA (10,000, m=5)	0	89.03	89.90	1.00 (baseline)
	10^{-3}	88.43	89.90	$< 10^{-3}$ (worse)
	10^{-2}	88.43	89.90	$< 10^{-3}$ (worse)
	10^{-1}	88.43	89.90	$< 10^{-3}$ (worse)
OGB-arXiv (∼170K nodes, symmetrized LCC): deferred; see Appendix B for discussion.				

Table 19: ABLATION **Training-objective drift on a road graph (DIMACS NY).** Extension of the Section 5.9 diagnostic to a directed road network (K_{ALT}=8, matched memory B=64 B/v: AAC K_0=32, m_{fwd}+m_{bwd}=8+8; ALT K=8). Per-cell mean ± std over 3 seeds (42, 123, 456) of test-set expansion reduction at five training-epoch checkpoints; 100 queries per seed. The **forced-first-m** arm (AAC with one-hot rows on the first K_{ALT} pool indices in each direction) algebraically equals FPS-ALT K=8 to rounding (91.20%); the **trained** compressor stays 1.7–3.3 % *below* this architectural ceiling across the entire 0–1000-epoch window, with the gap *widening* from −1.67 % at epoch 50 to −3.28 % at epoch 1000 — the multi-graph SBM/BA pattern of Figure 7 reproduced on a real road graph.

Variant	ep. 0	ep. 50	ep. 200	ep. 500	ep. 1000
AAC (trained, default; mean over 3 seeds)	89.13 ± 0.62	89.53 ± 0.24	88.82 ± 0.04	88.36 ± 1.08	87.92 ± 1.53
forced-first-m (arch. ceiling)	91.20	91.20	91.20	91.20	91.20
FPS-ALT K=8 (matched-memory reference)	91.20	91.20	91.20	91.20	91.20
gap (AAC − ceiling, %)	−2.07	−1.67	−2.38	−2.84	−3.28

(Section 5.8), and reads, verbatim: *on OGB-arXiv (citation graph, ∼170K nodes, non-metric, high clustering), AAC should beat ALT at every budget level with a gap of roughly +2 to +6 percentage points at 32 B/v narrowing to +1 to +3 % at 128 B/v; the hybrid* $\max(h_{AAC}, h_{ALT})$ *should be the top performer; admissibility violations should be zero.* We then ran the evaluation under the convention of E.2 below and report the outcome in the table at the top of this section, whether the prediction holds or fails.

Setup. OGB-arXiv is a directed citation graph (169,343 papers, 1,166,243 edges). We symmetrize (undirected), take the largest connected component (no change; the graph is already connected), and assign uniform-random edge weights in $[1, 10]$ to obtain a weighted undirected graph suitable for A* benchmarking. We run 100 uniform-random s-t queries per seed, 5 seeds, at budgets $B \in \{32, 64, 128\}$ per vertex (AAC $K_0/m \in \{32/8, 64/16, 128/32\}$; ALT $K \in \{4, 8, 16\}$). A matched-total-budget hybrid companion study (Section 5.8.3) evaluates $\max(h_{\text{AAC}}, h_{\text{ALT}})$ on OGB-arXiv, BA, and SBM under a half/half budget split (each arm at $B/2$ B/v). Reference oracle: Dijkstra (mean 78,616 expansions).

Table 20: ABLATION **Hyperparameter and initialization sweep on the SBM B=32 training-drift cell.** Each row aggregates 3 seeds (mean ± std) at one (learning rate, batch size, K_0, init) cell, 200 training epochs throughout. "Trained gap" is AAC (trained) − forced-first-m baseline (architectural ceiling 89.95% at K_0=32, m=8) in percentage points. *block_sparse* is the default block-sparse initialization used throughout the paper; *identity_first_m* is a soft one-hot init on the first-m pool indices that starts at the ceiling. Across the entire (lr, batch) grid the default-init drift is −0.5 to −1.5 %; under the identity init the trained selector *stays at the architectural ceiling* (+0.00 % std 0.00) at every cell. The table below shows the K_0=32 subset of a 36-cell sweep (3 learning rates × 3 batch sizes × 2 K_0s × 2 inits); the K_0=64 subset reproduces the same qualitative pattern.

lr	batch	init	trained red. (%)	trained gap (%)
3×10^{-4}	128	block_sparse	88.93 ± 0.59	−1.02
3×10^{-4}	128	identity_first_m	89.95	0.00
3×10^{-4}	256	block_sparse	89.24 ± 1.02	−0.71
3×10^{-4}	256	identity_first_m	89.95	0.00
3×10^{-4}	512	block_sparse	88.93 ± 0.59	−1.02
3×10^{-4}	512	identity_first_m	89.95	0.00
1×10^{-3}	128	block_sparse	89.11 ± 0.28	−0.84
1×10^{-3}	128	identity_first_m	89.95	0.00
1×10^{-3}	256	block_sparse	89.46 ± 0.71	−0.49
1×10^{-3}	256	identity_first_m	89.95	0.00
1×10^{-3}	512	block_sparse	89.45 ± 0.41	−0.50
1×10^{-3}	512	identity_first_m	89.95	0.00

Table 21: ABLATION Selection strategy ablation (K_0=64, m=16, 64 B/v; see Section 5.1). AAC (default): trained row-stochastic selection. Random-Subset: random m landmarks from the K_0 teacher pool. FPS-Subset: standard ALT with K=$m/2$=8 FPS-selected landmarks (no teacher pool, deterministic seed vertex from the largest SCC). FPS-RR$_{10}$: random-restart FPS, R=10 random seed vertices, best on a held-out 100-query validation split (test on disjoint 100-query split). Greedy-Max: greedily selects landmarks from the K_0 pool maximizing average ALT heuristic over the evaluation queries (query-adaptive oracle; see text). Mean ± std over 5 seeds, 100 queries. *Cross-table note:* this ablation uses an independent retraining protocol; values for the same nominal (K_0, m) may differ from Tables 22 and 7 by training stochasticity within the reported standard deviation.

Graph	Method	Exp. Red. (%) ↑	Std
Modena	AAC (default)	88.2	1.4
	FPS-Subset (ALT K=8, canonical seed)	91.2	0.0
	FPS-RR$_{10}$ (val-selected best of 10 restarts)	89.2	0.2
	Greedy-Max†	**91.5**	0.0
	Random-Subset	86.7	0.7
Manhattan	AAC (default)	87.6	0.9
	FPS-Subset (ALT K=8, canonical seed)	90.4	0.0
	FPS-RR$_{10}$ (val-selected best of 10 restarts)	89.2	0.5
	Greedy-Max†	**91.5**	0.0
	Random-Subset	86.7	2.1

† Query-adaptive oracle: selects from K_0 pool using evaluation queries.

E.2 Directionality-Aware Accounting Convention

Convention. Matched memory at B bytes per vertex matches the per-vertex deployed-label storage of the two arms. On directed graphs ALT stores both forward and backward landmark distances ($2K$ floats per vertex), so matched memory at B B/v is AAC m=$B/4$ vs. ALT K=$B/8$. On undirected graphs ALT stores a single direction (K floats per vertex), so matched memory at B B/v is AAC m=$B/4$ vs. ALT K=$B/4$. This is the convention used everywhere in the paper; the OGB-arXiv outcome reported in E.1 above and summarised in Table 9 uses the undirected rule (5 seeds, 100 queries per seed). One methodological lesson

Table 22: ABLATION **Compression efficiency curve** (K_0=64, $m \in \{4, 8, 16, 32, 64\}$). AAC (compression from K_0=64 teachers to m dimensions) vs. ALT (K=$m/2$ FPS landmarks) at matched memory budgets. Mean ± std over 5 seeds, 100 queries. ALT outperforms AAC at high compression (m=4–16); the gap narrows as m approaches K_0 and reverses slightly on Manhattan at m=32 and m=64. Note: at m=64 (m=K_0), this is the non-compression regime where AAC effectively recovers the full teacher (Proposition 4). *Cross-table note:* this ablation uses an independent retraining protocol; values for the same nominal (K_0, m) may differ from Tables 21 and 7 by training stochasticity within the reported standard deviation (see Section 5.2, "Cross-table protocol").

(a) Modena (30K nodes)

m	B/v	AAC (%)	ALT (%)
4	16	59.1±12.6	**71.8**
8	32	76.3 ± 5.3	**83.4**
16	64	84.9 ± 1.8	**91.2**
32	128	91.9 ± 0.7	**92.8**
64	256	94.2 ± 0.3	**94.4**

(b) Manhattan (4.6K nodes)

m	B/v	AAC (%)	ALT (%)
4	16	64.4 ± 9.7	**72.8**
8	32	75.0 ± 8.8	**83.9**
16	64	87.4 ± 1.3	**90.4**
32	128	**92.2 ± 0.2**	92.1
64	256	**93.7 ± 0.2**	93.5

generalizes: pre-registration is necessary but not sufficient for matched-memory comparisons, because the accounting convention itself can swing the headline magnitudes; both the prediction and the convention must be fixed together before evaluation. The pre-registered protocol and bands are archived alongside the artifact (Appendix G).

F Contextual Variant: Pipeline Specification and Warcraft Compatibility Study

This appendix collects (i) the formal differentiable training-proxy specification that Section 3.4 points to and (ii) the Warcraft 12×12 compatibility case study referenced from Section 5.10. None of these details are needed to read or reproduce the static experiments (Sections 5.3–5.9).

Pipeline Specification

Pipeline. The pipeline composes as follows:

1. An encoder f_ϕ (CNN or ResNet) predicts cell costs $\hat{c} = f_\phi(\text{context})$ from input features.
2. Cell costs are converted to edge weights: $w(u, v) = \frac{1}{2}(\hat{c}_u + \hat{c}_v) \cdot \gamma_{uv}$, where γ_{uv} is the Euclidean distance factor (1 for cardinal, $\sqrt{2}$ for diagonal neighbors).
3. A differentiable Bellman–Ford solver computes soft SSSP distances from K anchors.
4. During training, the row-stochastic compressor operates on soft anchor distances as a differentiable proxy. At deployment, exact anchor distances are recomputed on the predicted-cost graph, and the resulting compressed heuristic is admissible for A* on that graph by Proposition 2.

Smooth Bellman–Ford. Standard Bellman–Ford iterates $d(v) \leftarrow \min\big(d(v), \min_{(u,v)\in E} d(u) + w(u, v)\big)$. We replace the hard min with a differentiable smooth-min:

$$\text{smoothmin}_\beta(x_1, \ldots, x_n) = -\frac{1}{\beta} \log \sum_{i=1}^{n} \exp(-\beta x_i). \tag{12}$$

Proposition 10 (Smooth-min is a lower bound on hard min)**.** $\text{smoothmin}_\beta(\mathbf{x}) \leq \min(\mathbf{x})$ *for all* $\mathbf{x} \in \mathbb{R}^n$ *and* $\beta > 0$*, with equality as* $\beta \to \infty$*.*

Proof. $\log \sum_i \exp(-\beta x_i) \geq \log \exp(-\beta \min_i x_i) = -\beta \min_i x_i$. Multiplying by $-1/\beta$ (flipping the inequality): $-\frac{1}{\beta} \log \sum_i \exp(-\beta x_i) \leq \min_i x_i$. For convergence: $\log \sum_i \exp(-\beta x_i) \leq -\beta \min_i x_i + \log n$, so $\text{smoothmin}_\beta(\mathbf{x}) \geq \min(\mathbf{x}) - \frac{\log n}{\beta} \to \min(\mathbf{x})$. □

Admissibility at deployment vs. during training. Smooth-min underestimates the hard min at every Bellman–Ford iteration, so the soft per-vertex distances satisfy $\tilde{d}(l_k, v) \leq d(l_k, v)$. However, the ALT heuristic uses *differences* of the form $d(l_k, t) - d(l_k, u)$, and underestimating both terms individually does not preserve a lower bound on their difference. Therefore, the smooth teacher heuristic used during training is *not* guaranteed to be admissible with respect to true distances. At *deployment*, this issue does not arise: exact distances are computed on the predicted-cost graph via standard Dijkstra, and the row-stochastic compression of these exact distances is admissible by Proposition 2. The smooth pipeline serves only as a differentiable training proxy; admissibility is guaranteed architecturally at inference, not through the training dynamics. Temperature β is annealed from 1.0 to 30.0 during training, implementing a smooth-to-hard curriculum that improves gradient flow in early training while approaching the exact solution at convergence.

Combined loss.

$$\mathcal{L}_{\text{ctx}} = \underbrace{(d_{\text{true}} - \tilde{h}_A)_+}_{\text{gap-closing}} + \alpha_{\text{cost}} \cdot \underbrace{\|\hat{c} - c_{\text{gt}}\|^2}_{\text{cost supervision}} + \lambda_{\text{cond}} \cdot R_{\text{ent}}, \tag{13}$$

where d_{true} denotes the ground-truth shortest-path distance on the true-cost graph (computed offline via Dijkstra), and $\tilde{h}_A$ is the smooth compressed heuristic from the predicted-cost graph. The gap-closing term is a *cross-graph path-quality surrogate*: d_{true} and $\tilde{h}_A$ live on different graphs, so the gap is not an admissibility bound in the classical sense. Admissibility at deployment is guaranteed only with respect to the predicted-cost graph (Proposition 2). Cost supervision provides direct signal for the encoder; the gap-closing term provides signal for both encoder and compressor.

Encoder architectures. We evaluate two encoders on 96×96 RGB terrain images:

- **CNN**: 3 convolutional layers (3→32→64→128 channels), AdaptiveAvgPool to grid size, 1×1 conv to scalar costs, $\text{softplus}(\cdot) + 0.01$ for strict positivity.
- **ResNet**: 5×5 stem with batch normalization, 3 residual blocks (64 channels each), AdaptiveAvgPool, two 1×1 convolutions, $\text{softplus}(\cdot) + 0.01$.

Warcraft Compatibility Study

We report a *compatibility* case study (demonstrating that AAC integrates into a differentiable planning pipeline with admissibility guaranteed at deployment) rather than claiming synergy or competitive advantage over specialized differentiable planners.

Setup. We evaluate on Warcraft 12×12 grid maps (Vlastelica et al., 2020): 10,000 train / 1,000 test instances, 144 nodes, 8-connected. RGB terrain images (96×96) serve as context; a CNN or ResNet encoder predicts edge costs, then the AAC compressor constructs admissible landmarks for A* search on the predicted-cost graph.

Table 23 compares AAC contextual variants against two differentiable baselines. The comparison regime is heterogeneous: AAC results are 3-seed means, BB uses published numbers from Vlastelica et al. (2020), and DataSP is our reproduction with a single seed. We do not claim competitive performance and we omit the cost-regret column from Table 23 because cost regret on dense 12×12 grids is not commensurable across methods with such different path-matching propensities. The numerical observation is narrowly that AAC with a ResNet encoder and cost supervision (α_{cost}=10) trains end-to-end to completion: the compressor stays frozen, yields zero admissibility violations on all 1000 test instances, and the surrounding encoder learns. The blackbox method of Vlastelica et al. (2020) remains substantially better on path-match accuracy (54.7% vs. $20.3 \pm 2.4\%$), likely because it differentiates through a full combinatorial shortest-path solution (Dijkstra in the forward pass), whereas AAC trains through a smooth label-construction surrogate built from soft shortest-path distances. We report these numbers solely to establish that the end-to-end differentiable pipeline trains to completion, not to argue AAC is preferable to blackbox differentiation on grid-map planning.

Table 23: ABLATION Compatibility study on Warcraft 12×12 grids: AAC integrates into a differentiable planning pipeline and retains deployment-time architectural admissibility on the predicted-cost graph. The table reports (a) whether the pipeline trains end-to-end and (b) the resulting path-match accuracy. The cost-regret column is omitted because the metric is not commensurable across methods with very different path-matching propensities (on 12×12 grids many paths have near-identical cost, so a method that produces *entirely different* paths can still hit a near-zero cost regret while having a near-zero path match); cost-regret numbers remain in the released artifact. AAC entries are mean ± std over 3 seeds; Warcraft 12×12 (10,000 train / 1,000 test). At deployment, AAC requires per-instance anchor SSSP on the predicted-cost graph followed by $O(m)$ per-query heuristic evaluation; BB uses Dijkstra-based combinatorial differentiation; DataSP uses $O(V^3)$ Floyd-Warshall. Admissibility column reports the empirical violation rate over the 1,000 test instances. The comparison is not parity-style and is included for context only — the baselines do not provide admissibility guarantees.

Method	**Configuration**	**Path Match ↑**	**Jaccard ↑**	**Adm. violations ↓**
AAC (admissible on predicted-cost graph; $O(m)$ query after per-instance label construction)				
ResNet, α_c=10	encoder only (frozen comp.)	$20.3\% \pm 2.4$ **	$59.3\% \pm 3.9$ **	0/1000
CNN, α_c=1	encoder only (frozen comp.)	$0.4\% \pm 0.1$	$20.9\% \pm 0.8$	0/1000
Baselines (no admissibility guarantee; cited for context only — comparison is heterogeneous)				
BB†	blackbox diff. solver	**54.7**%	—	—
DataSP‡	diff. Floyd-Warshall	10.8%	40.8%	—
Neural A*§	diff. A* with learned guide	∼98% *	—	—
iA*‖	improved Neural A*	—	—	—

† Published results from Vlastelica et al. (ICLR 2020), Table 1.

‡ Reproduced from released code (Lahoud et al., 2024), seed=0, 100 epochs.

§ Yonetani et al. (ICML 2021).

* Reported as cost-optimality (cost-optimal paths), not exact path match; not directly comparable with the AAC path-match column.

‖ Chen et al. (2025). Metrics not directly comparable; included for completeness.

** ResNet numbers correspond to the released encoder configuration (5×5 stem, 3 BasicBlocks at 64 channels, AdaptiveAvgPool, 1×1 convolutions; full spec in Appendix B); the released unit tests reproduce these numbers from `results/warcraft/ablation_results_resnet_a10_multiseed.csv`.

Ablation: encoder capacity (admissibility-preserving). Upgrading from the 3-layer CNN to ResNet with 10× cost supervision raises Jaccard from $20.9\% \pm 0.8$ to $59.3\% \pm 3.9$ and improves path match from $0.4\% \pm 0.1$ to $20.3\% \pm 2.4$ (3 seeds, 10,000 train maps), with admissibility preserved at 0/1000 violations across both configurations. Encoder capacity and cost supervision are the primary drivers; the compressor's role is providing structurally guaranteed admissibility at deployment rather than contributing to the learning signal (across an instrumented 30-epoch CNN training run, the per-epoch ratio of encoder to compressor gradient norms ranged from $\sim 10^2$ at training start to $\sim 10^4$ during the bulk of training, i.e. the compressor receives between 2 and 4 orders of magnitude less gradient signal than the encoder; the CNN configuration typically reaches its early-stopping point well before the 100-epoch cap under patience 15). Freezing the compressor leaves the per-mode metrics within standard error of the full-end-to-end run on both encoders, while freezing the encoder is catastrophic (cost regret 1.16 on ResNet versus 0.017 for the trained encoder).

For the CNN configuration, Table 24 below repeats the encoder-vs-compressor decomposition on cost regret to make the contribution split visible on a single shared scalar: freezing the compressor has negligible effect (Full E2E ≈ Frozen Compressor), while freezing the encoder is catastrophic (cost regret >1). The decomposition is qualitative; the absolute cost-regret values are not commensurable with the baselines for the reasons given in Table 23's caption.

Comparison with DataSP. We reproduce DataSP (Lahoud et al., 2024) from released code, obtaining 0.384 cost regret and 10.8% path match on the test set (seed=0, 100 epochs). The value of AAC lies in the combination of deployment-time architectural admissibility, $O(m)$-per-lookup heuristic evaluation after label construction, and a cost regret in the same order of magnitude as the blackbox baseline; among the baselines compared here, no other method offers all three simultaneously.

Table 24: ABLATION Cost regret by training mode on Warcraft 12×12 (CNN encoder, 3 seeds). The encoder drives the learning signal; freezing the compressor has no effect. BB: blackbox baseline (Vlastelica et al., 2020).

Training Mode	Cost Regret ↓
Full E2E	0.144 ± 0.027
Frozen Compressor	0.144 ± 0.027
Frozen Encoder	1.012
BB† (baseline)	0.173

† Vlastelica et al. (2020).

Comparison with Neural A* and iA*. Neural A* (Yonetani et al., 2021) reports ∼98% cost-optimal paths on Warcraft 12×12 maps (their "optimality" metric measures whether the found path achieves optimal cost, which is strictly weaker than our "path match" metric that requires the exact same path). iA* (Chen et al., 2025) improves Neural A*'s search efficiency and out-of-distribution generalization on similar grid-planning tasks. Neither method provides admissibility guarantees. Direct numerical comparison in Table 23 is complicated by metric differences: Neural A* reports optimality rate and search efficiency rather than cost regret, Jaccard, or exact path match. We include both in Table 23 with their published optimality metric for reference. The key regime distinction is that Neural A* and iA* learn powerful guidance functions that may violate admissibility, while AAC guarantees admissibility architecturally at the cost of lower path-match accuracy; the approaches are complementary rather than competing.

G Reproducibility

Data and code availability. The full source tree, all result tables and run logs backing every Table and Figure in this paper, and a verbatim copy of the pre-registered OGB-arXiv prediction (Appendix E) are released at `https://anonymous.4open.science/r/aac`. The pre-registration record is at `results/PREREG.md`. The repository's `results/README.md` provides the per-Section / per-Table / per-Figure mapping to its generating experiment record, and `src/experiments/README.md` documents the configuration-driven runner layout. Installation, environment setup, and full reproduction commands are in the repository `README.md`.

Hyperparameters. Key settings: $K_0 \in \{32, 64, 128\}$, $m \in \{8, 16, 32, 64\}$, learning rate 10^{-3}, 200 epochs (static) / 100 epochs (contextual), batch size 256 (static) / 24 (contextual), Adam optimizer. The full hyperparameter grid lives in YAML configuration files alongside the released code.

Sentinel masking. Unreachable landmark distances are stored as a sentinel (10^{18}) and masked out of the max computation; the theorems of Section 3.3 do not depend on the specific magnitude. The released unit tests verify admissibility under sentinel masking, and the run-time SCC audit confirms that query endpoints share an SCC on every benchmark graph so the masking path is never taken in production runs.

H Broader Impact

The value of AAC lies in safety-critical, ML-integrated planning pipelines: the architectural admissibility guarantee (Proposition 2) holds in exact arithmetic given exact teacher labels regardless of training quality, distribution shift, or early stopping, a property optimization-dependent guarantees cannot match. Fewer A* expansions reduce compute and energy cost for large-scale routing (up to 92% reduction at 128 B/v on the 4.5M-node Netherlands network). Admissibility guarantees optimal path *cost* but not other path qualities (smoothness, safety-corridor width); the method requires precomputed landmark distances, restricting it to settings where graph structure is known offline; we do not foresee direct negative societal impacts beyond the dual-use nature of any routing improvement.