A HEURISTIC FOR REGRESSION SEARCH IN LEARNED QUASIMETRIC SPACES WITH CONDITIONAL BOUNDED-SUBOPTIMALITY GUARANTEES

Anonymous authors
Paper under double-blind review

ABSTRACT

Classical planning tasks often require reasoning over large state spaces to achieve conjunctive goals. Regression search reasons backward over partial states and considers actions that can support the current subgoals, making it useful in a range of planning settings. Yet comparatively few informative heuristics have been designed for regression search. The classical h^m family provides admissible estimates by evaluating the most expensive m-fact subsets and returning their maximum, but this scalar aggregation can lose information when costs accumulate across many serial subgoals. We introduce **Partial-State IQE (PS-IQE)**, a learned heuristic that constructs a lower-bound representation of each partial state in the Interval Quasimetric Embedding (IQE) latent space. We extend QRL with a relative local constraint alongside its original absolute form and use log-sum-exp to emphasize large transition violations. When either local bound holds for every transition, regression search with A* guided by PS-IQE satisfies the corresponding bounded-suboptimality guarantee. After transition-system-specific preprocessing, both the evaluation time and storage are polynomial in the number of grounded facts for any fixed subset order m. As with h^m, the cost grows exponentially in m, and m is small in practice. Experiments on tasks with serial subgoals show that PS-IQE remains tight at small subset orders, and that this tightness can translate into reduced regression-search effort.

1 INTRODUCTION

Classical planning is a central problem in AI and a general framework for sequential decision making, with applications in logistics, robotics, scheduling, and autonomous systems. It seeks a sequence of actions that transforms a given initial state into one satisfying a goal condition, and is commonly formulated as search in an implicitly defined state-transition system. Although actions are specified compactly, the induced state space can be enormous. Planning is therefore typically approached as a search problem rather than by explicitly constructing the full state-transition graph (Ghallab et al., 2004).

State-space search for classical planning can proceed in either direction. Progression, i.e., forward search, starts from the initial state and applies actions forward, whereas regression search starts from the goal condition and reasons backward about the conditions required to achieve it. The effectiveness of state-space search depends heavily on its heuristic guidance. A* orders search nodes by $f(s) = g(s) + h(s)$, where $g(s)$ is the cost already incurred and $h(s)$ estimates the remaining cost (Hart et al., 1968). In progression search, $h(s)$ estimates the cost from a complete state s to the fixed goal. However, reversing the search direction introduces a distinct heuristic challenge: each search node is a partial state representing a set of complete states while leaving some facts unspecified. This incomplete specification can weaken standard heuristic estimates and complicate subsumption checking and reachability pruning (Alcázar et al., 2013; 2014; Bonet & Geffner, 2001). These limitations, together with a lack of informative heuristics for partial states, have restricted the practical use of backward heuristic search (Lei & Lipovetzky, 2021).

Many widely used planning heuristics were designed for progression search and are not directly suited to regression search. Traditional heuristics can face two difficulties in regression search: evaluations on partial states may be less informative, and preprocessing tied to a fixed goal may need to be repeated for each regression node. The critical-path heuristic h^m is an important exception: it was designed for regression and reuses fact-subset costs computed from the initial state. However, it

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

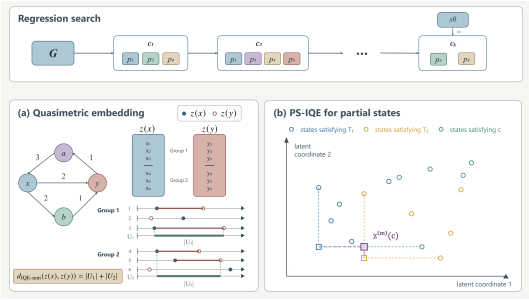


Figure 1: PS-IQE for regression search. Regression search generates new subgoals by regressing the goal through actions. These subgoals are partial states, each representing a set of complete states. PS-IQE uses the IQE learned over complete states to construct heuristic estimates for these partial states from their fact subsets.

only considers the most costly fact subset up to size m, and often underestimates the total cost when costs accumulate across multiple subgoals. Learned planning heuristics, on the other hand, have mainly been evaluated on unseen states across planning instances in progression search. Relational methods such as WL-GOOSE explicitly represent the goal condition and could in principle evaluate changing regression subgoals, but their reported results do not establish their performance in this setting. Moreover, most learned planning heuristics are not admissible, trading optimality guarantees for greater search efficiency.

To address this need, we propose **Partial-State IQE (PS-IQE)**, a learned heuristic for regression search. Like h^m, PS-IQE evaluates regression subgoals by precomputing information for small fact subsets and reusing that information during search. Unlike h^m, which takes the maximum over scalar subset values, PS-IQE leverages coordinate-wise information in an IQE (Wang & Isola, 2022) latent space. We adapt the QRL objective (Wang et al., 2023) to learn the quasimetric for planning and show that A* guided by PS-IQE is bounded-suboptimal under stated conditions. A model is trained for each state-transition system: the quasimetric is learned on complete states, while estimates for partial states are constructed from fact-subset coordinate-wise lower bounds. Like any heuristic learned per instance, PS-IQE pays an offline training cost before search can begin. This cost is amortized within the state-transition system—across tasks with different initial states and goals. Figure 1 illustrates the key idea of PS-IQE.

We adapt QRL's local objective to classical planning by introducing relative violations into local constraints and by annealing the local objective toward a squared log-sum-exp loss that emphasizes large violations in each minibatch. For regression A* guided by PS-IQE, we prove bounded-suboptimality guarantees under both absolute and relative per-transition error bounds. When the corresponding bound holds for every transition, the absolute form gives an additive bound on the returned plan cost, while the relative form gives a multiplicative bound. Our experiments indicate that PS-IQE can provide informative estimates for partial states at small subset orders: by taking the coordinate-wise maximum in the embedding space, PS-IQE combines complementary lower bounds from different fact subsets, which may help it capture costs that accumulate across serial subgoals.

In summary, our main contributions are as follows:

1. **PS-IQE.** We propose a novel latent-space construction that turns a quasimetric learned over complete states into a heuristic for partial states encountered in regression search. For any fixed subset order m, both evaluation time and storage are polynomial in the number of grounded facts after transition-system-specific preprocessing.

2. **Conditional bounded-suboptimality.** We adapt QRL's objective to classical planning, and show that local bounds on the learned quasimetric yield bounded-suboptimality guarantees for regression A* guided by PS-IQE when they hold for every transition.

3. **Empirical analysis.** We show that, compared with baselines, PS-IQE is an informative heuristic on several IPC instances and a robotic planning task, and that the computational cost can be amortized by its cache over repeated regression searches in the same transition system.

2 Related Work

2.1 Regression search and backward reasoning in planning

Regression search represents search nodes as partial states containing only the conditions needed by the current plan suffix and considers actions that support those conditions without conflicting with the remainder. This goal-directedness can substantially reduce the branching factor when relatively few actions are relevant to the goal, while providing backward reachability information that complements forward search.

In behavior-tree planning, BT Expansion replaces an unsatisfied subgoal with the preconditions of an action that achieves it (Cai et al., 2021). What it builds is not a plan for one initial state but a reactive policy whose guarantee is a region of attraction, so its condition nodes denote sets of states by construction. Regression search can also be combined with forward exploration in bidirectional search (Alcázar et al., 2014). In abstraction refinement, it can complement forward refinement by checking an abstract plan backward from the goal, producing abstractions that are less focused on the initial state (Pozo et al., 2024). Other work has focused on improving explicit regression search, in which each search node represents an individual partial state. Alcázar et al. (2013, 2014) introduced invariant-based disambiguation, e-deletion, cached best supporters, preferred operators, duplicate detection, and subsumption to address information loss and improve search control. These settings motivate informative heuristic estimates for regression search.

2.2 Traditional Planning Heuristics

Traditional heuristics derive cost estimates from the symbolic task description. Delete-relaxation methods include the admissible $h^{\max}$, the generally inadmissible additive estimate h^{add} (Bonet & Geffner, 2001), and the relaxed-plan heuristic h^{FF} (Hoffmann & Nebel, 2001). Abstraction methods include pattern databases (Culberson & Schaeffer, 1998; Edelkamp, 2001), their cost-partitioned combinations (Pommerening et al., 2013; Seipp et al., 2020), and merge-and-shrink abstractions (Helmert et al., 2007; 2014). Landmark methods provide another family of estimates; LM-cut combines landmarks with delete relaxation to obtain an admissible heuristic (Helmert & Domshlak, 2009).

The admissible h^m family is most closely related to our work: it generalizes $h^{\max}$ by accounting for interactions among sets of at most m facts and approximates larger conditions by the maximum value of their m-fact subsets (Haslum & Geffner, 2000). Additive h^m partitions the action set into disjoint subsets and sums the h^m estimates obtained by assigning zero cost to actions outside each subset (Haslum et al., 2005). Another improvement uses relaxed search to partially compute higher-order h^m values at lower computational cost (Haslum, 2006).

2.3 Learned heuristics for planning

Alongside traditional symbolic approaches, recent work has explored learning planning heuristics from data generated on planning tasks. Early learned heuristics focused on a fixed state space and goal condition. Ferber et al. (2020) studied neural-network heuristics, evaluating their generalization to unseen initial states, while Bettker et al. (2024) further examined how the distribution of training states and the accuracy of their cost estimates affect the resulting heuristic and search performance. Another line of work is relational methods, which use task representations to transfer knowledge across instances: STRIPS-HGN encodes a delete-relaxed task as a hypergraph, GOOSE uses grounded or lifted graphs, and WL-GOOSE combines Weisfeiler–Leman features with classical statistical learning (Chen et al., 2024a;b; Shen et al., 2020). These methods can provide useful guidance, but their estimates are generally not admissible.

3 Preliminaries

3.1 Classical planning and regression search

We consider finite deterministic propositional STRIPS tasks. Complete states, goal conditions, and regressed partial states are represented as sets of ground facts; facts absent from a complete state are false under the closed-world assumption.

The standard propositional STRIPS semantics is defined as follows (Bonet & Geffner, 2001; Ghallab et al., 2004). A finite deterministic planning task is a tuple $\Pi = \langle \mathcal{P}, \mathcal{A}, s_0, G \rangle$, where $\mathcal{P}$ is a set of ground facts, $\mathcal{A}$ is a set of actions, s_0 is the initial state, and G is a partial state specifying the goal condition. Each action a has a precondition set $\mathrm{pre}(a)$, an add-effect set $\mathrm{add}(a)$, a delete-effect set $\mathrm{del}(a)$, and a cost $w(a)$. The task induces an edge-weighted directed state-transition graph

$\mathcal{M} = (\mathcal{S}, E)$, where edges correspond to action applications and carry the associated action costs. For states $s, t \in \mathcal{S}$, let $d^*(s,t)$ denote the shortest-path cost from s to t, with the convention that $d^*(s,t) = \infty$ if t is unreachable from s.

Regression search starts at G. A regression node is a partial state c specifying the facts that must hold before the action sequence regressed so far can be executed. An action a is relevant to c if it achieves at least one fact in c. It must also delete no fact in c, so that regressing through a does not undo part of c. Regression search considers only actions meeting both conditions. Regressing c through a gives

$$\text{regress}(c, a) = \text{pre}(a) \cup \big(c \setminus \text{add}(a)\big). \tag{1}$$

Search terminates when it reaches a partial state c such that $s_0 \models c$. Regression uses the forward action model to compute predecessor conditions; it does not require executable inverse actions and therefore also applies to irreversible domains (Alcázar et al., 2013; Ghallab et al., 2004).

For any fact set $C \subseteq \mathcal{P}$, write

$$\mathcal{S}_C = \{y \in \mathcal{S} \mid C \subseteq y\}$$

for the set of complete states satisfying C. A partial state c thus denotes $\mathcal{S}_c$. If $\mathcal{S}_c \neq \emptyset$, we define its optimal cost from the initial state as

$$H^*(s_0, c) = \min_{y \in \mathcal{S}_c} d^*(s_0, y). \tag{2}$$

Regression search can be performed with A*. A* orders nodes by $f(c) = g(c) + h(c)$ (Hart et al., 1968). Here $g(c)$ is the accumulated cost of the plan suffix regressed from G to c, each regression edge has the cost of its corresponding forward action, and $h(c)$ estimates $H^*(s_0, c)$. Since regression search proceeds backward from the goal, the heuristic estimates the forward distance from s_0 to the set of states satisfying c. A* maintains OPEN, the set of generated nodes awaiting expansion, and CLOSED, the set of expanded nodes. A node is reopened if a lower-cost path to it is found, and search terminates when a solution node is selected from OPEN.

3.2 Quasimetric representations

A quasimetric $d : X \times X \to \mathbb{R}_{\geq 0}$ satisfies nonnegativity, $d(x,x) = 0$, and the triangle inequality, $d(x,z) \leq d(x,y) + d(y,z)$. Unlike a metric, a quasimetric is not required to be symmetric, allowing $d(x,y) \neq d(y,x)$. This asymmetry matches the directed state-transition graph of a planning task: the minimum path cost from x to y may differ from that from y to x. Shortest-path distance on a directed state-transition graph is a quasimetric within each strongly connected component; assigning $+\infty$ to unreachable pairs yields an extended quasimetric over the full state space. Neural distance models can impose the triangle inequality as an architectural constraint (Pitis et al., 2020).

IQE applies a learned encoder to each state s and reshapes its latent representation as a matrix $z(s) \in \mathbb{R}^{k \times l}$. For rows $x_i, y_i \in \mathbb{R}^l$, the i-th IQE component is the Lebesgue measure of a union of intervals

$$q_i(x,y) = \left| \bigcup_{j=1}^{l} [x_{ij}, \max(x_{ij}, y_{ij})] \right|, \tag{3}$$

The component values are combined using either summation or the maxmean reduction:

$$\begin{aligned} d_{\text{IQE-sum}}(x,y) &= \sum_{i=1}^{k} q_i(x,y), \\ d_{\text{IQE-maxmean}}(x,y) &= \alpha \max_{1 \leq i \leq k} q_i(x,y) \\ &\quad + (1-\alpha)\frac{1}{k}\sum_{i=1}^{k} q_i(x,y), \end{aligned}$$

where $\alpha \in [0,1]$ is learned. We write $d_\theta(s,t) = d_{\text{IQE}}(z(s), z(t))$, where θ denotes the learned model parameters and d_{IQE} the selected IQE formulation. Both formulations satisfy the quasimetric

constraints for any encoder. IQE-maxmean can represent any finite-valued quasimetric on a finite space exactly and can approximate continuous quasimetrics on compact spaces; IQE-sum approximates finite, finite-valued quasimetric spaces with a distortion bound (Wang & Isola, 2022).

3.3 Quasimetric reinforcement learning

QRL uses a quasimetric model to learn goal-conditioned distances: it increases distances between sampled state–goal pairs while constraining overestimation on local transitions (Wang et al., 2023). Writing the cost of a transition e as $w(e)$, its objective is

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{(s,g)\sim\mu}[d_\theta(s,g)] \\ \text{subject to} \quad & \mathbb{E}_{e=(s,s')\sim\rho} \\ & \left[\, [d_\theta(s,s') - w(e)]_+^2 \,\right] \le \epsilon^2, \end{aligned}$$

where μ denotes the distribution induced by sampling $s \sim p_{\text{state}}$ and $g \sim p_{\text{goal}}$ as in QRL, ρ is the sampling distribution over transitions, and ϵ^2 bounds the expected squared one-step overestimation. The local constraint controls expected one-step violations, while the global objective discourages the all-zero solution. A bound that holds on every transition would extend along paths by the triangle inequality.

4 Partial-State IQE

PS-IQE learns a quasimetric between complete states using an adaptation of the QRL loss and extends it, through coordinate-wise lower bounds for small fact subsets, to the case in which its second argument is a partial state. We first describe how the complete-state quasimetric is learned, then present the construction of a heuristic for partial states, and finally provide the corresponding theoretical analysis. Figure 2 summarizes the PS-IQE pipeline.

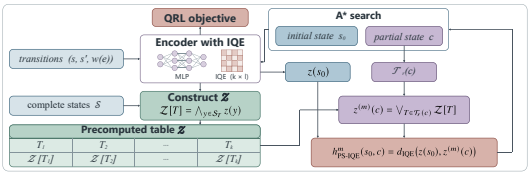


Figure 2: Overview of the PS-IQE pipeline. Learned complete-state embeddings are used to construct the fact-subset table $\mathcal{Z}$, from which PS-IQE evaluates partial states during regression search.

4.1 Learning the quasimetric

QRL constrains the expected squared absolute violation $[d_\theta(s,s') - w(e)]_+^2$ of the local transition bound $d_\theta(s,s') \le w(e)$. For strictly positive action costs, we introduce a multiplicative alternative based on the relative violation

$$v_+(e) = [d_\theta(s,s')/w(e) - 1]_+. \tag{4}$$

for a sampled transition $e = (s,s')$ with cost $w(e)$. The absolute and relative forms induce additive and multiplicative relaxations of the local transition constraint, respectively. As shown in Corollary 1, under uniform per-transition bounds, they yield a path-length-dependent additive bound and a scale-independent relative suboptimality bound, respectively.

The original QRL constraint controls the expected squared violation under the transition distribution ρ, whereas the search analysis assumes a uniform bound over transitions. To obtain a smooth upper approximation to the maximum violation in each minibatch, we use the log-sum-exp function (Nesterov, 2005). For a nonempty batch B and sharpness parameter $\beta_{\text{LSE}} > 0$, define

$$\text{LSE}_{\beta_{\text{LSE}}}(B) = \beta_{\text{LSE}}^{-1} \log \sum_{e\in B} \exp(\beta_{\text{LSE}} v_+(e)) \,. \tag{5}$$

The LSE upper-bounds every violation in the batch and approaches the maximum as β_{LSE} increases. During training, we anneal the local objective from the mean squared relative violation to the squared LSE, gradually concentrating updates on the largest violations. Appendix C.3 gives the bounds and the complete annealed objective.

4.2 Extending IQE to partial states

Let $\mathcal{S}$ be the set of complete states, $\mathcal{P}$ the set of ground facts, $z : \mathcal{S} \to \mathbb{R}^{k\times l}$ the learned embedding map, and $m \geq 1$ the subset order. For $u, v \in \mathbb{R}^{k\times l}$, write $u \preceq v$ when $u_{ij} \leq v_{ij}$ for every coordinate. Let $u \wedge v$ and $u \vee v$ denote their coordinate-wise minimum and maximum, with $\bigwedge$ and $\bigvee$ denoting the corresponding operations over nonempty finite families. We write $\boldsymbol{\infty}_{k\times l}$ for the $k \times l$ matrix whose entries are all $+\infty$. We now formalize the partial-state construction; Appendix A.1 gives complete preprocessing and evaluation pseudocode.

For a complete state x and a partial state c with $\mathcal{S}_c \neq \emptyset$, define the learned distance from x to the set of states satisfying c as

$$H_\theta(x, c) = \min_{y\in\mathcal{S}_c} d_\theta(x, y), \tag{6}$$

For every nonempty $T \subseteq \mathcal{P}$ with $|T| \leq m$, define

$$\underline{z}_T = \begin{cases} \bigwedge_{y\in\mathcal{S}_T} z(y), & \mathcal{S}_T \neq \emptyset, \\ \boldsymbol{\infty}_{k\times l}, & \mathcal{S}_T = \emptyset. \end{cases} \tag{7}$$

For a nonempty partial state c, let $\mathcal{T}_{\leq m}(c) = \{T \subseteq c : 1 \leq |T| \leq m\}$, and define

$$z^{(m)}(c) = \bigvee_{T\in\mathcal{T}_{\leq m}(c)} \underline{z}_T. \tag{8}$$

We call $z^{(m)}(c)$ the order-m **relaxed lower corner** of c. The corresponding order-m PS-IQE heuristic is

$$h^m_{\text{PS-IQE}}(x, c) = d_{\mathrm{IQE}}(z(x), z^{(m)}(c)). \tag{9}$$

If any subset bound in Equation 8 equals the infeasibility sentinel $\boldsymbol{\infty}_{k\times l}$, evaluation returns $+\infty$ without calling d_{IQE}; otherwise Equation 9 applies. The empty partial state is satisfied by every complete state, so we set $h^m_{\text{PS-IQE}}(x, \emptyset) = 0$. Equation 8 ranges over all nonempty subsets of size at most m, but the same value is obtained using only subsets of size $r = \min(m, |c|)$, as shown in Appendix A.5.

The IQE quasimetric has the following monotonicity.

Lemma 1 (Monotonicity of the IQE quasimetric in its second argument)**.** *For any fixed u, if $v \preceq v'$, then $d_{\mathrm{IQE}}(u, v) \leq d_{\mathrm{IQE}}(u, v')$.*

The proof follows directly from inclusion of the interval unions in equation 3 and is given in Appendix A.2.

This monotonicity gives the following lower bound for the PS-IQE heuristic.

Proposition 1 (Lower bound on the learned partial-state distance)**.** *For every complete state x and partial state c with $\mathcal{S}_c \neq \emptyset$,*

$$h^m_{\text{PS-IQE}}(x, c) \leq H_\theta(x, c). \tag{10}$$

For any completion $y \in \mathcal{S}_c$, every subset bound used in equation 8 is coordinate-wise no greater than $z(y)$. Their coordinate-wise maximum therefore remains below $z(y)$, and Lemma 1 gives the result. The full proof is provided in Appendix A.3.

The construction does not require partial states during training, and the order-m relaxed lower corner need not correspond to the embedding of any state in $\mathcal{S}$. Increasing m can only tighten this relaxation; when $m = |c|$, $z^{(m)}(c)$ coincides with the exact lower corner of the completion embeddings.

For comparison, consider a scalar-level alternative

$$h^m_{\text{scalar}}(x, c) = \max_{T\in\mathcal{T}_{\leq m}(c)} d_{\mathrm{IQE}}(z(x), \underline{z}_T). \tag{11}$$

It computes the quasimetric distance to the coordinate-wise lower bound for each fact subset and then takes the maximum.

Proposition 2 (Dominance over scalar-level combination)**.** *For every complete state x and nonempty partial state c with $\mathcal{S}_c \neq \emptyset$,*

$$h^m_{\text{scalar}}(x, c) \leq h^m_{\text{PS-IQE}}(x, c). \tag{12}$$

The inequality follows from Lemma 1 because every z_T is coordinate-wise no greater than $z^{(m)}(c)$. It can be strict when different subsets give tighter bounds in different coordinates. Proposition 2 compares only constructions based on the same latent subset bounds, not h^m, which uses a different relaxation. The full proof is given in Appendix A.4.

4.3 Bounded suboptimality in search

Here, we establish the bounded suboptimality of regression A* guided by PS-IQE and state the conditions under which this result holds. We begin by decomposing an optimal plan at each partial state along its induced regression path.

Lemma 2 (Optimal-plan decomposition)**.** *Let π^* be an optimal plan with cost C^*, and let c lie on the regression path induced by π^*. Let $g^*(c)$ denote the minimum cost of a regression path from G to c. Then*

$$g^*(c) + H^*(s_0, c) = C^*.$$

The proof splits an optimal plan at c and uses regression soundness for the converse direction; see Appendix B.1.

Theorem 1 (Bounded suboptimality under local-error bounds)**.** *Consider regression A* guided by PS-IQE, and let π^* be an optimal plan of cost C^* whose corresponding regression path is preserved by the pruning rules used in this search. Suppose that, for every transition $e = (u, v) \in E$, the learned distance satisfies*

$$d_\theta(u, v) \leq w(e) + \Delta(e), \qquad \Delta(e) \geq 0. \tag{13}$$

If the search reopens an expanded node whenever a lower g-value is found and returns a plan of cost $\widehat{C}$, then

$$\widehat{C} \leq C^* + \sum_{e \in \pi^*} \Delta(e). \tag{14}$$

The proof follows the regression path induced by π^*. Until a solution is selected, OPEN contains a node on this path with its optimal g-value. Proposition 1, the triangle inequality, and equation 13 bound that node's f-value by the right-hand side of equation 14. Since A* selects a node with minimum f-value, the returned solution satisfies the same bound. Appendix B.2 gives the full proof.

Corollary 1. *Suppose the assumptions in Theorem 1 hold, and let L^* denote the number of actions in the optimal plan π^*.*

1. *If $d_\theta(u, v) \leq w(e) + \delta$ for every transition $e = (u, v) \in E$, where $\delta \geq 0$, then*

$$\widehat{C} \leq C^* + \delta L^*.$$

2. *If $d_\theta(u, v) \leq (1 + \epsilon)w(e)$ for every transition $e = (u, v) \in E$, where $\epsilon \geq 0$, then*

$$\widehat{C} \leq (1 + \epsilon)C^*.$$

The two cases follow from Theorem 1 by setting $\Delta(e) = \delta$ and $\Delta(e) = \epsilon w(e)$, respectively; details are given in Appendix B.3. The multiplicative bound also remains valid under cost-aware subsumption pruning, as proved in Appendix B.4.

4.4 Complexity Analysis

Let $P = |\mathcal{P}|$ denote the number of ground facts and $D = k \times l$ the number of projected IQE coordinates. The table requires $\mathcal{O}(P^m D)$ storage, and each heuristic evaluation takes $\mathcal{O}(P^m D)$ time in the worst case. For fixed m, both complexities are polynomial in P and linear in D after transition-system-specific preprocessing.

5 EXPERIMENTS

5.1 SETUP

We first study five IPC Gripper instances, then compare regression-search heuristics on one IPC Logistics task and one VirtualHome-based task (Puig et al., 2018; Chen et al., 2025)—chosen for their uniform and heterogeneous action costs, respectively—and finally analyze amortization in BT Expansion. All PS-IQE models are trained using the IQE-sum quasimetric with the relative local constraint over three seeds. All methods use regression A* with duplicate detection and reopening but no subsumption pruning; for the non-admissible heuristics we also report the search strategy conventionally used with each: GBFS (priority h alone) for h^{FF} and WL-GOOSE, weighted A* ($w = 5$) for HSPr. Both comparisons ran on the same server (dual Xeon Platinum 8481C CPUs, 16 vCPUs available to the container; RTX 4090D GPU); reported times are indicative only. Appendix C gives the full protocol.

5.2 CASE STUDY: GRIPPER

Gripper instances that we study place a robot with two grippers in a room with $n \in \{4, 6, 8, 10, 12\}$ balls; the goal is for all n balls to reach another room. The robot can carry only two balls per trip, so reaching the goal requires repeated trips whose cost accumulates across all n balls. h^m cannot exceed the optimal cost of moving m balls no matter how large n grows.

Experiments show that at the root of regression search, $h^2 = 4$ and $h^3 = 8$ stay constant while the PS-IQE estimates grow with C^* (Figure 3): $h^2_{\text{PS-IQE}}/C^*$ ranges from 0.880 to 0.934 and $h^3_{\text{PS-IQE}}/C^*$ from 0.930 to 0.966 across instances, with a maximum absolute difference of 0.31 across three seeds (Appendix D.2). In this case, PS-IQE tracks accumulated cost, showing its potential for tasks with many serially coupled subgoals.

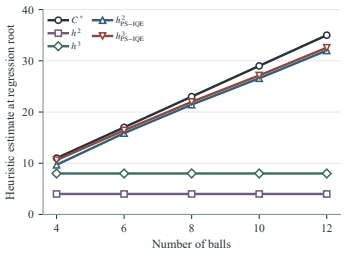


Figure 3: Heuristic estimates at the root of regression search as the number of Gripper subgoals increases. PS-IQE estimates track the growth of the optimal plan cost C^*.

5.3 REGRESSION-SEARCH COMPARISON

`logistics-4-0` is the official IPC Logistics instance, with airplanes, trucks, and packages moving between cities under uniform unit action costs (941,192 states, 7,663,992 edges, a 4-atom goal, $C^* = 20$). `vhm_7o` is a VirtualHome-based instance with 7 objects initially located in two places, to be moved to a refrigerator, under heterogeneous action costs (361,584 states, 1,813,752 edges, an 8-atom goal, $C^* = 203$). Tables 1 and 2 compare PS-IQE with baselines on these two instances. Total time includes per-task preprocessing but excludes transition-system-level construction and offline training.

Table 1: Regression-search results on the IPC `logistics-4-0` instance.

Heuristic	Search	Nodes	Time (s)	Prep (ms)	Adm.
h^{FF}	GBFS	22	0.005	0.5	No
HSPr	wA* ($w = 5$)	23	0.004	0.4	No
h^{FF}	A*	219	0.048	0.5	No
HSPr	A*	257	0.039	0.4	No
SysPDB-SCP$_{\leq 2}$	A*	936	0.173	8.2	Yes
$h^2_{\text{PS-IQE}}$	A*	$1{,}073.7 \pm 162.7$	0.332 ± 0.046	3.7 ± 0.2	△
$h^2_{\text{PS-IQE}}/1.1$	A*	$1{,}765.7 \pm 198.0$	0.550 ± 0.063	4.5 ± 1.0	△
Additive h^2 ($q = 4$)	A*	2,116	0.403	16.7	Yes
SysPDB-UCP$_{\leq 2}$	A*	5,186	0.974	1.8	Yes
WL-GOOSE	GBFS	5,827	6.130	2,258.8	No
h^2	A*	5,954	1.057	5.2	Yes
h^1	A*	15,359	1.842	0.3	Yes
UCS	A*	21,303	2.577	0.0	Yes

Table 2: Regression-search results on the VirtualHome-based `vhm_7o` instance.

Heuristic	Search	Nodes	Time (s)	Prep (ms)	Adm.
h^{FF}	GBFS	45	0.017	0.4	No
WL-GOOSE	GBFS	61	3.337	2,971.2	No
HSPr	wA* ($w = 5$)	178	0.065	0.5	No
HSPr	A*	215	0.068	0.4	No
h^{FF}	A*	32,996	12.800	0.4	No
$h^2_{\text{PS-IQE}}$	A*	$58{,}150.7 \pm 10{,}653.0$	26.715	5.4	△
$h^2_{\text{PS-IQE}}/1.1$	A*	$79{,}381.0 \pm 11{,}149.9$	36.985	5.0	△
Additive h^2 ($q = 2$)	A*	122,142	51.004	22.2	Yes
h^2	A*	135,485	45.547	8.8	Yes
SysPDB-SCP$_{\leq 2}$	A*	148,428	50.939	8.9	Yes
SysPDB-UCP$_{\leq 2}$	A*	199,191	66.844	1.5	Yes
h^1	A*	211,631	55.922	0.4	Yes
UCS	A*	261,411	75.646	0.0	Yes

△ The triangle marks the condition in Theorem 1: when it holds, the scaled heuristic is admissible and the unscaled heuristic has a bounded-suboptimality factor of $1 + \epsilon$.

PS-IQE gives a competitive heuristic on both instances, expanding fewer nodes than h^2 while returning optimal-cost plans. Appendix C.5 describes the baselines and their implementations, and Appendix D.1 reports the measured edge relaxation underlying its bounded-suboptimality guarantee on each instance.

5.4 Amortization in BT Expansion

We run behavior tree expansion on `vhm_7o`, whose transition system has 361,584 complete states. For each partial state c expanded during regression search, PS-IQE caches its coordinate-wise lower bound $z^{(m)}(c)$, which depends only on c, so this cache can be shared across replanning tasks that start from different states in the same transition system. Across 100 tasks with a fixed goal and independently sampled start states, the cumulative symbolic-successor cache hit rate reaches 87.1% once these searches have together expanded 138,261 unique partial states, and this reuse cuts cumulative wall-clock time to 3.19× faster than restarting cold at every task (Figure 4).

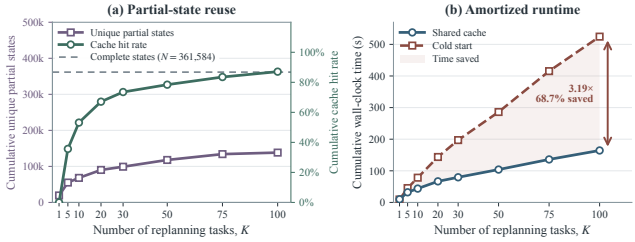


Figure 4: Amortization across repeated regression searches on `vhm_7o`. (a) Cumulative unique partial states visited and cumulative symbolic-successor cache hit rate as the number of searches K grows. (b) Cumulative wall-clock time with a shared cache versus a cold restart at each search.

6 Discussion and Future Work

Although the PS-IQE heuristic can also be applied to progression search, we study it here in the context of regression search, motivated by its connection to h^m and by the different role that partial states play in the two search directions. In progression search, the goal condition remains fixed, so the lower-bound construction uses the same set of goal completions throughout the search. Regression search instead evaluates a changing partial state at each node. A regression step replaces conditions achieved by an action with its preconditions. The resulting partial state may be more restrictive and therefore satisfied by fewer complete states. This can reduce the relaxation gap introduced by the partial-state construction and provide more informative guidance as search proceeds.

For future work, mutex groups or other planning invariants could be used to partition the completions of a partial state into disjoint groups represented by more specific partial states. Evaluating $h^m_{\text{PS-IQE}}$ for the partial state representing each group and taking the minimum may give a tighter lower bound on $H_\theta(x, c)$. When every group contains a single completion, the construction exactly recovers $H_\theta(x, c)$.

Another promising direction is to separate reachability prediction from finite-distance estimation for state pairs. We evaluate state-transition systems whose directed graphs are strongly connected, so that $d^*(u, v) < \infty$ for every ordered pair of states. This restriction is consistent with QRL's theoretical setting: its function-approximation result assumes a compact state space and a continuous, hence bounded, V^*, while its recovery analysis treats the optimal goal-reaching cost as a finite-valued quasimetric. Future work could apply the learned quasimetric to reachable state pairs and explicitly represent unreachable pairs as having infinite distance.

It is also worth exploring whether PS-IQE can generalize to planning instances with an unseen state-transition system. The current multi-hot MLP encoder is tied to the facts of a single state-transition system and therefore requires retraining for each new system. Replacing the state encoder with a relational graph encoder could capture structure shared across instances and allow the learned representation to transfer across problem sizes. Distances would still be computed with IQE, preserving the quasimetric properties by construction, while the coordinate-wise lower-bound table would be constructed separately for each state-transition system. Bounded suboptimality on a different system would remain conditional on the overestimation bound holding for every transition in that system.

AI USE STATEMENT

Generative AI tools, including large language models, were used to assist the authors with research implementation, code review and testing, the synthesis of QRL training data from planning problems, related-work discovery, manuscript drafting and language editing, and the development and checking of mathematical derivations. In each of these activities, the authors established the research direction, provided the models with the initial conceptual framework, mathematical formulations, and implementation prototypes, specified the tasks and relevant constraints, reviewed the generated outputs, corrected errors, and made the final methodological and technical decisions. The authors independently checked the implementation and experimental results, validated the synthesized data against the corresponding planning problems, verified the cited literature and mathematical arguments, and reviewed the complete manuscript. The authors take full responsibility for all data, code, claims, derivations, results, and text presented in this work.

REFERENCES

Vidal Alcázar, Daniel Borrajo, Susana Fernández, and Raquel Fuentetaja. Revisiting regression in planning. In *International Joint Conference on Artificial Intelligence*, pp. 2254–2260, 2013.

Vidal Alcázar, Susana Fernández, and Daniel Borrajo. Analyzing the impact of partial states on duplicate detection and collision of frontiers. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 24, pp. 350–354, 2014. doi: 10.1609/icaps.v24i1.13677.

Rafael Vales Bettker, Pedro P. Minini, André G. Pereira, and Marcus Ritt. Understanding sample generation strategies for learning heuristic functions in classical planning. *Journal of Artificial Intelligence Research*, 80:243–271, 2024. doi: 10.1613/jair.1.15742.

Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33, 2001.

Zhongxuan Cai, Minglong Li, Wanrong Huang, and Wenjing Yang. BT expansion: A sound and complete algorithm for behavior planning of intelligent robots with behavior trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6058–6065, 2021. doi: 10.1609/aaai.v35i7.16755.

Dillon Z. Chen, Sylvie Thiébaux, and Felipe Trevizan. Learning domain-independent heuristics for grounded and lifted planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 20078–20086, 2024a. doi: 10.1609/aaai.v38i18.29986.

Dillon Z. Chen, Felipe Trevizan, and Sylvie Thiébaux. Return to tradition: Learning reliable heuristics with classical machine learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pp. 68–76, 2024b. doi: 10.1609/icaps.v34i1.31462.

Xinglin Chen, Yishuai Cai, Minglong Li, Yunxin Mao, Zhou Yang, Wenjing Yang, Weixia Xu, and Ji Wang. Btpg: A platform and benchmark for behavior tree planning in everyday service robots. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 8715–8723, 2025. doi: 10.24963/ijcai.2025/969.

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

Stefan Edelkamp. Planning with pattern databases. In *Proceedings of the Sixth European Conference on Planning*, pp. 13–24, 2001.

Patrick Ferber, Malte Helmert, and Jörg Hoffmann. Neural network heuristics for classical planning: A study of hyperparameter space. In *Proceedings of the 24th European Conference on Artificial Intelligence*, pp. 2346–2353, 2020. doi: 10.3233/FAIA200364.

Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

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.

Patrik Haslum. Improving heuristics through relaxed search: An analysis of tp4 and HSP$^*_a$ in the 2004 planning competition. *Journal of Artificial Intelligence Research*, 25:233–267, 2006. doi: 10.1613/jair.1885.

Patrik Haslum and Héctor Geffner. Admissible heuristics for optimal planning. In *International Conference on Artificial Intelligence Planning Systems*, pp. 140–149, 2000.

Patrik Haslum, Blai Bonet, and Héctor Geffner. New admissible heuristics for domain-independent planning. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pp. 1163–1168, 2005.

Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 19, pp. 162–169, 2009. doi: 10.1609/icaps.v19i1.13370.

Malte Helmert and Gabriele Röger. How good is almost perfect? In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 944–949, 2008.

Malte Helmert, Patrik Haslum, and Jörg Hoffmann. Flexible abstraction heuristics for optimal sequential planning. In *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling*, pp. 176–183, 2007.

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):16:1–16:63, 2014. doi: 10.1145/2559951.

Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

Chao Lei and Nir Lipovetzky. Width-based backward search. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, pp. 219–224, 2021.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103 (1):127–152, 2005. doi: 10.1007/s10107-004-0552-5.

Silviu Pitis, Harris Chan, Kiarash Jamali, and Jimmy Ba. An inductive bias for distances: Neural nets that respect the triangle inequality. In *International Conference on Learning Representations*, 2020.

Florian Pommerening, Gabriele Röger, and Malte Helmert. Getting the most out of pattern databases for classical planning. In *International Joint Conference on Artificial Intelligence*, pp. 2357–2364, 2013.

Martín Pozo, Álvaro Torralba, and Carlos Linares López. When CEGAR meets regression: A love story in optimal classical planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 20238–20246, 2024. doi: 10.1609/aaai.v38i18.30004.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8494–8502, 2018. doi: 10.1109/CVPR.2018.00886.

Jendrik Seipp, Thomas Keller, and Malte Helmert. Saturated cost partitioning for optimal classical planning. *Journal of Artificial Intelligence Research*, 67:129–167, 2020.

William Shen, Felipe Trevizan, and Sylvie Thiébaux. Learning domain-independent planning heuristics with hypergraph networks. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pp. 574–584, 2020. doi: 10.1609/icaps.v30i1.6754.

Tongzhou Wang and Phillip Isola. Improved representation of asymmetrical distances with interval quasimetric embeddings. In *Workshop on Symmetry and Geometry in Neural Representations (NeurReps) at the Conference on Neural Information Processing Systems (NeurIPS)*, 2022. URL `https://openreview.net/forum?id=KRiST_rzkGl`. Proceedings Track.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36411–36430, 2023.

A Construction and Properties of the Partial-State Heuristic

A.1 Preprocessing and Heuristic Evaluation

Algorithm 1 summarizes PS-IQE preprocessing and evaluation. Let $\mathcal{S}$ be the set of complete states, $\mathcal{P}$ the set of ground facts, $z : \mathcal{S} \to \mathbb{R}^{k\times l}$ the learned embedding map, and $m \geq 1$ the subset order. For each fact subset T, preprocessing stores the coordinate-wise lower bound $\underline{z}_T$ of the embeddings over $\mathcal{S}_T$ as $\mathcal{Z}[T] = \underline{z}_T$. Given a complete state x and a partial state $c \subseteq \mathcal{P}$, evaluation takes the coordinate-wise maximum of the relevant stored bounds and returns $h^m_{\text{PS-IQE}}(x, c)$. In regression search, $x = s_0$. The notation for coordinate-wise operations is the same as in the main text.

Algorithm 1 Construction and evaluation of the PS-IQE heuristic

procedure PREPROCESS($\mathcal{S}, \mathcal{P}, z, m$)
  $\mathcal{Z} \leftarrow \varnothing$
  for all $T \subseteq \mathcal{P}$ such that $1 \leq |T| \leq m$ **do**
    $\mathcal{S}_T \leftarrow \{y \in \mathcal{S} : T \subseteq y\}$
    if $\mathcal{S}_T = \emptyset$ **then**
      $\mathcal{Z}[T] \leftarrow \boldsymbol{\infty}_{k\times l}$
    else
      $\mathcal{Z}[T] \leftarrow \bigwedge_{y\in\mathcal{S}_T} z(y)$
    end if
  end for
  return $\mathcal{Z}$
end procedure
procedure EVALUATE($x, c, z, m, \mathcal{Z}$)
  if $c \subseteq x$ **then return** 0
  $r \leftarrow \min(m, |c|)$; $\mathcal{T}_r(c) \leftarrow \{T \subseteq c : |T| = r\}$
  if $\mathcal{Z}[T] = \boldsymbol{\infty}_{k\times l}$ for some $T \in \mathcal{T}_r(c)$ **then return** $+\infty$
  return $d_{\text{IQE}}\left(z(x), \bigvee_{T\in\mathcal{T}_r(c)} \mathcal{Z}[T]\right)$
end procedure

A.2 Monotonicity of IQE in Its Second Argument

The IQE quasimetric has the following monotonicity.

Lemma (Monotonicity of the IQE quasimetric in its second argument (Lemma 1, restated))**.** *For any fixed u, if $v \preceq v'$, then $d_{\text{IQE}}(u, v) \leq d_{\text{IQE}}(u, v')$.*

Proof. If $v \preceq v'$, then, for every i, the interval union in equation 3 for (u, v) is contained in that for (u, v'). Monotonicity of the Lebesgue measure therefore gives $q_i(u, v) \leq q_i(u, v')$. These inequalities are preserved by summation, the maximum, the mean, and the nonnegative combination of the latter two used by IQE-maxmean. Hence $d_{\text{IQE}}(u, v) \leq d_{\text{IQE}}(u, v')$ for both IQE formulations.

A.3 Lower Bound for Partial States

This monotonicity gives the following lower bound for the PS-IQE heuristic.

Proposition (Lower bound on the learned partial-state distance (Proposition 1, restated))**.** *For every complete state x and partial state c with $\mathcal{S}_c \neq \emptyset$,*

$$h^m_{\text{PS-IQE}}(x, c) \leq H_\theta(x, c).$$

Proof. If $c = \emptyset$, then $h^m_{\text{PS-IQE}}(x, c) = 0$. Otherwise, for any $y \in \mathcal{S}_c$ and $T \in \mathcal{T}_{\leq m}(c)$, we have $T \subseteq c \subseteq y$, and hence $y \in \mathcal{S}_T$. Equation 7 therefore gives $\underline{z}_T \preceq z(y)$. Taking the coordinate-wise maximum over T yields $z^{(m)}(c) \preceq z(y)$. Lemma 1 then gives

$$h^m_{\text{PS-IQE}}(x, c) \leq d_{\text{IQE}}(z(x), z(y)) = d_\theta(x, y).$$

Since this inequality holds for every $y \in \mathcal{S}_c$, minimizing its right-hand side over y proves the result. The construction does not require partial states during training, and the order-m relaxed lower corner need not correspond to the embedding of any state in $\mathcal{S}$. Increasing m can only tighten this relaxation. When $m \geq |c|$, $z^{(m)}(c)$ coincides with the exact lower corner of the completion embeddings.

A.4 Dominance over Scalar-Level Combination

For comparison, consider a scalar-level alternative

$$h^m_{\text{scalar}}(x, c) = \max_{T \in \mathcal{T}_{\leq m}(c)} d_{\text{IQE}}(z(x), \underline{z}_T).$$

It computes the quasimetric distance to the coordinate-wise lower bound for each fact subset and then takes the maximum.

Proposition (Dominance over scalar-level combination (Proposition 2, restated))**.** *For every complete state x and nonempty partial state c with $\mathcal{S}_c \neq \emptyset$,*

$$h^m_{\text{scalar}}(x, c) \leq h^m_{\text{PS-IQE}}(x, c).$$

Proof. By equation 8, $\underline{z}_T \preceq z^{(m)}(c)$ for every $T \in \mathcal{T}_{\leq m}(c)$. Lemma 1 therefore gives

$$d_{\text{IQE}}(z(x), \underline{z}_T) \leq d_{\text{IQE}}(z(x), z^{(m)}(c)).$$

Taking the maximum over $T \in \mathcal{T}_{\leq m}(c)$ yields equation 12.

The inequality can be strict, since different fact subsets may provide tighter bounds for different coordinates. Their coordinate-wise maximum retains these complementary improvements before d_{IQE} is evaluated. Proposition 2 compares only two heuristic constructions based on the same latent subset bounds. The comparison does not extend to h^m, whose subset estimates are computed under a different relaxation.

A.5 Redundancy of Lower-Order Subsets

The definition of the order-m relaxed lower corner uses all nonempty fact subsets of size at most m. The same lower corner is obtained using only subsets of the largest applicable size. This simplifies evaluation without changing the heuristic value.

Lemma 3 (Redundancy of lower-order subsets)**.** *Let c be a nonempty partial state with $\mathcal{S}_c \neq \emptyset$, and let $r = \min(m, |c|)$. Then*

$$\bigvee_{T \in \mathcal{T}_{\leq m}(c)} \underline{z}_T = \bigvee_{T \in \mathcal{T}_r(c)} \underline{z}_T, \qquad \mathcal{T}_r(c) = \{T \subseteq c : |T| = r\}. \tag{15}$$

Proof. Since $\mathcal{T}_r(c) \subseteq \mathcal{T}_{\leq m}(c)$, the right-hand side is coordinate-wise no greater than the left-hand side. For the reverse inequality, consider any $T \in \mathcal{T}_{\leq m}(c)$. If $|T| < r$, extend T to some $T' \subseteq c$ with $|T'| = r$. Since $T \subseteq T'$, we have $\mathcal{S}_{T'} \subseteq \mathcal{S}_T$. Both sets are nonempty because $\mathcal{S}_c \subseteq \mathcal{S}_{T'}$. Hence

$$\underline{z}_T = \bigwedge_{y \in \mathcal{S}_T} z(y) \preceq \bigwedge_{y \in \mathcal{S}_{T'}} z(y) = \underline{z}_{T'}.$$

Thus every coordinate-wise lower bound in the embedding space contributed by a subset smaller than r is dominated by one contributed by a subset of size r. Removing the smaller subsets leaves the coordinate-wise maximum unchanged.

This is the analogue of the standard h^m construction in the embedding space. Evaluating a fixed partial state c requires only subsets of size $\min(m, |c|)$.

B Search Guarantees

B.1 Optimal-Plan Decomposition

We first establish the optimal-plan decomposition used in the proof of Theorem 1.

Lemma (Optimal-plan decomposition (Lemma 2, restated))**.** *Let π^* be an optimal plan with cost C^*, and let c lie on the regression path induced by π^*. Let $g^*(c)$ denote the minimum cost of a regression path from G to c. Then*

$$g^*(c) + H^*(s_0, c) = C^*.$$

Proof. Let $C_{\text{pre}}(c)$ and $C_{\text{suf}}(c)$ denote the costs of the prefix and suffix, respectively, obtained by splitting the optimal plan at c. The prefix leads from s_0 to a state satisfying c, while the suffix leads from that state to a goal state and induces a regression path from G to c. Therefore,

$$H^*(s_0, c) \leq C_{\text{pre}}(c), \qquad g^*(c) \leq C_{\text{suf}}(c).$$

Since

$$C_{\mathrm{pre}}(c) + C_{\mathrm{suf}}(c) = C^*,$$

it follows that

$$H^*(s_0, c) + g^*(c) \leq C^*.$$

Conversely, consider the action sequence associated with a minimum-cost regression path from G to c. By the soundness of regression, executing this sequence from any state satisfying c reaches a goal state. Concatenating it with a minimum-cost path from s_0 to a state satisfying c therefore yields a valid plan of cost $H^*(s_0, c) + g^*(c)$. Hence

$$C^* \leq H^*(s_0, c) + g^*(c),$$

and equality follows.

B.2 Bounded Suboptimality under Local-Error Bounds

For an edge $e = (u, v)$, write $d_\theta(e) = d_\theta(u, v)$.

Theorem (Bounded suboptimality under local-error bounds (Theorem 1, restated))**.** *Consider regression A* guided by PS-IQE, and let π^* be an optimal plan of cost C^* whose corresponding regression path is preserved by the pruning rules used in this search. Suppose that, for every transition $e = (u, v) \in E$, the learned distance satisfies*

$$d_\theta(u, v) \leq w(e) + \Delta(e), \qquad \Delta(e) \geq 0.$$

If the search reopens an expanded node whenever a lower g-value is found and returns a plan of cost $\widehat{C}$, then

$$\widehat{C} \leq C^* + \sum_{e \in \pi^*} \Delta(e).$$

Proof. Write the regression path induced by π^* as $c_0, c_1, \ldots, c_k$, ordered in the direction of regression search, where $c_0 = G$. Initially, $c_0 = G$ is in OPEN with $g(c_0) = g^*(c_0) = 0$. Whenever c_i is expanded with $g(c_i) = g^*(c_i)$, its successor c_{i+1} is generated with $g(c_{i+1}) = g^*(c_{i+1})$ and reopened if it was previously expanded with a higher g-value. Thus, until a solution node is selected, OPEN contains a node c on this path with its optimal g-value. Let $\pi^*_{\mathrm{pre}}(c)$ denote the prefix of π^* corresponding to c, and let y_c be the complete state reached by executing this prefix from s_0. By construction, y_c satisfies c, and Lemma 2 shows that this prefix has cost $H^*(s_0, c)$. Proposition 1, the triangle inequality, and the edge-wise bound therefore give

$$\begin{aligned} h^m_{\text{PS-IQE}}(s_0, c) &\leq d_\theta(s_0, y_c) \\ &\leq \sum_{e \in \pi^*_{\mathrm{pre}}(c)} d_\theta(e) \\ &\leq H^*(s_0, c) + \sum_{e \in \pi^*_{\mathrm{pre}}(c)} \Delta(e). \end{aligned}$$

Together with Lemma 2, this inequality gives

$$\begin{aligned} f(c) &= g^*(c) + h^m_{\text{PS-IQE}}(s_0, c) \\ &\leq g^*(c) + H^*(s_0, c) + \sum_{e \in \pi^*_{\mathrm{pre}}(c)} \Delta(e) \\ &= C^* + \sum_{e \in \pi^*_{\mathrm{pre}}(c)} \Delta(e) \\ &\leq C^* + \sum_{e \in \pi^*} \Delta(e). \end{aligned}$$

Let c_{sol} be the terminal regression node selected by A*, so that $s_0 \models c_{\mathrm{sol}}$. Proposition 1 and nonnegativity imply $h^m_{\text{PS-IQE}}(s_0, c_{\mathrm{sol}}) = 0$. Since A* selects a node of minimum f-value from OPEN,

$$\widehat{C} = g(c_{\mathrm{sol}}) = f(c_{\mathrm{sol}}) \leq f(c).$$

This proves the theorem.

B.3 Additive and Multiplicative Bounds

Corollary (Corollary 1, restated)**.** *Suppose the assumptions in Theorem 1 hold, and let L^* denote the number of actions in the optimal plan π^*.*

1. *If $d_\theta(u,v) \leq w(e) + \delta$ for every transition $e = (u,v) \in E$, where $\delta \geq 0$, then*
$$\widehat{C} \leq C^* + \delta L^*.$$
2. *If $d_\theta(u,v) \leq (1+\epsilon)w(e)$ for every transition $e = (u,v) \in E$, where $\epsilon \geq 0$, then*
$$\widehat{C} \leq (1+\epsilon)C^*.$$

Under uniform per-transition bounds, Corollary 1 gives corresponding bounds on the cost of the returned plan. The additive relaxation gives an absolute error bound proportional to the number of actions in the optimal plan. The multiplicative relaxation, based on the relative violation in equation 4, gives a relative suboptimality bound. Both follow from Theorem 1 by setting $\Delta(e) = \delta$ and $\Delta(e) = \epsilon w(e)$, respectively. The multiplicative bound also holds with cost-aware subsumption pruning: a partial state c may be pruned if an expanded partial state $c' \subseteq c$ satisfies $g(c') \leq g(c)$ (Alcázar et al., 2014). Appendix B.4 proves this claim.

B.4 Guarantee under Subsumption Pruning

This appendix proves that the multiplicative bound in Corollary 1 remains valid with cost-aware subsumption pruning. We use the notation of the main text. We assume sound and complete regression for the propositional STRIPS setting considered in the paper, strictly positive action costs, exact duplicate detection with reopening, and no other pruning rule that removes every regression path induced by an optimal plan.

A generated partial state c may be pruned if an expanded partial state c' satisfies

$$c' \subseteq c \qquad \text{and} \qquad g(c') \leq g(c). \tag{16}$$

B.4.1 Preservation of an Optimal Plan

Lemma 4 (Cost-safe subsumption)**.** *Suppose that a generated partial state c, reached with cost $g(c)$, lies on the regression path induced by an optimal plan and is pruned according to equation 16. Then some optimal plan induces a regression path through c', and $g(c') = g^*(c')$.*

Proof. Split the original optimal plan at c, and let y be the complete state reached by its prefix. Since $y \in \mathcal{S}_c$ and $c' \subseteq c$, we also have $y \in \mathcal{S}_{c'}$. By soundness of regression, the action sequence associated with the regression path from G to c' takes any state satisfying c', including y, to a goal state. Replacing the original plan suffix with this sequence gives a valid plan of no greater cost, because $g(c') \leq g(c)$. Since the original plan is optimal, the replacement plan also has cost C^*.

Let $C_{\mathrm{pre}}(c')$ be the cost of its prefix. We have

$$C_{\mathrm{pre}}(c') + g(c') = C^*, \qquad C_{\mathrm{pre}}(c') \geq H^*(s_0, c'), \qquad g(c') \geq g^*(c').$$

Lemma 2 gives $H^*(s_0, c') + g^*(c') = C^*$. Both inequalities must therefore be equalities. Hence $g(c') = g^*(c')$, and the replacement plan induces an optimal regression path through c'.

B.4.2 Preservation of an Optimal Frontier

Lemma 5 (Optimal frontier under subsumption pruning)**.** *Until A* selects a solution node, OPEN contains a partial state c on the regression path induced by some optimal plan such that $g(c) = g^*(c)$.*

Proof. Initially, G is in OPEN with $g(G) = g^*(G) = 0$. Suppose that OPEN contains a partial state c with the stated property. If c is not selected, it remains in OPEN. If c is selected and is not a solution node, regression generates the next partial state d on the same optimal path. The generated path to d corresponds to the associated plan suffix, so Lemma 2 implies that its cost is $g^*(d)$. If d has not already been expanded with this value and is not pruned, duplicate detection places it in OPEN or reopens it.

If d was already expanded with $g(d) = g^*(d)$, we continue from the next partial state on that optimal path. If d is pruned, Lemma 4 provides another optimal path through the expanded partial state d' that subsumes d, with $g(d') = g^*(d')$, and we continue from the next partial state generated when d' was expanded. The same argument applies whenever the traced state has already been expanded or pruned.

An optimal plan cannot contain a cycle because all action costs are strictly positive. Exact duplicates are handled separately, so every genuine subsumption step replaces a partial state by a strict subset and decreases its number of facts. Because the regression graph is finite, this tracing process must reach either a solution node or an unexpanded partial state after finitely many steps. In the latter case, that state is in OPEN with its optimal g-value. Induction over A* expansions proves the claim.

B.4.3 Search Bound with Subsumption Pruning

Theorem 2 (Multiplicative bound under subsumption pruning)**.** *Suppose that, for every transition $e = (u, v) \in E$,*

$$d_\theta(u, v) \le (1 + \epsilon)w(e), \qquad \epsilon \ge 0. \tag{17}$$

If A* uses the subsumption rule in equation 16 and returns a plan of cost $\widehat{C}$, then

$$\widehat{C} \le (1 + \epsilon)C^*. \tag{18}$$

Proof. By Lemma 5, before a solution node is selected, OPEN contains a partial state c on the regression path induced by an optimal plan π, with $g(c) = g^*(c)$. Let $\pi_{\mathrm{pre}}(c)$ be the corresponding prefix of π, and let y_c be the complete state reached by this prefix. Then $y_c \in \mathcal{S}_c$, and Lemma 2 shows that the prefix has cost $H^*(s_0, c)$. Proposition 1, the triangle inequality, and equation 17 give

$$\begin{aligned} f(c) &= g^*(c) + h^m_{\text{PS-IQE}}(s_0, c) \\ &\le g^*(c) + (1 + \epsilon)H^*(s_0, c) \\ &= C^* + \epsilon H^*(s_0, c) \\ &\le (1 + \epsilon)C^*. \end{aligned}$$

A* selects a node of minimum f-value from OPEN. At a solution node, s_0 satisfies the corresponding partial state, so its learned partial-state distance is zero. Proposition 1 and nonnegativity therefore give a heuristic value of zero. The node's f-value equals the returned plan cost $\widehat{C}$, and equation 18 follows.

C Experimental and Implementation Details

C.1 Planning Tasks and State Enumeration

We use the official initial states and goal conditions of the IPC `logistics-4-0` instance and the first five STRIPS instances of the IPC 1998 Gripper domain. Logistics has 941,192 legal complete states, 7,663,992 directed transitions, 54 grounded facts, 164 actions, and optimal plan cost 20. The Gripper instances have 4, 6, 8, 10, and 12 balls. Their state spaces range from 256 states and 896 transitions to 376,832 states and 1,654,784 transitions; their optimal plan costs are 11, 17, 23, 29, and 35. Both domains use unit-cost actions.

The VirtualHome-based task called `vhm_7o` in the main text is the `vhm7_2surffar` scenario in the experiment code. It uses the action model and object vocabulary of BTPG's VirtualHome environment (Chen et al., 2025). Seven movable objects—`milk`, `chicken`, `cupcake`, `wine`, `book`, `juice`, and `cereal`—start on two surfaces, `kitchentable` and `desk`. The goal places all seven objects inside `fridge` and closes it. The `tv` is retained as an object outside the goal. Fixed locations have manually assigned coordinates: `kitchentable` at $(0, 0)$, `desk` at $(0, 20)$, `fridge` at $(0, 7)$, and `tv` at $(10, 4)$. The cost of `Walk` is the Manhattan distance between locations. The task has 361,584 legal complete states, 1,813,752 directed transitions, 46 grounded facts, 104 actions, and optimal plan cost 203.

Each evaluated transition graph forms a single strongly connected component. We enumerate legal complete states satisfying the planning invariants and construct the PS-IQE fact-subset table from those states.

C.2 Training Architecture and Objective

For the reported PS-IQE models, complete states are encoded as multi-hot vectors. The encoder is an MLP with two hidden layers of width 512 and ReLU activations. The IQE-sum head has dimension 512, divided into 32 components. We train with the relative violation in equation 4, a batch size of 512, a learning rate of 2×10^{-4}, and $\epsilon = 0.1$. During the first 60% of training, the local objective is annealed from the mean squared relative violation to the squared LSE. We train three models independently, using seeds 42, 7, and 123.

C.3 Log-Sum-Exp Objective

For the LSE defined in equation 5,

$$\max_{e \in B} v_+(e) \leq \mathrm{LSE}_{\beta_{\mathrm{LSE}}}(B) \leq \max_{e \in B} v_+(e) + \frac{\log |B|}{\beta_{\mathrm{LSE}}}.$$

Thus the LSE upper-bounds every violation in B and exceeds the largest violation by at most $\log |B| / \beta_{\mathrm{LSE}}$.

We anneal the local objective from the mean squared relative violation to the squared LSE:

$$\mathcal{L}_{\mathrm{local}} = (1 - a)\, \mathbb{E}_{e \in B}[v_+(e)^2] + a\, \mathrm{LSE}_{\beta_{\mathrm{LSE}}}(B)^2,$$

where $a \in [0, 1]$ increases from zero to one during the first 60% of training and remains one thereafter. The mean-squared term provides gradients across violating transitions early in training, while the LSE term increasingly emphasizes the largest violations in each minibatch. Once $a = 1$, the local objective is the squared LSE. A small sampled training loss does not establish the edge-wise premise of Theorem 1; the measured maximum violation must be checked over the complete edge set.

C.4 Search and Timing Protocol

Regression A* performs duplicate detection and permits reopening, but does not use subsumption pruning. Every method uses the same invariant-based consistency checker to discard regressed partial states that violate counting invariants. Every returned action sequence is checked by forward execution from the initial state. The official initial state and goal condition define one planning task per benchmark instance. The main comparisons use a 120-second time limit per task. All evaluated action costs are positive, as required by equation 4.

Each task starts with empty symbolic-successor and heuristic-query caches. Total runtime includes search and initial-state-dependent preprocessing but excludes transition-system-level preprocessing and offline model training. The reported preprocessing time isolates initialization tied to a new initial state. The Logistics and VirtualHome-based comparisons ran on the same server, with two Intel Xeon Platinum 8481C processors, 16 vCPUs allocated to the container, and an NVIDIA RTX 4090D GPU. Each table was produced in one consecutive run, but the two tables were produced separately. Runtime comparisons across the two tasks are therefore indicative only.

C.5 Baselines and Metrics

Tables 1 and 2 compare UCS, h^1 (Bonet & Geffner, 2001), h^2 and additive h^2 (Haslum & Geffner, 2000; Haslum et al., 2005), h^{FF} (Hoffmann & Nebel, 2001), HSPr (Bonet & Geffner, 2001), SysPDB-UCP$_{\leq 2}$ and SysPDB-SCP$_{\leq 2}$ (Pommerening et al., 2013; Seipp et al., 2020), WL-GOOSE (Chen et al., 2024b), and the unscaled and $1/1.1$-scaled PS-IQE heuristics. Alongside A* results, we report greedy best-first search (GBFS) for h^{FF} and WL-GOOSE and weighted A* with $w = 5$ for HSPr. UCS, h^1, h^2, additive h^2, and both SysPDB variants are admissible; h^{FF}, HSPr, and WL-GOOSE are not generally admissible. Additive h^2 uses the loss-based per-goal-atom action assignment of Haslum et al. (2005). Both SysPDB variants use systematic patterns of size at most 2, combined with uniform and saturated cost partitioning respectively, and are anchored at s_0 rather than at the fixed goal used in forward search, since s_0 is the quantity that stays fixed across regression nodes. WL-GOOSE is our own Python/NumPy reimplementation, since the reference `wlplan` package has no Windows build, and fits a Gaussian process posterior mean under a dot-product kernel, the configuration reported as strongest in the original paper. The triangle in the PS-IQE rows indicates a conditional guarantee: if the relative bound holds on every transition, the scaled heuristic is admissible and A* with the unscaled heuristic returns a solution costing at most $1.1C^*$. The reported metrics are expanded partial-state nodes, total runtime, per-task preprocessing, and admissibility. PS-IQE results use three independently trained models; other methods are evaluated once per task. For the VirtualHome-based task, the measured edge-wise bound holds for all three PS-IQE models, whereas it does not hold at $\epsilon = 0.1$ for any of the three Logistics models.

C.6 Cross-Initial-State Amortization Protocol

The amortization experiment is the exception to the cold-start protocol. It uses `vhm_7o`, its fixed goal, and the seed-42 PS-IQE model. We sample 100 legal initial states uniformly without replacement using random seed 42. In the shared-cache run, searches retain the symbolic successor cache keyed by the regressed partial state c and the PS-IQE cache of the aggregated embedding $z^{(m)}(c)$. In

Table 3: Distribution of the relative edge violation on the Logistics instance. Each row evaluates one training seed over the complete edge set.

Seed	$\widehat{\epsilon}_{\max}$	p99.9	p99	p95
42	0.1567	0.0811	0.0741	0.0684
7	0.1482	0.0801	0.0718	0.0657
123	0.1589	0.0826	0.0738	0.0676

the cold-start control, both caches are cleared before each search. Both runs process the same initial states, with a 90-second time limit per search, and expand the same nodes. The transition-system-level PS-IQE table is constructed before both runs; its construction time is excluded. Symbolic-successor cache hits and misses are counted over expanded partial states. Appendix D.3 reports the results.

C.7 Computational Complexity and Storage

Let $P = |\mathcal{P}|$ be the number of ground facts, $D = k \times l$ the number of projected IQE coordinates, and $r = \min(m, |c|)$. Storing all entries up to subset order m requires $\mathcal{O}(D \sum_{j=1}^{m} \binom{P}{j}) = \mathcal{O}(P^m D)$ space for fixed m. Evaluating a partial state c takes $\mathcal{O}(\binom{|c|}{r} D)$ time to aggregate the relevant entries, with $\mathcal{O}(P^m D)$ as the worst case. Thus online evaluation depends on the size of c, not the number of complete states.

D Additional Experimental Results

This appendix provides the complete-edge violation measurements for Logistics, the Gripper ladder, and the `vhm_o` ladder; further details of the Gripper case study; and the cross-initial-state cache results for the VirtualHome-based task.

D.1 Regression-Search Comparison and Edge Violations

Table 1 compares methods on the official IPC `logistics-4-0` task. The PS-IQE rows give the mean and standard deviation across three independently trained models; the other methods are evaluated once. Unscaled $h^2_{\text{PS-IQE}}$ expands about 5.5 times fewer nodes than h^2 and fewer than additive h^2 and SysPDB-UCP$_{\leq 2}$, but slightly more than SysPDB-SCP$_{\leq 2}$. The h^{FF} and HSPr baselines have shorter runtimes. This single-task comparison does not establish performance across the Logistics domain.

For an edge $e = (u, v)$, define the normalized learned distance, relative violation, and maximum relative violation by

$$r_\theta(e) = \frac{d_\theta(u, v)}{w(e)}, \qquad \widehat{\epsilon}_\theta(e) = [r_\theta(e) - 1]_+, \qquad \widehat{\epsilon}_{\max} = \max_{e \in E} \widehat{\epsilon}_\theta(e).$$

Table 3 reports the distribution of $\widehat{\epsilon}_\theta(e)$ over all 7,663,992 Logistics edges for the three PS-IQE models. For every seed, at least 99.9% of edges have a relative violation no greater than 0.0826. However, $\widehat{\epsilon}_{\max}$ is 0.1567, 0.1482, and 0.1589, respectively, with mean and standard deviation 0.1546 ± 0.0046. Thus none of the three models satisfies the complete-edge premise of Theorem 1 at $\epsilon = 0.1$.

The same measurement on the Gripper and `vhm_o` ladders shows the opposite pattern: every instance satisfies $\widehat{\epsilon}_{\max} < 0.1$ across all three seeds (Tables 4 and 5). Logistics is the only instance in our experiments where the measured relaxation exceeds this threshold.

Table 4: Maximum relative edge violation across the Gripper ladder. Mean $\pm$ standard deviation across three seeds.

Instance	N	$\widehat{\epsilon}_{\max}$
$n = 4$	256	0.0807 ± 0.0010
$n = 6$	1,856	0.0826 ± 0.0022
$n = 8$	11,776	0.0808 ± 0.0013
$n = 10$	68,608	0.0819 ± 0.0026
$n = 12$	376,832	0.0822 ± 0.0002

The vhm_o instances are a family of tasks built from BTPG's VirtualHome action model (Chen et al., 2025), with $O \in \{3, 4, 5, 6, 7\}$ objects to be placed in a refrigerator that is then closed, giving a goal of size $O + 1$ and optimal cost $C^* \in \{103, 119, 134, 163, 203\}$ for $O = 3, \ldots, 7$. Action costs are heterogeneous, drawn from $\{3, 5, 6, 7, 8, 10, 13, 14, 20, 26\}$, with the largest values coming from walking between two surfaces placed far apart; the state-space size grows from 1,584 states at $O = 3$ to 361,584 states at $O = 7$. vhm_7o, used in Table 2 and the amortization experiment, is the largest instance in this family.

Table 5: Maximum relative edge violation across the vhm_o ladder. Mean $\pm$ standard deviation across three seeds.

Instance	N	$\widehat{\epsilon}_{\max}$
vhm_3o	1,584	0.0824 ± 0.0013
vhm_4o	6,480	0.0852 ± 0.0045
vhm_5o	25,488	0.0857 ± 0.0018
vhm_6o	97,200	0.0902 ± 0.0025
vhm_7o	361,584	0.0922 ± 0.0006

D.2 Gripper Case Study: Scaling with Serial Subgoals

We use Gripper as a mechanism-focused case study, rather than an overall planner comparison. The five official IPC 1998 instances have $n \in \{4, 6, 8, 10, 12\}$ balls, all of which must reach the destination room. The number of goal atoms grows with n, while each goal condition has exactly two satisfying complete states. Thus the completion-set size stays fixed as the number of serial subgoals and the optimal cost increase. The state-space size grows from 256 to 376,832 states, and $C^* = 6\lceil n/2 \rceil - 1$.

At the root of regression search, the classical values remain $h^2 = 4$ and $h^3 = 8$ for every instance. Their ratios to C^* fall from $4/11 = 0.364$ and $8/11 = 0.727$ at four balls to $4/35 = 0.114$ and $8/35 = 0.229$ at twelve balls. In contrast, the PS-IQE estimates increase with the task cost at both subset orders. Across the five instances, the three-seed mean of $h^2_{\text{PS-IQE}}/C^*$ lies between 0.880 and 0.934, and that of $h^3_{\text{PS-IQE}}/C^*$ lies between 0.930 and 0.966. Within each instance, the range across the three seeds is at most 0.31 for either PS-IQE estimate. Figure 3 plots the three-seed mean estimates alongside h^2, h^3, and C^*.

Additive h^m does not strengthen h^m when its action-cost partition collapses to a single nonzero partition. In that case, zeroing the costs of actions outside the remaining partition cannot increase its h^m value, so $\sum_i h^m_{A_i} \leq h^m$ and the combined estimate $\max(h^m, \sum_i h^m_{A_i})$ equals h^m. This occurs in Gripper under the loss-based goal-atom partition used here: the ball-goal atoms are symmetric and rely on the same move, pick, and drop action structure, so the allocation places all nonzero action costs in one partition. The lack of improvement is therefore a property of this degenerate partitioning, not of additive h^m in general. Similar behavior can arise when interchangeable goal atoms share the same supporting actions and the cost-partitioning rule cannot separate their costs across subproblems.

This case study illustrates the effect of combining subset bounds in embedding space. At fixed orders two and three, PS-IQE tracks the accumulation of transport cost that the scalar maximum in h^m does not capture. The result does not imply that PS-IQE is globally admissible or that it prevents the exponential search growth known for Gripper (Helmert & Röger, 2008).

D.3 Amortization in BT Expansion

The PS-IQE fact-subset table and the symbolic successors of a regressed partial state do not depend on the initial state. The aggregated embedding $z^{(m)}(c)$ can also be cached by partial state c. We test the resulting reuse on vhm_7o with the fixed goal and 100 sampled initial states, comparing shared caches across searches with caches cleared before each search (Appendix C.6). By the 100th search, the cumulative symbolic-successor cache hit rate is 87.1%, with 138,261 distinct cached partial states across 1,071,466 expansions. Cumulative wall-clock time is 164.343 s with shared caches and 524.512 s with cold restarts, a $3.19\times$ speedup. These measurements exclude construction of the transition-system-level fact-subset table and cover one transition system and one start-state distribution.