PLAYING TO PAR: REINFORCEMENT LEARNING FOR PROVABLY OPTIMAL QUADRILATERAL BLOCK DECOMPOSITIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

A quadrilateral block decomposition of a planar domain is judged by whether it is complete, whether its elements are well shaped, and how many of its vertices are irregular. The last has a provable floor: the discrete Gauss–Bonnet identity enforces a lower bound on the total vertex irregularity of any all-quadrilateral mesh of a given domain purely based on its topology and corner angles. We train a reinforcement learning agent to build decompositions that reach this bound, which we call *par*. It acts directly on the mesh's half-edge data structure through local edits, with a policy network whose convolutions follow the mesh's own connectivity, so it applies unchanged to domains larger than any seen in training. The reward targets the floor directly, and it is sparse: random play reaches it on no domain with more than eight sides. We overcome this exploration barrier via behaviour cloning on optimal meshes that are trivial to construct, walked backward into demonstrations, before training it with PPO. On 96 held-out domains the agent produces an all-quadrilateral mesh on every one, a usable one on 95.7 on average, and a provably optimal one on 90; Gmsh's strongest configuration at the same element count completes 51, is usable on 38 and optimal on none, and even at three to fourteen times the elements never produces a more regular mesh. On 64 domains twice the training size the agent completes all, is usable on 62, and keeps a median excess over par below one against Gmsh's 39 at the same element count.

1 INTRODUCTION

A block decomposition is a coarse all-quadrilateral partition of a domain with as *regular* a vertex connectivity as the domain allows. Connectivity is crucial: a thin element can be smoothed or split, but a vertex of the wrong valence is a singularity that survives every refinement. Structured and multi-block discretisations, tensor-product bases and subdivision all rely on regular connectivity (Bommes et al., 2013; Armstrong et al., 2015); production meshers clean up valence after the fact (Kinney, 1997), and advancing fronts introduce more singularities than a domain needs, though topology makes some unavoidable (Armstrong et al., 2015; Fogg et al., 2018). Block decompositions are still built largely by hand or by heuristics, and often do not target optimality.

That optimality criterion exists. The discrete Gauss–Bonnet identity gives every domain a lower bound on the irregularity of any all-quadrilateral mesh of it, computed from the domain's corner angles and topology alone (Section 2). We call it *par*. It is computed once per domain before any mesh exists, and a decomposition that reaches it is provably optimal in its connectivity. That makes block decomposition an unusual reinforcement learning (RL) problem: the reward is sparse, but success carries a certificate.

The sparsity is severe. We pose the problem as an MDP whose state is a half-edge mesh and whose actions are four local operations (Section 3); uniform random play reaches par on about three percent of five- and six-sided polygons and on none with more than eight sides, and a policy trained from scratch rarely leaves that regime. We overcome this exploration barrier by behaviour cloning on known optimal trajectories, and continuing with reinforcement learning. The policy is a convolution on the half-edge structure (Section 4): each half-edge receives messages from its `next`, `previous` and `twin`, the pointers that define local topological connectivity. The input features

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.

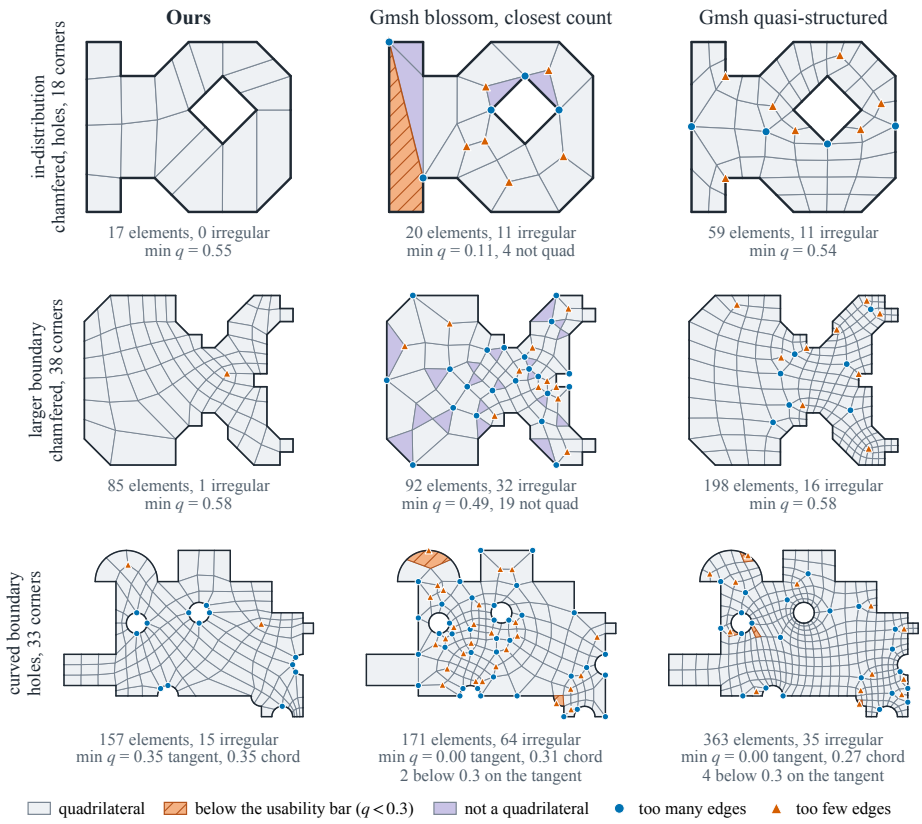


Figure 1: Our agent and Gmsh on the same three domains. *Top:* an in-distribution domain (chamfered, with a hole, 18 corners). *Middle:* a larger-boundary domain (38 corners), beyond any seen in training. *Bottom:* a curved domain with holes (33 corners); the agent was never trained on a curved boundary, and quality is given against the arc tangents and on chords (Section 6.4). Gmsh BLOSSOM is asked for our element count and returns the closest it can reach; QUASI-STRUCTURED runs at its natural size. Every irregular vertex is marked. Our top mesh has none, which the Gauss–Bonnet bound (Section 2) certifies as optimal; the middle one has one, against 32 and 16 for Gmsh, and the curved one 15, against 64 and 35. Domains are chosen by the rule of Appendix M, which shows one per family.

and state representation are designed to allow the agent to scale to larger meshes than seen during training. Unlike prior reinforcement learning for block decomposition (Narayanan et al., 2024), whose half-edge framework we build on and which edits an existing all-quadrilateral mesh toward ideal vertex degrees, our agent starts from the bare boundary, targets a bound computed from the domain, includes element quality in its reward, and overcomes the sparse reward by cloning. We also evaluate more comprehensively: on domains with holes, on domains twice the training size and on curved boundaries, against the production mesher Gmsh.

Contributions. (i) An action space and a policy network that operate directly on the half-edge data structure of any unstructured polygonal tiling. (ii) GEO2D, a seeded generator and scoring protocol of mechanical-part outlines with chamfers, fillets, and holes, released as a benchmark suite. (iii) A training recipe for the sparse reward: a shaped reward that targets par directly and is normalised so that returns share one scale across domain sizes; and optimal meshes that are trivial to construct, such as polyominoes, walked backward into verified demonstrations in the agent's action space. Cloning on them overcomes the exploration barrier that otherwise blocks PPO.

2 QUADRILATERAL BLOCK DECOMPOSITION AND ITS OPTIMALITY BOUND

A *domain* Ω is a bounded planar polygonal region whose boundary consists of finitely many disjoint simple closed polygons: one outer boundary and, optionally, H hole boundaries, whose vertices are the *corners* of Ω. A *quadrilateral block decomposition* of Ω is a conforming mesh $\mathcal{M} = (V, E, F)$ covering Ω in which every face is bounded by four edges. We write $\deg(v)$ for the number of edges incident on v, $B \subseteq E$ for the boundary edges, and $\chi = |V| - |E| + |F| = 1 - H$ for the Euler characteristic, which depends on Ω alone.

In a *regular* quadrilateral grid, each face presents an angle of $\pi/2$ at each corner, so the number of faces meeting at v is $k(v) = \operatorname{round}\big(\theta_v/(\pi/2)\big)$, where θ_v is the interior angle at v between the incident boundary edges, taken as 2π at an interior vertex. A cycle of k faces about an interior vertex uses k edges and a fan of k faces at a boundary vertex uses $k+1$, giving the *desired degree* $d(v)$ and the *irregularity* $I(\mathcal{M})$ of a mesh,

$$d(v) = \begin{cases} k(v), & v \in \operatorname{int}\Omega, \\ \max\{k(v)+1,\, 2\}, & v \in \partial\Omega, \end{cases} \qquad I(\mathcal{M}) = \sum_{v \in V} \big|\deg(v) - d(v)\big|, \tag{1}$$

the floor at 2 recording that a boundary vertex always carries its two boundary edges. A decomposition with $I(\mathcal{M}) = 0$ is fully structured, but not every domain admits one. How far from zero it must be is fixed by the discrete Gauss–Bonnet identity for quadrilateral meshes (Peng & Wonka, 2013; Peng et al., 2014). Write $g(v) = 4$ at an interior vertex and $g(v) = 3$ at a boundary vertex, the degree a vertex would want were no corner of Ω sited there.

For any conforming all-quadrilateral mesh of Ω the identity $\sum_{v \in V}(g(v) - \deg(v)) = 4\chi$ holds (Lemma 2, Appendix A): vertex, edge and face counts may differ arbitrarily between two decompositions of one domain, but this signed total cannot move. Separating the corner-dependent part of d from the uniform part gives the bound.

Theorem 1. *Let $c(\Omega) = \sum_v \big(d(v) - g(v)\big)$, summed over the corners of Ω. Every conforming all-quadrilateral mesh $\mathcal{M}$ of Ω satisfies*

$$I(\mathcal{M}) \;\geq\; \operatorname{par}(\Omega) \;:=\; \big|\, c(\Omega) + 4\chi \,\big|. \tag{2}$$

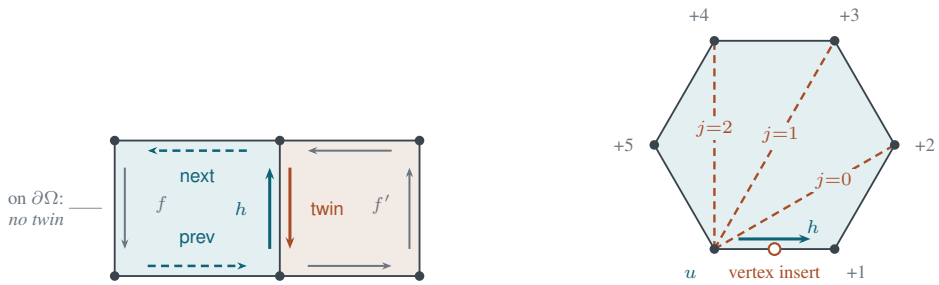
Away from the corners $d(v) = g(v)$, so the bound is the triangle inequality applied to the identity (proof in Appendix A). Both terms of equation 2 are fixed before meshing begins — $c(\Omega)$ is read off the corner angles, χ off the topology — so $\operatorname{par}(\Omega)$ is computed once per domain and serves as an exact termination test: a decomposition that reaches it is provably optimal in its connectivity, although the bound does not guarantee that such a decomposition exists. When $\theta_v/(\pi/2)$ is a half-integer the rounding in equation 1 is a tie between two equally admissible corner treatments; such a corner carries both, with par minimised over the choice (Appendix A).

3 THE DECOMPOSITION MDP

3.1 STATES AND OBSERVATIONS

We represent a mesh by a doubly connected edge list (DCEL): every edge carries two oppositely oriented half-edges, and all connectivity is expressed through three pointers per half-edge — `next` and `previous`, which walk a face loop counter-clockwise, and `twin`, which crosses to the adjoining face (Figure 2a). Holes enter through a slit joining each hole boundary to the outer one, so a multiply connected domain is still a single face loop. A state of the MDP is the current mesh together with the half-edge on which the observation window is centred and the number of moves taken. An episode begins with Ω as a single face and ends when the mesh is solved or the move budget is exhausted.

The mesh grows during an episode, so the observation is restricted to a fixed-size *template* of n_T half-edges: the centre's face loop, then a breadth-first traversal of the `next`/`previous`/`twin` pointers. Three integer arrays give each slot's neighbours in the template, with two sentinel values distinguishing a neighbour that exists but fell outside the window from one that does not exist because the half-edge lies on $\partial\Omega$. Actions are restricted to the template; par, the scores and the termination test are computed on the whole mesh. Each slot carries ten features. *Topological:* the



(a) The half-edge representation

(b) The four moves available at h

Figure 2: **(a)** A mesh is stored as a doubly connected edge list: every edge carries two oppositely oriented half-edges, and all connectivity is expressed through next and prev, which walk a face loop counter-clockwise, and twin, which crosses to the adjoining face. A half-edge on $\partial\Omega$ has no twin, and the network is told so by a sentinel distinct from the one marking a neighbour that merely fell outside the window. **(b)** From a half-edge h with source u, three actions insert a chord to the vertex $j+2$ positions ahead in the face loop, cutting off a sub-face of $j+3$ sides; the fourth splits h's own edge with a new vertex. Chords leaving a 2-gon on either side are masked out.

desired and actual degree of the source and of the head vertex of the half-edge, and the number of sides of its face. *Geometric:* the half-edge's length divided by the template's median, and the interior angles at its source and head as multiples of $\pi/2$. *Flags:* whether the source vertex lies on $\partial\Omega$, and whether it is a corner of Ω. Degrees are clipped (vertices at eight, faces at fifteen) so that a transient high-valence vertex cannot dominate the input scale.

A local window cannot say how far the whole mesh is from par, so the observation also carries a *global vector* of seven scalars: the distance of the current irregularity from par, the total face defect, the number of odd-sided faces, par itself, the worst element quality, the mesh size relative to the window, and the face count, the first five divided by the number of faces so that each stays in its training range however large the mesh grows. The fraction of the move budget consumed is a separate scalar. After each move the centre stays on its face until that face is a quadrilateral, then moves to the most irregular face in view, or, when nothing irregular is in view, to the most irregular face or vertex in the whole mesh (Appendix B).

3.2 Actions

For a half-edge h with source u, three actions insert a *chord* from u to the vertex $j+2$ positions ahead of it in h's face loop, $j \in \{0,1,2\}$, cutting a sub-face of $j+3$ sides off the face that h bounds; a fourth splits h's edge with a new vertex (Figure 2b). A new vertex takes the generic desired degree $g(v)$ of Section 2, so par is unchanged by every move. Invalid actions are masked: no chord may leave a 2-gon, and no move may close a face of at most four sides at a boundary vertex that wants three or more edges (a corner of more than $135°$, including every flat boundary point) when both of its sides in that face lie on $\partial\Omega$: that element would have an angle of $180°$ or more there, and is degenerate whatever the smoother does. While our environment also supports deletions, we disable them to avoid free move-and-undo cycles. Between moves a few Laplace smoothing sweeps keep the observed angles meaningful; the edit operations themselves fix only connectivity.

3.3 Reward and termination

Let $F(s) = \sum_f |\,\mathrm{sides}(f) - 4\,|$ be the face defect of a state, $I(s)$ its irregularity equation 1, and $q(s)$ the minimum *shape quality* over the corners of the faces that are already quadrilaterals, where the shape quality of a corner with edge vectors a, b is $2\,(a \times b)/(|a|^2 + |b|^2)$: 1 for a square corner, falling with the aspect ratio of the two sides, 0 at a flat corner and negative when the element folds. The reward is shaped (Ng et al., 1999) by the potential

$$\Phi(s) \;=\; -\Big(F(s) \;+\; w_v \,\big|\, I(s) - \mathrm{par}(\Omega) \,\big| \;+\; w_q \,\max\big(0,\; q^\star - q(s)\big)\Big), \tag{3}$$

with $w_v = 1$, $w_q = 8$ and a quality target $q^\star = 0.4$. The per-move reward is the increase in potential divided by the initial defect $D_0 = \max(-\Phi(s_0), 1)$, less a charge per move, plus a *solve bonus* b:

$$r_t \;=\; \frac{\Phi(s_{t+1}) - \Phi(s_t)}{D_0} \;-\; \frac{c}{T_{\max}} \;+\; b\,\mathbb{1}[\,s_{t+1} \text{ solved}\,], \qquad c = 0.1,\; b = 1, \tag{4}$$

where $\mathbb{1}[\cdot]$ is 1 when its argument holds and 0 otherwise, so the solve bonus is paid once, on the move that reaches a solved state (defined below). Dividing by D_0 makes a return the *fraction* of the initial defect recovered, so large domains do not dominate the advantages, and keeps the exchange rate between quality and irregularity the same at every size. The quality term sees only finished quadrilaterals, so each poorly shaped element is charged at the move that creates it. The move budget scales with the start state, $T_{\max} = \max(12,\, \lceil 3\,|\mathcal{H}_0| \rceil)$ for $|\mathcal{H}_0|$ half-edges, and the per-move charge is scaled by it rather than applied as a discount: an episode that runs to its limit pays c whatever the domain, where a discount would penalise a large domain for legitimately needing more moves. With $\gamma = 1$ the return telescopes to $(\Phi(s_T) - \Phi(s_0))/D_0 - cT/T_{\max} + b\,\mathbb{1}[s_T \text{ solved}]$, on one scale across the distribution, which lets one critic fit every domain.

A state is *solved* when every face is a quadrilateral, $I(s) = \mathrm{par}(\Omega)$, and the mesh admits a non-degenerate embedding: it is passed once through an untangling smoother (Escobar et al., 2003; Freitag, 1997) that moves interior vertices and slides inserted boundary vertices along their edges, keeping the corners of Ω fixed (Appendix K), and accepted if $q \geq 0.1$ afterwards. The same test validates certified instances (Section 5), and the same smoother is applied to every method before scoring (Section 6). The three quality thresholds differ deliberately: $q^\star = 0.4$ gives the reward gradient across the range the agent's meshes occupy, 0.1 is a degeneracy test, and the usability bar of 0.3 used in evaluation is an engineering criterion.

4 A POLICY ON THE HALF-EDGE STRUCTURE

4.1 CONVOLUTION OVER `NEXT`, `PREVIOUS` AND `TWIN`

The observation is a set of half-edges together with the three pointers that relate them, and the network's message passing follows those pointers. A block reads, for every half-edge in the template, its own feature vector and those of its `next`, `previous` and `twin`, concatenates the four, and applies a shared linear map, layer normalisation and a LeakyReLU. A neighbour that falls outside the window and one that does not exist because the half-edge lies on $\partial\Omega$ are each replaced by their own learned vector, so the network can tell the edge of its view from the edge of the domain. Features are projected to a common width and passed through residual stages of two blocks each; every block is one hop. No parameter is indexed by position in the template or by mesh size, so a checkpoint applies unchanged to a larger window or a larger mesh, and a configuration presents the same input wherever in the mesh it occurs. An ablation in Appendix J compares this network with a Transformer given the same features but not the adjacency.

4.2 FROM HALF-EDGES TO A MASKED ACTION DISTRIBUTION

The global vector of Section 3.1 never enters message passing. The per-half-edge latents are pooled by mean and by maximum, concatenated with the global vector and the progress scalar, and mapped by a two-layer MLP to a *context* vector (Figure 4 in Appendix E). On the actor side each slot's latent passes through a shared layer, the context is added to every slot through a shared linear map, and a shared linear head emits four action logits per slot; flattened over the template and masked (Section 3.2), these form one categorical distribution over the action space. Because the context enters after the per-slot nonlinearity and the head is linear, it shifts every slot's logits by the same vector: it can re-weight the four action types but cannot favour one half-edge over another. *Where* to act rests on the convolution's local features and the centring rule; *what* to do there is what the context can shift. The critic reads only the context, through its own two-layer MLP, since the value of a state is a property of the whole mesh and not of the half-edge the window happens to be centred on; the normalised return of Section 3.3 is what lets one such critic fit every domain size. The network has 96 channels, four residual stages and about 3.7×10^5 parameters.

5 TRAINING UNDER SPARSE REWARD

Reaching par is a sparse event. Under the MDP of Section 3, uniform random play reaches par on 3% of polygons with five or six sides, 2% of those with up to eight, and on none with more than eight. An explorer that has never reached par has never seen the only signal that distinguishes optimal from merely finished, and PPO from a random initialisation reaches it on 7 of 96 domains after a million steps (ablation in Appendix J). We overcome this exploration wall via behaviour cloning on families of meshes which are optimal by construction.

5.1 CERTIFIED INSTANCES WITHOUT SEARCH

A polyomino, a connected set of unit lattice cells, is already an at-par all-quadrilateral mesh of its outline. Rather than search for a solution we start from one and undo it: a *backward walk* removes edges and dissolves vertices, choosing only removals that invert a legal forward move, until a single face remains, and the reversed record is a move-by-move solution in the agent's own action space. Four seed families supply variety (polyominoes, annuli, polar meshes about a hub, and *pinwheel* annuli, whose hole is rotated half a step against its rim, a shear in polar coordinates), and polyomino seeds are deformed by an affine map and per-corner jitter within its corners' angle bins, which leaves its solution valid; moving corners across a bin boundary yields instances whose bound is above zero (Appendix C).

5.2 CLONING, THEN REINFORCEMENT

Replaying a certified solution yields (observation, optimal move) pairs. Because independent moves commute, a state usually has a *set* of equally optimal moves; we record the set and minimise cross-entropy against a uniform distribution over it, together with value targets for the critic. The agent is cloned from $9{,}000$ certified instances (about $187{,}000$ pairs) for six epochs, reaching 93.7% top-one agreement with the certified sets, with model selection on a development set of fifteen hand-made levels of simple primitives (Appendix I).

PPO (Schulman et al., 2017) then continues from the cloned weights on a mixture of generated domains (Appendix K) and certified ones: 35% of episodes start from a freshly generated certified instance, so the policy keeps being asked to solve the problems cloning taught it and keeps meeting instances whose bound is not zero. Since the cloned value head was fit to returns on a different scale from PPO's, and an advantage measured against it would be dominated by that offset, we first train the value head on a frozen policy for forty epochs, and only after that run full PPO. Training runs with $\gamma = \lambda = 1$ and a target KL of 0.03 for four million steps (Appendix F). The network is small enough that the whole recipe, generating the certified instances, cloning and four million PPO steps, takes about four hours on a laptop (Apple M2, eight cores).

6 EXPERIMENTS

6.1 SETUP

Domains. Training and evaluation outlines come from GEO2D, a seeded generator of lattice-based mechanical parts: rectangular bases with cuts, additions and combs, optional 45° chamfers and interior holes. A defined preset and a seed determine a domain exactly (Appendix K). All evaluation domains are straight-sided and unseen in training. The *in-distribution* set is 96 domains ranging in size from 8 to 24 boundary corners drawn from four families: rectilinear or chamfered, with or without holes. The *larger-boundary* set is 64 domains from the same four families with 25 to 50 corners, beyond anything seen in training. Every generated outline has par zero, but the faces the agent works through on the way do not: each intermediate polygon is a sub-problem with its own bound, non-zero for 39% of them in distribution and 73% on the larger set (Appendix D).

What is measured. Every method is judged on the mesh as delivered, after one pass of the same untangling smoother. We report, in this order: whether the mesh is *all-quadrilateral*; whether it is *usable* (shape quality $q \geq 0.3$); the *excess over par*, $I(\mathcal{M}) - \mathrm{par}(\Omega)$ (tables give the median over

all-quadrilateral meshes); and whether it is *at par*, $I(\mathcal{M}) = \mathrm{par}(\Omega)$ with $q \geq 0.1$, the certificate of Section 2.

Evaluation-time procedure. The same procedure is applied to every domain. Each is attempted five times — one greedy rollout and four sampled — with the move budget of Section 3.3 doubled to $\lceil 6|\mathcal{H}_0| \rceil$; every all-quadrilateral state along every attempt is untangled and scored, and the best is kept, ranked by all-quadrilateral first, then minimum quality, then closeness to par. If that mesh is below the quality bar, a *split-and-continue repair* locates its worst corner, makes one of the environment's own moves there — a vertex on the longer side of a thin corner, a chord across a badly angled one — and hands the state back to the policy to finish, for up to three rounds. Every count reported is the mean of this procedure over six rollout seeds on the in-distribution set and four on the larger set. Across seeds, all-quadrilateral does not move and usable moves by about one domain; at par moves by 1.3 of 96 and 4.1 of 64 (Appendix H).

Baselines. Seven quadrilateral meshing configurations of Gmsh (Geuzaine & Remacle, 2009; Reberol et al., 2021) are run on the same domains and scored by the same criteria after the same smoother. Each is run at the element count the agent used on that domain, so that no method gains by refining its way to a better score; those that cannot produce a mesh that coarse are reported at their natural sizes. The elements column is the mean over a method's all-quadrilateral meshes, and $\times$ours is that mean as a multiple of ours. We also report a *head-to-head* win rate: on each domain the agent wins if it produces an all-quadrilateral mesh and the baseline does not, loses in the reverse case, and otherwise wins on *quality* if its minimum quality is higher and on *regularity* if its excess over par is lower (equal values tie). This avoids comparing medians over the different subsets of domains each method completes.

6.2 Main result

The top block of Table 1 gives the in-distribution result. The agent produces an all-quadrilateral mesh on every one of the 96 domains, a usable one on 95.7, and reaches the certified optimum on 90.2; the median excess over par is zero on every family. The Gmsh configurations that can be asked for the agent's element count complete about half the domains, and those that complete all of them need three to fourteen times the elements to do so, and still carry a median excess of 7.5 and 26. Across all seven configurations (Table 6) the certified optimum is reached three times in 672 attempts: a quality-driven mesher has no notion of the bound. Head to head the agent wins on 90% of decided domains against BLOSSOM and 90% against FRONTAL-QUAD at matched count, 67% against QUASI-STRUCTURED at three times the elements, and 37% against BLOSSOM-FULL at fourteen times, the one case in which more elements buy Gmsh the better minimum quality more often than not. On regularity it wins on every decided domain against all four, at any of their element counts.

6.3 Larger boundaries

The bottom block of Table 1 reports the 64 domains with 25 to 50 corners, twice anything seen in training, under the same agent and procedure. The agent completes every domain, clears the quality bar on 62.0, keeps a median excess over par below one against Gmsh's 39 at matched count, and reaches the certified optimum on 31.0, where no Gmsh configuration does. Head to head at matched count it wins on quality on 83% and 87% of decided domains against BLOSSOM and FRONTAL-QUAD, and on regularity on all of them, as against every configuration at any element count.

The evaluation-time procedure changes almost nothing in distribution; on the larger set it matters, and the last rows of Table 1 remove its components one at a time. With the default move budget a few domains end at the move cap with faces still open; doubling it completes all 64 and lifts the usable count from 41.0 to 54.5 at about a quarter more elements, and tripling buys nothing further. The repair lifts it to 62.0; one round usually suffices. Seventeen attempts instead of five raise the usable count to 63.5. While exact optimality drops on larger domains, with about half at par, the median excess over par stays below one, against 22 for the most regular Gmsh configuration.

Table 1: Top: the 96 in-distribution domains, mean over six evaluation seeds (per family: Table 7). Bottom: the 64 larger-boundary domains, four evaluation seeds; its last rows remove parts of the evaluation-time procedure (cumulatively) and add attempts. Columns and win rates as defined in Section 6.1; each Gmsh configuration is asked for our element count and returns the closest it can reach. Bold: best in column within a block (elements: among methods complete on every domain).

	all-quad	usable	excess	at par	elements	×ours	our win rate quality	our win rate regularity
96 in-distribution domains (8–24 corners)								
Gmsh blossom	51	38	9	0	19	1.2	90%	100%
Gmsh frontal-quad	43	29	8	1	16	1.0	90%	100%
Gmsh quasi-structured	**96**	**96**	7.5	2	51	3.2	67%	100%
Gmsh blossom-full	**96**	**96**	26	0	220	14.0	37%	100%
Ours	**96.0**	95.7	**0**	**90.2**	**16**	1.0		
64 larger-boundary domains (25–50 corners)								
Gmsh blossom	26	18	39	0	97	1.2	83%	100%
Gmsh frontal-quad	15	14	34	0	96	1.2	87%	100%
Gmsh quasi-structured	**64**	60	22	0	232	3.0	23%	100%
Gmsh blossom-full	**64**	**64**	156	0	1180	15.2	2%	100%
Ours	**64.0**	62.0	**0.75**	**31.0**	**78**	1.0		
no repair	64.0	54.5	0.5	32.8	74			
and default move budget	58.2	41.0	0	34.8	59			
17 attempts	64.0	63.5	1.75	26.0				

Table 2: Curved boundaries: the released agent without retraining, one evaluation pass on 96 domains with 8–24 corners. Left: columns that do not depend on how quality is measured, as in Table 1 (at par with $q \geq 0.1$ on the tangents; on chords QUASI-STRUCTURED reaches par on 2). Right: usable count and our win rate on quality, with element quality judged against the arc tangents or chords. Ours: with the repair search; the row below it is the five attempts and doubled budget of Section 6.1 without any repair. Bold: best in column (elements: among methods complete on every domain).

	all-quad	excess	at par	elem.	×ours	reg. win	usable tangent	usable chord	quality win tangent	quality win chord
Gmsh blossom	62	20	0	60	1.1	100%	7	54	94%	54%
Gmsh frontal-quad	55	22	0	58	1.1	100%	2	42	100%	78%
Gmsh quasi-structured	**96**	10	0	83	1.5	100%	6	64	95%	67%
Gmsh blossom-full	93	32	0	235	4.4	100%	25	79	74%	47%
Ours	**96**	**2**	**37**	**54**	1.0		**95**	**88**		
no repair search	96	2	19	52			74	80		

6.4 CURVED BOUNDARIES

We run the released agent, unchanged and never trained on a curved boundary, on 96 domains of 8–24 corners with fillets and semicircular notches, with and without holes. Elements along arcs often fall below the quality bar, so we add a test-time repair search. Full details are in Appendix L. Element quality along an arc can be judged against the arc's tangent or the chord connecting two vertices. For block decomposition, the tangent matters, since refinement will place vertices on the curved surface. However, we compute and report both metrics for the sake of comparison since it significantly affects the quality reading of Gmsh.

The agent delivers median excess over par of 2 and reaches par on 37 of 96 domains, while every Gmsh configuration carries an excess of 10 to 32 and reaches par on at most 2; the agent wins on regularity on every decided domain against all four (Table 2). It is usable on 95 of 96 against the tangents and 88 on chords. On chords Gmsh clears the bar far more often, on up to 79 with BLOSSOM-FULL at $4.4\times$ the elements, but at a matched element count the agent remains ahead of BLOSSOM and FRONTAL-QUAD. On 64 larger curved domains of 25–50 corners (Appendix L) the

regularity result holds, a median excess of 8 against 28 to 101, while the agent clears the quality bar on fewer domains than Gmsh configurations that use several times its elements.

7 Limitations and related work

7.1 Limitations

- *The smoother is in the loop.* The solved test untangles the mesh before judging it, and a policy trained against that test relies on it; we score every method after the same smoother, but an agent that drew a well-shaped mesh without one would be preferable.
- *Runtime.* Gmsh meshes a domain in milliseconds; our unoptimised agent takes under a second per in-distribution domain on a laptop, seconds per larger domain, and minutes on the largest holed ones.
- *Curved boundaries need a search.* The agent meshes curved domains close to the bound without retraining, but clearing the quality bar along arcs takes a test-time repair search of up to ten minutes per domain (Section 6.4); a policy that draws well-shaped elements along arcs without it remains open.

7.2 Related work

Learning to mesh. Learned quadrilateral meshing has so far learned *where to place the next element*: Pan et al. (2023) train a soft actor–critic to make sequential meshing decisions, Tong et al. (2023) steer an advancing front with supervised and reinforcement learning, and, concurrently, Kalyan et al. (2026) pair geometric decomposition with all-quadrilateral generation in a multi-agent framework. Our actions are instead edits on a general polygonal half-edge mesh, and an episode ends at a certificate of optimality rather than when a front is consumed. Closest in objective is Narayanan et al. (2024), whose moves take one all-quadrilateral mesh to another toward ideal vertex degrees (contrasted in Section 1). Other learned mesh work optimises geometry or error rather than connectivity (Lorsung & Barati Farimani, 2023; Yang et al., 2023; Foucart et al., 2023; Thacher et al., 2025).

Networks on mesh connectivity; learning backward from solutions. Edge and half-edge convolutions (Hanocka et al., 2019; Ludwig et al., 2023) analyse fixed triangle meshes; ours drives a policy on a polygonal mesh that changes every move, with action heads on the pointers it convolves over. Learning backward from a solved state is established when there is one goal (McAleer et al., 2018; Agostinelli et al., 2019; Florensa et al., 2017; Resnick et al., 2018); here each domain has its own optimum, so certified optima are constructed and walked back to their boundary.

Classical construction and the bound. Paving (Blacker & Stephenson, 1991), Q-Morph (Owen et al., 1999), medial-axis decomposition (Tam & Armstrong, 1991; Fogg et al., 2016; Sun et al., 2021) and the production meshers descended from them (Geuzaine & Remacle, 2009; Reberol et al., 2021) decide locally against quality heuristics; field-aligned methods (Bommes et al., 2009; Kälberer et al., 2007; Jakob et al., 2015; Ebke et al., 2013) optimise alignment to a cross field. Neither targets a topological bound. The bound itself is due to Peng & Wonka (2013) and Peng et al. (2014); we build on their work by making the bound the termination criterion of an MDP, reached by a learned policy.

8 Conclusion

Quadrilateral block decomposition is an unusual reinforcement learning problem: the space of connectivities grows combinatorially with the boundary and the reward is sparse, yet every success carries a certificate of optimality. Exploration almost never reaches that certificate, which stalls PPO from a random start; cloning on optima that are trivial to construct gets past it, and PPO on generated domains improves on what cloning taught. The resulting decompositions are complete, usable and provably as regular as the domain permits, where production meshers at the same coarseness essentially never reach the bound, and remain nearly regular at twice the training scale and on curved boundaries the agent never saw in training.

AI USE STATEMENT

We used generative AI tools, primarily large language model coding and writing assistants, throughout this project, and disclose their use by task following the ICLR 2027 policy.

Tasks with required disclosure where AI was used.

- *Implementing methods:* AI was used extensively to write and refactor code throughout the codebase.
- *Proposing hypotheses:* AI proposed behaviour cloning and helped design the families of constructible meshes with certified solutions.
- *Mathematical claims and proofs:* the optimality bound is due to prior work (Peng & Wonka, 2013; Peng et al., 2014); AI assisted with writing its proof in the form used here.
- *Research methodology and experiments:* the experimental design was proposed by the authors. AI ran and monitored the experiments, explored hyperparameter settings, including the relative weights of the reward terms, and suggested follow-up experiments. The authors analyzed results, diagnosed failure modes (often requiring human intuition), and devised mitigations that overcame them.
- *Cleaning and reformatting data:* AI was used in result collection and consolidation.

Tasks with required disclosure where AI was not used or which do not apply.

- Interpreting results: the authors examined the metrics and failure modes and drew the conclusions.
- Translation, qualitative or thematic data analysis, and surveys or interviews: not applicable.

Tasks with recommended disclosure where AI was used.

- Drafting and revising the text of the paper and figures over several rounds of revision.
- Summarising and searching the literature, and identifying relevant work.

What the authors did without AI.

- The MDP formulation: the environment, the action space, the input features, and the state and template construction.
- The reward formulation, including its potential-based shaping, and the termination condition and acceptance criteria.
- The convolution over the half-edge data structure and the action distribution.
- The experimental design, the interpretation of results, and the design of GEO2D.

Verification. We reviewed all AI-assisted work. Every reported number was checked against the per-domain records it was computed from, bibliographic entries were checked against the publishers' records, and AI-written code was tested and its outputs spot-checked by re-running evaluations. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The domain generator is seeded, so every evaluation domain in Section 6 is reproduced by a preset and a seed; the certified-instance generator, the environment, the network, trained checkpoints and configurations, and the scoring scripts will be made public, and Appendix F lists every hyperparameter. Scoring is deterministic given the domain seed and the rollout seed, so any table can be regenerated exactly, and the per-domain records behind every table will be released with two scripts that rebuild them (Appendix H).

ETHICS STATEMENT

This work concerns geometric mesh generation for engineering analysis and raises no ethical concerns beyond those of general-purpose numerical software.

REFERENCES

Forest Agostinelli, Stephen McAleer, Alexander Shmakov, and Pierre Baldi. Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1(8):356–363, 2019. doi: 10.1038/s42256-019-0070-z.

Cecil G. Armstrong, Harold J. Fogg, Christopher M. Tierney, and Trevor T. Robinson. Common themes in multi-block structured quad/hex mesh generation. *Procedia Engineering*, 124:70–82, 2015. doi: 10.1016/j.proeng.2015.10.123.

Ted D. Blacker and Michael B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 1991. doi: 10.1002/nme.1620320410.

David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Transactions on Graphics*, 2009. doi: 10.1145/1531326.1531383.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Computer Graphics Forum*, 2013. doi: 10.1111/cgf.12014.

Hans-Christian Ebke, David Bommes, Marcel Campen, and Leif Kobbelt. QEx: Robust quad mesh extraction. *ACM Transactions on Graphics*, 2013. doi: 10.1145/2508363.2508372.

J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, and J. M. González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2775–2787, 2003. doi: 10.1016/S0045-7825(03)00299-8.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning (CoRL)*, volume 78 of *Proceedings of Machine Learning Research*, pp. 482–495, 2017.

Harold J. Fogg, Cecil G. Armstrong, and Trevor T. Robinson. Enhanced medial-axis-based block-structured meshing in 2-D. *Computer-Aided Design*, 2016. doi: 10.1016/j.cad.2015.07.001.

Harold J. Fogg, Liang Sun, Jonathan E. Makem, Cecil G. Armstrong, and Trevor T. Robinson. Singularities in structured meshes and cross-fields. *Computer-Aided Design*, 105:11–25, 2018. doi: 10.1016/j.cad.2018.06.002.

Corbin Foucart, Aaron Charous, and Pierre F. J. Lermusiaux. Deep reinforcement learning for adaptive mesh refinement. *Journal of Computational Physics*, 491:112381, 2023. doi: 10.1016/j.jcp.2023.112381.

Lori A. Freitag. On combining Laplacian and optimization-based mesh smoothing techniques. In *Trends in Unstructured Mesh Generation*, volume 220 of *AMD*, pp. 37–43. ASME, 1997.

Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi: 10.1002/nme.2579.

Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: A network with an edge. *ACM Transactions on Graphics*, 38(4):1–12, 2019. doi: 10.1145/3306346.3322959.

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. Instant field-aligned meshes. *ACM Transactions on Graphics*, 2015. doi: 10.1145/2816795.2818078.

Felix Kälberer, Matthias Nieser, and Konrad Polthier. QuadCover – surface parameterization using branched coverings. *Computer Graphics Forum*, 2007. doi: 10.1111/j.1467-8659.2007.01060.x.

Anirudh Kalyan, Cosmin Anitescu, Xiaoying Zhuang, Timon Rabczuk, Somdatta Goswami, and Sundararajan Natarajan. Dmsh: A multi-agent reinforcement learning framework for all-quad mesh generation. *arXiv preprint arXiv:2606.10601*, 2026.

Paul Kinney. CleanUp: Improving quadrilateral finite element meshes. In *Proceedings of the 6th International Meshing Roundtable*, Park City, UT, 1997.

Cooper Lorsung and Amir Barati Farimani. Mesh deep Q network: A deep reinforcement learning framework for improving meshes in computational fluid dynamics. *AIP Advances*, 2023. doi: 10.1063/5.0138039.

Ingmar Ludwig, Daniel Tyson, and Marcel Campen. HalfedgeCNN for native and flexible deep learning on triangle meshes. *Computer Graphics Forum*, 42(5), 2023. doi: 10.1111/cgf.14898.

Stephen McAleer, Forest Agostinelli, Alexander Shmakov, and Pierre Baldi. Solving the Rubik's cube without human knowledge. *arXiv preprint arXiv:1805.07470*, 2018.

Arjun Narayanan, Yulong Pan, and Per-Olof Persson. Learning topological operations on meshes with application to block decomposition of polygons. *Computer-Aided Design*, 175:103744, 2024. doi: 10.1016/j.cad.2024.103744.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, 1999.

S. J. Owen, M. L. Staten, S. A. Canann, and S. Saigal. Q-Morph: an indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering*, 1999. doi: 10.1002/(sici)1097-0207(19990330)44:9⟨1317::aid-nme532⟩3.0.co;2-n.

Jie Pan, Jingwei Huang, Gengdong Cheng, and Yong Zeng. Reinforcement learning for automatic quadrilateral mesh generation: A soft actor–critic approach. *Neural Networks*, 2023. doi: 10.1016/j.neunet.2022.10.022.

Chi-Han Peng and Peter Wonka. Connectivity editing for quad-dominant meshes. *Computer Graphics Forum*, 2013. doi: 10.1111/cgf.12171.

Chi-Han Peng, Michael Bartoň, Caigui Jiang, and Peter Wonka. Exploring quadrangulations. *ACM Transactions on Graphics*, 2014. doi: 10.1145/2541533.

Maxence Reberol, Christos Georgiadis, and Jean-François Remacle. Quasi-structured quadrilateral meshing in Gmsh – a robust pipeline for complex CAD models. *arXiv preprint arXiv:2103.04652*, 2021.

Cinjon Resnick, Roberta Raileanu, Sanyam Kapoor, Alexander Peysakhovich, Kyunghyun Cho, and Joan Bruna. Backplay: "man muss immer umkehren". *arXiv preprint arXiv:1807.06919*, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Liang Sun, Cecil G. Armstrong, Trevor T. Robinson, and Dimitrios Papadimitrakis. Quadrilateral multiblock decomposition via auxiliary subdivision. *Journal of Computational Design and Engineering*, 2021. doi: 10.1093/jcde/qwab020.

T. K. H. Tam and C. G. Armstrong. 2d finite element mesh generation by medial axis subdivision. *Advances in Engineering Software and Workstations*, 1991. doi: 10.1016/0961-3552(91)90035-3.

Will Thacher, Yulong Pan, and Per-Olof Persson. Optimization of a triangular Delaunay mesh generator using reinforcement learning. *Computer-Aided Design*, 2025. doi: 10.1016/j.cad.2025.103964.

Hua Tong, Kuanren Qian, Eni Halilaj, and Yongjie Jessica Zhang. SRL-assisted AFM: Generating planar unstructured quadrilateral meshes with supervised and reinforcement learning-assisted advancing front method. *Journal of Computational Science*, 2023. doi: 10.1016/j.jocs.2023.102109.

Jiachen Yang, Ketan Mittal, Tarik Dzanic, Socratis Petrides, Brendan Keith, Brenden Petersen, Daniel Faissol, and Robert Anderson. Multi-agent reinforcement learning for adaptive mesh refinement. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 14–22, 2023.

A THE PAR CONVENTION IN FULL

Lemma 2 (Discrete Gauss–Bonnet (Peng & Wonka, 2013; Peng et al., 2014)). *For any conforming all-quadrilateral mesh $\mathcal{M}$ of Ω, $\sum_{v\in V}\big(g(v)-\deg(v)\big)=4\chi$.*

Proof of Lemma 2. Counting incidences between faces and edges, each of the $|F|$ faces contributes four while each interior edge lies on two faces and each boundary edge on one, so $4|F| = 2(|E| - |B|) + |B| = 2|E| - |B|$. The boundary is a disjoint union of cycles and so carries $|B|$ vertices; summing degrees counts each edge twice. Hence $\sum_v \deg(v) = 2|E|$ and $\sum_v g(v) = 4(|V|-|B|)+3|B| = 4|V|-|B|$, and substituting $|B| = 2|E|-4|F|$ into their difference gives $4|V|-|B|-2|E| = 4(|V| - |E| + |F|) = 4\chi$. □

Proof of Theorem 1. A vertex of $\mathcal{M}$ that is not a corner of Ω is either interior, where $d(v) = g(v) = 4$, or lies in the interior of a boundary arc, where tangent continuity gives $\theta_v = \pi$ and so $d(v) = g(v) = 3$. Only corners contribute, so $\sum_{v\in V}(d(v) - g(v)) = c(\Omega)$ for every such mesh, and with Lemma 2, $\sum_v(d(v) - \deg(v)) = 4\chi + c(\Omega)$. The claim is then the triangle inequality, $\sum_v |d(v) - \deg(v)| \geq |\sum_v(d(v) - \deg(v))|$. □

Ties. When $\theta/(\pi/2)$ is exactly a half-integer $k+\frac{1}{2}$ — a 135° or 225° corner under the quadrilateral target, and every rectilinear corner under a triangular one — two treatments are equally admissible: k elements of angle θ/k or $k+1$ of angle $\theta/(k+1)$ sit the same distance either side of the target. Rounding picks one by convention, and the convention then appears in both the irregularity and the bound. We therefore let such a corner carry the pair $\{k+1, k+2\}$: its irregularity is the distance to the nearer member, which keeps $I(\mathcal{M})$ separable over vertices, and par is the smallest value of $|c + 4\chi|$ over all assignments of one member to each tie corner. Because the two options differ by exactly one, that minimum is found by a scan over how many tie corners are taken high rather than an enumeration, and it remains a bound: the assignment that minimises a given mesh's irregularity has its own $|c + 4\chi|$ below that irregularity by Theorem 1.

B THE CENTRING RULE

Write the *irregularity* of a face as $|\,\text{sides} - 4\,|$ and the *defect* of a vertex as $|\deg(v) - d(v)|$, to the nearer option at a tie corner. In the initial state the centre is the half-edge whose source vertex has the largest defect on the most irregular face. After each move: if the centre's own face is not yet a quadrilateral, the centre stays where it is and only the template around it is rebuilt. Otherwise, if any face meeting the current window is still irregular, the centre moves to one of maximal irregularity among those and, within it, to the half-edge whose source vertex has the largest defect. Only when nothing irregular remains in view does the rule consult the whole mesh, moving to a face of maximal irregularity and its worst vertex; if every face is quadrilateral, the centre moves to the worst vertex anywhere.

C CERTIFIED INSTANCES

A polyomino is an at-par mesh of its outline because every interior vertex has degree four and every boundary vertex the degree its 90°, 180° or 270° corner asks for. The four seed families contribute different structure. Polyominoes give rectilinear domains at par zero. Annuli give domains with a hole; because the backward walk leaves the connecting slit in place, they arrive in the same single-face representation the agent meets at test time. Polar meshes of m quadrilaterals about a single hub give par $|4 - m|$.

Polyomino seeds are then deformed by an affine map plus per-corner jitter, taken as far as the angle bins allow; seeds with a hole and polar meshes are only mirrored (and rotated and scaled, which the normalised features do not see), so their variety comes from the construction itself. Holes in annuli are rotated relative to the outer boundary to create pinwheels. A corner's desired degree depends only on which bin its interior angle falls in, so the known solution stays valid on the skewed, irregular polygon. Redrawing an outline from prescribed corner angles is how instances with par above zero are produced: par is the total rounding error of the corner angles (Section 2), so moving k corners

consistently across a bin boundary yields an instance of par k. About a third of the instances are of this kind. Every candidate is replayed through the environment and checked against the solved test of Section 3.3 before it enters the dataset. Figure 3 shows one instance of each family.

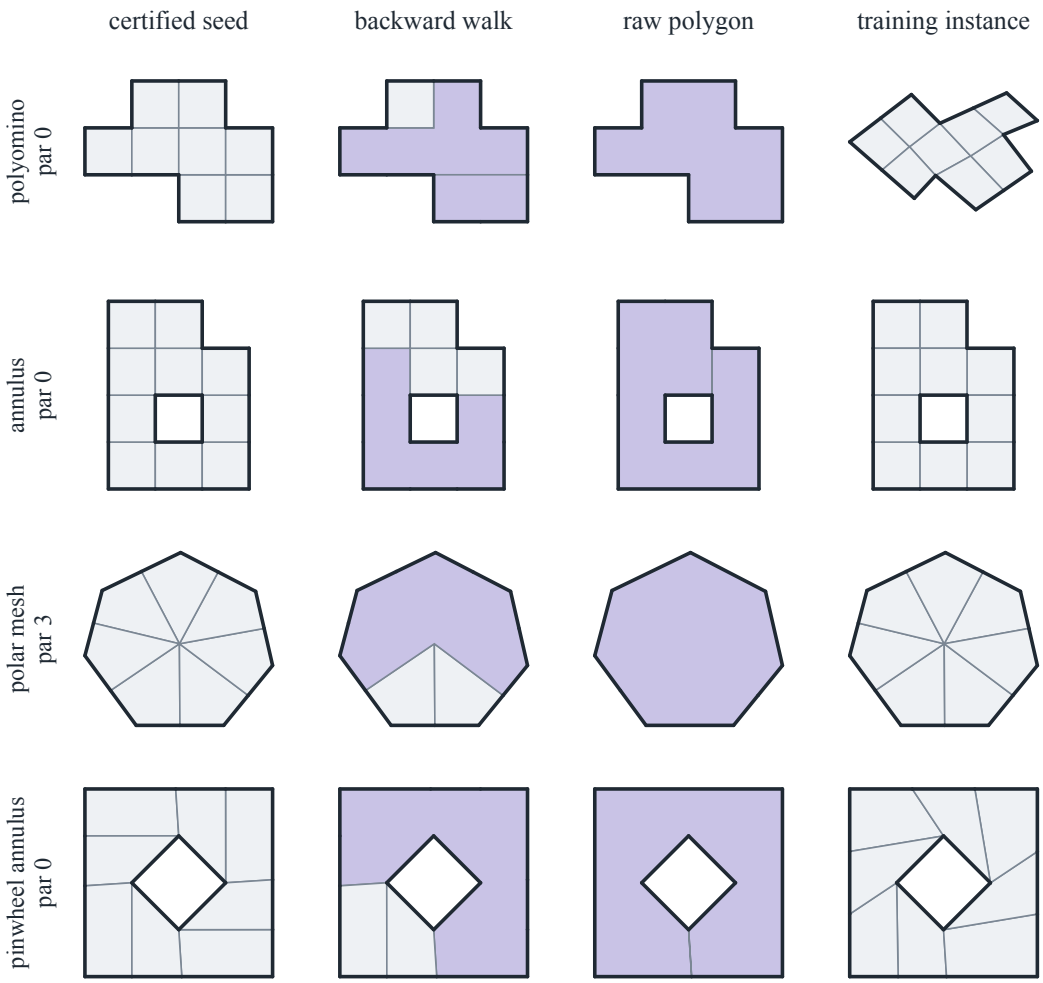


Figure 3: Certified instances, one per family, each the first accepted at a fixed seed. Columns: the constructed optimal mesh; the backward walk a third of the way through; the single face it ends on, which is the start state; and the training instance, the transformed outline with the recorded moves replayed on it and accepted as solved. The last three rows are drawn with rotation and scale undone. Holed outlines keep one edge joining the hole to the outer boundary, so that the start state is a single face. Shaded: faces that are not quadrilaterals.

D PAR-ZERO DOMAINS ARE NOT THE EASY CASE

Every evaluation domain has par zero (Section 6.1), which might suggest the benchmark never tests the bound on non-zero cases. However, in the course of solving a par zero geometry, the agent routinely encounterss sub-problems with non-zero par. Each chord splits a face in two, and every face that is not yet a quadrilateral is a domain of its own: a vertex on its loop with $\deg(v)$ edges, two of them the face's sides, wants $d(v) - \deg(v) + 2$ edges within it, and Theorem 1 applied to the face gives the face's bound. Table 3 counts the distinct unfinished faces the released agent creates in its greedy attempt on every evaluation domain; faces that still contain a hole's slit, whose corners the count cannot separate, are left out. In distribution, 39% of sub-problems have a nonzero bound and 75 of 96 episodes meet one; on the larger set, 73% do and every episode meets one.

Table 3: Bounds of the sub-problems met in the released agent's greedy attempt on the par-zero evaluation domains; the last column counts episodes that meet at least one sub-problem with par above zero.

set	sub-problems	par 0	1	2	3	≥ 4	par > 0	episodes
96 in-distribution	2,195	60.7%	33.8%	4.6%	0.8%	0.1%	39.3%	75 of 96
64 larger	11,717	26.7%	59.0%	10.9%	3.0%	0.4%	73.3%	64 of 64

The same holds for whole domains. On 48 domains with par 1 to 4, drawn from the certified generator at a held-out seed, the agent reaches par on 47. No Gmsh configuration at the matched element count reaches par on more than 8, and at par 3 or more only PACKING reaches it at all, on 4 of 20 (Table 4).

Table 4: 48 held-out domains with par above zero: domains at par, by par. Ours: best of five attempts, ranked by completion and then closeness to par, at the default move budget. Gmsh: at the agent's element count where it can reach it.

	par 1 (11)	2 (17)	3 (17)	4 (3)	total	all-quad
Ours	**11**	**16**	**17**	**3**	**47**	**48**
Gmsh quasi-structured	2	6	0	0	8	**48**
Gmsh packing	0	3	4	0	7	8
Gmsh frontal-quad	1	0	0	0	1	27
Gmsh blossom	0	0	0	0	0	30
Gmsh blossom-full	0	0	0	0	0	9
Gmsh simple-recombine	0	0	0	0	0	2
Gmsh subdivide	0	0	0	0	0	**48**

E THE POLICY NETWORK

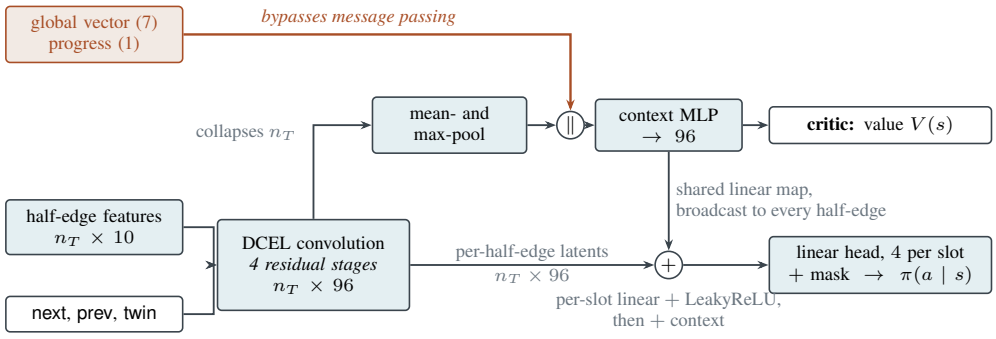


Figure 4: The policy. The trunk is per half-edge: convolved features pass through a shared per-slot linear layer to the action head. Pooling is a *branch* — it collapses the half-edge axis, and the global vector and progress scalar, which never enter message passing, join there. The context vector it produces is one vector per state: the critic's only input, and, on the actor side, an identical shift added to every half-edge before a shared linear head. It can therefore re-weight the four action types but cannot by itself favour one half-edge over another — selecting *where* to act rests on the convolution and on the rule that decides which half-edges are in the window at all.

F HYPERPARAMETERS

Table 5: The released configuration.

observation window n_T	64 half-edges
edge-length feature	divided by the window's median
global scalars	per face, seven entries
actions per half-edge	4: chords to $+2$, $+3$, $+4$; vertex insertion
default move budget $T_{\max}$	$\max(12, \lceil 3\lvert\mathcal{H}_0\rvert\rceil)$
smoothing per move	5 Laplacian sweeps, corners pinned
potential weights w_v, w_q, $q^\star$	1, 8, 0.4
quality metric	shape, $2(a \times b)/(\lvert a\rvert^2 + \lvert b\rvert^2)$
step charge c, solve bonus b	0.1, 1.0
solved test	topologically at par and $q \geq 0.1$ after 4 untangling iterations
network	$10 \to 96$ projection, 4 residual stages of 2 DCEL blocks, LayerNorm, LeakyReLU
context	mean- and max-pool (192) + global (7) + progress (1) $\to$ MLP 96
parameters	3.7×10^5
cloning	9,000 instances, 187k pairs, 6 epochs, uniform target over the optimal set
certified anchor in PPO	35% of episodes, fresh batches of 256
PPO	$\gamma = \lambda = 1$, clip 0.2, target KL 0.03, entropy $5 \cdot 10^{-3}$, value coef. 0.5 512 steps per worker, 6–8 workers, minibatch 256, 4 epochs, lr 10^{-4} linear decay
critic warm-up	40 epochs, policy frozen
budget	4M PPO steps from the cloned weights
training time	about 4 h on a laptop (Apple M2, 8 cores): instances and cloning ≈ 50 min, PPO ≈ 3 h at ≈ 355 steps/s
evaluation: attempts	5 (1 greedy + 4 sampled), best kept
evaluation: move budget	doubled, $\max(12, \lceil 6\lvert\mathcal{H}_0\rvert\rceil)$
evaluation: repair	up to 3 rounds while $q < 0.3$: insert on the long side if aspect ≥ 2, else chord across the corner; then the policy finishes

G ALL SEVEN GMSH CONFIGURATIONS

Table 6: Seven Gmsh configurations on the 96 in-distribution domains, each asked for the agent's element count where it can produce one; the elements column gives the mean over its all-quadrilateral meshes and its multiple of our 15.7. On packing and simple-recombine nearly every win is decided by completion. Bold: best in column; for elements, among methods complete on every domain (packing and simple-recombine average over 8 and 2 meshes). Our win rate: fraction of decided domains on which our mesh is better on quality and on regularity, under the rule of Section 6.1.

							our win rate	
	all-quad	usable	excess	at par	elements	$\times$ours	quality	regularity
Gmsh blossom	51	38	9	0	19	1.2	90%	100%
Gmsh frontal-quad	43	29	8	1	16	1.0	90%	100%
Gmsh packing	8	6	5	0	15	0.9	98%	100%
Gmsh simple-recombine	2	1	11	0	14	0.9	100%	100%
Gmsh subdivide	**96**	43	49	0	85	5.4	97%	100%
Gmsh quasi-structured	**96**	**96**	7.5	2	51	3.2	67%	100%
Gmsh blossom-full	**96**	**96**	26	0	220	14.0	37%	100%
Ours	**96.0**	95.7	**0**	**90.2**	**16**	1.0		

Figure 5 shows the same comparison domain by domain, for the two matched-count configurations and QUASI-STRUCTURED, on both sets. In the quality panels a mesh that is not all-quadrilateral is counted in its own bar, so every method is shown on every domain; the per-mesh minimum quality is the quantity the usability bar is applied to. The excess over par is only defined on an all-quadrilateral mesh, so its panels show each method's all-quadrilateral meshes, as a share of them.

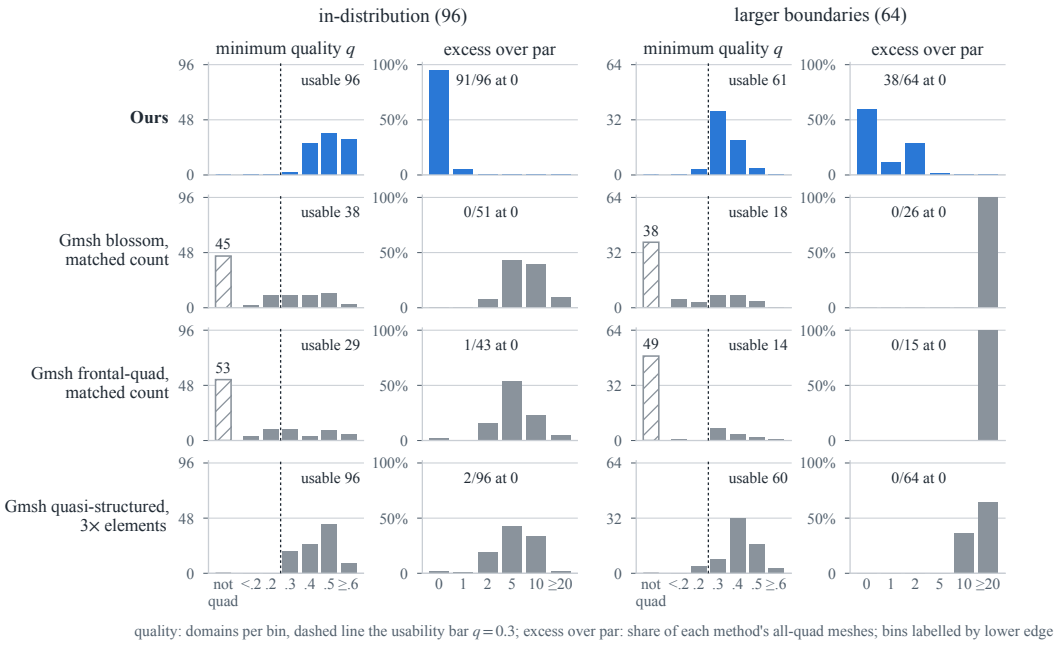


Figure 5: Per-mesh minimum quality and excess over par, one mesh per domain: the agent in one evaluation pass (rollout seed 0) and Gmsh asked for that pass's element count on each domain. Quality panels: domains per bin, hatched bar not all-quadrilateral. Excess panels: share of the method's all-quadrilateral meshes, with the number at zero quoted. The counts are this pass's; the tables report the mean over evaluation seeds.

Table 7: The in-distribution block of Table 1 by family: best of five attempts under the procedure, mean over six evaluation seeds. Columns in the order a mesher is judged: all-quadrilateral, usable ($q \geq 0.3$), median excess over par, and at par (the certified optimum). Gmsh BLOSSOM is given the agent's element count on each domain; QUASI-STRUCTURED cannot go that coarse and runs at its natural size, 3.2× the agent's. Bold: best in column over all 96 domains.

family	method	all-quad	usable	excess	at par	elements
chamfered	Ours	24.0	24.0	0	20.7	14
	Gmsh blossom	14	12	8.5	0	19
	Gmsh quasi-structured	24	24	6.5	0	54
rectilinear	Ours	24.0	24.0	0	23.8	10
	Gmsh blossom	16	10	8	0	15
	Gmsh quasi-structured	24	24	5	1	38
chamfered, holes	Ours	24.0	23.7	0	21.7	18
	Gmsh blossom	14	11	9	0	19
	Gmsh quasi-structured	24	24	9	1	48
rectilinear, holes	Ours	24.0	24.0	0	24.0	21
	Gmsh blossom	7	5	18	0	24
	Gmsh quasi-structured	24	24	12	0	64
all 96	Ours	**96.0**	95.7	**0**	**90.2**	**16**
	Gmsh blossom	51	38	9	0	19
	Gmsh quasi-structured	**96**	**96**	7.5	2	51

H PER-DOMAIN RESULTS AND EVALUATION NOISE

Every per-domain record behind Tables 1 and 9 — draw index, corner count, par, face defect, excess over par, minimum shape quality, element and vertex counts, and the Gmsh rows paired by draw

— will be released with the code as a single archive, together with the two scripts that regenerate every table and every head-to-head rate from it. Table 8 gives the released agent's spread across evaluation seeds: six on the in-distribution set and four on the larger one. The domain seed is fixed, so these deviations measure rollout sampling alone, and a comparison between two checkpoints on one evaluation seed is paired.

Table 8: The released agent under the procedure of Section 6.1: mean, standard deviation across evaluation seeds, and range.

set	column	mean	sd	range
96 in-distribution, 6 seeds	all-quad	96.0	0.0	96–96
	usable	95.7	0.5	95–96
	excess	0	0.0	0–0
	at par	90.2	1.3	88–92
64 larger, 4 seeds	all-quad	64.0	0.0	64–64
	usable	62.0	1.0	61–63
	excess	0.75	0.43	0–1
	at par	31.0	4.1	28–38

I THE DEVELOPMENT SET

Model selection during cloning used fifteen hand-made levels — L, T, U and I brackets, a staircase, a plus, a triangle and a pentagon, a star, plates with square and triangular holes, a gear, and three with curved sides — each with a known expert solution. They are small (three to ten corners), five of them are shapes the certified generator can itself produce, and they were never used for any number in Section 6. The released agent solves all fifteen greedily and twelve of the fourteen with a known optimum in the expert's move count; we report this as a development score only.

J ABLATIONS

Cloning is necessary for PPO to learn. At one million PPO steps and the same seed, the recipe of Section 5 is at 91.8 usable and 76.2 at par of 96, where the same PPO run from a freshly initialised network completes 73.5 but passes the quality bar on 10 and reaches the bound on 7, with a median excess over par of four; on the larger domains nothing it completes is usable (Table 9, right). Cloning alone completes 92.8 and is usable on 73.5 but reaches par on only 22.3; a million PPO steps from it take that to 76.2. This is the exploration wall of Section 5: without a supply of solved instances the policy rarely sees the event that the reward targets.

DCEL connectivity has to be an input. We train a Transformer encoder with the same parameter count, features, action space, data, reward and budget, whose input is a breadth-first serialisation of the template with a positional encoding of slot order. This is a lossy serialisation of the adjacency. It reaches par on about half as many domains as the convolution, and already lagged at the end of cloning (Table 9, left). We read this as a difference in what the input identifies rather than in capacity, and expect an attention model masked or biased by `next`, `previous` and `twin` to do comparably well.

Run-to-run variance. Two training seeds of the released configuration without its quality term, and two seeds of the whole pipeline (new certified instances, cloning and PPO), differ in distribution by at most 0.7 domains on usable and 2.7 on at par; on the larger set, by 3–5 on usable and about 10 on at par. Out of distribution the seed spread is about five domains, above the evaluation-seed spread of Table 8, so the larger-boundary numbers should be read as ranges.

Table 9: Left: domains at par of the 96 in-distribution domains, best of five, at a matched 2M steps under an earlier configuration of the recipe, mean of two seeds. Right: PPO from a fresh network, cloning only, and cloning then PPO, at 1M PPO steps, seed 1, best of five at the default move budget and without repair.

extractor	params	at par
DCEL, 4 layers	299k	**94.5**
Transf., 4 layers	300k	51.5

set	training	all-quad	usable	excess	at par
96	PPO	73.5	10.2	4	7.2
	clone	92.8	73.5	2	22.3
	clone + PPO	95.5	91.8	0	76.2
64	PPO	13.5	0.0	16.75	0.0
	clone	32.5	0.0	6.6	0.0
	clone + PPO	51.5	14.0	2	16.0

K GENERATING DOMAINS, AND DRAWING THE MESH

Outlines. Training and evaluation domains come from GEO2D, a seeded generator of lattice-based mechanical parts. A shape begins as an occupancy raster on an integer lattice — a base rectangle, then rectangular cuts, additions and combs anchored on the current boundary, each kept only if the material stays connected and free of pinches and voids — and tracing that raster gives a counter-clockwise lattice polygon, to which corner modifiers ($45°$ chamfers for the straight-sided families; fillets and semicircular notches for the curved ones of Appendix L) and interior holes with at least one lattice unit of clearance are applied with exact geometry. One lattice unit is by construction the smallest feature, so a domain of ratio R fits in an $R \times R$ box and admits a uniform mesh of unit size, which keeps element counts comparable across a sampled set. A preset fixes the parameters and a seed determines a domain exactly, so a training distribution and a held-out evaluation set are specified by a preset and two seed ranges. The larger-boundary set also caps the ratio of longest to shortest boundary edge at ten, so a drop there measures size rather than geometric extremity.

Smoothing. The edit operations of Section 3.2 set connectivity and say nothing about where nodes sit, so every geometric property of the result is the smoother's doing. Between moves we run a few Laplacian sweeps, which are cheap and keep the observed angles meaningful. Where quality decides something — the solved test, and every reported number — the mesh is instead untangled: for each free node a condition-number objective over its adjacent corners is minimised by Newton steps, with the corner Jacobian entering through $h(J) = (J + \sqrt{J^2 + \delta^2})/2$ so that an inverted corner is pushed back out rather than sending the objective to infinity (Escobar et al., 2003). The corners of Ω never move; nodes the agent inserted on a boundary edge slide along it toward the same objective, and a slide is accepted only if the worst adjacent corner does not get worse (Freitag, 1997). The same smoother, with the same settings, is applied to every method before it is scored.

L CURVED BOUNDARIES

The bound on curved domains. Section 2 states the bound for polygons, but its proof uses no straightness. A vertex inside an arc has $\theta_v = \pi$ by tangent continuity, so the only geometry entering equation 2 is again the set of corner angles: a smooth arc contributes to par exactly as a straight edge does. A square and a disc are both topological discs, but the square's four corners give $\mathrm{par} = 0$ while the disc, with none, has $\mathrm{par} = 4$.

The repair search. Section 6.4 runs the released agent, trained only on straight-sided domains, with no change to its weights, observation or action space. The domains are the four curved suites, with fillets and semicircular notches: 48 each at 8–24 and 25–50 corners, with and without holes (16 for the holed larger suite), every arc discretised at $45°$. The agent's five attempts of Section 6.1, at the doubled move budget, supply a pool of all-quadrilateral states, each untangled with the arcs' tangents. The search then repeats: take the most regular states still below the quality bar; at the worst element of each apply one edit, either refining the quadrilateral sheet through it (all-quadrilateral in and out, every new vertex at its want, so the excess over par is unchanged), or inserting a vertex on one of its sides or a chord across it and letting the agent finish the mesh; untangle the results and add

them to the pool. It stops after 600 s. The answer is the mesh with the lowest excess over par among those that clear the bar, ties going to fewer elements, so that refinement is kept only when it buys the bar. The row without the search in Table 2 selects from the same pool by quality first, as Section 6.1 does. The search judges quality with each corner on an arc measured against the arc's tangent; this is consistent with block decomposition wherein refinement will add new vertices on the arc and not on the chord. However, Gmsh often fails the quality bar when measured on tangents, therefore we report quality measured by the tangent as well as the chord for fairness.

Table 10: Table 2 by suite, with the larger domains; every suite is curved. Columns, rows and win rates as there; our regularity win rate is 100% against every configuration in every suite and is omitted. Bold: best in column within a suite. The holed larger suite has 16 domains, the others 48.

							usable		quality win	
suite	method	all-quad	excess	at par	elem.	×ours	tangent	chord	tangent	chord
8–24	Gmsh blossom	35	18	0	60	1.1	3	31	94%	50%
	Gmsh frontal-quad	30	18	0	58	1.1	2	23	100%	77%
	Gmsh quasi-structured	**48**	9	0	81	1.6	3	33	94%	71%
	Gmsh blossom-full	**48**	31	0	249	4.8	13	44	73%	46%
	Ours	**48**	**2**	**21**	**52**	1.0	**47**	**45**		
	no search	48	2	11	51		39	42		
holes, 8–24	Gmsh blossom	27	24	0	61	1.1	4	23	94%	58%
	Gmsh frontal-quad	25	22	0	59	1.1	0	19	100%	79%
	Gmsh quasi-structured	**48**	14	0	84	1.5	3	31	96%	62%
	Gmsh blossom-full	45	32	0	221	4.0	12	35	75%	48%
	Ours	**48**	**2**	**16**	**55**	1.0	**48**	**43**		
	no search	48	4	8	54		35	38		
25–50	Gmsh blossom	28	64	0	185	1.1	2	23	96%	54%
	Gmsh frontal-quad	24	52	0	191	1.2	0	19	100%	60%
	Gmsh quasi-structured	**48**	25	0	323	2.0	0	33	100%	31%
	Gmsh blossom-full	**48**	98	0	1064	6.5	10	**41**	79%	15%
	Ours	**48**	**8**	**1**	**164**	1.0	**36**	30		
	no search	48	8	0	138		12	13		
holes, 25–50	Gmsh blossom	5	60	0	201	1.1	1	5	100%	81%
	Gmsh frontal-quad	10	58	0	179	1.0	0	9	100%	56%
	Gmsh quasi-structured	**16**	32	0	295	1.7	0	10	94%	25%
	Gmsh blossom-full	**16**	117	0	960	5.4	**4**	**14**	75%	19%
	Ours	**16**	**13**	**1**	**178**	1.0	**4**	6		
	no search	16	12	1	153		2	2		

On the larger domains the regularity result holds, a median excess over par of 8 and 13 against 25 to 117, but the agent clears the quality bar less often on larger domains. One domain per suite is drawn in the gallery (Figure 8).

M GALLERY

Figures 6 and 7 show one domain from each family of the two evaluation sets. We show domains whose agent result sits closest to the family's medians of corner count, element count and minimum quality. Gmsh BLOSSOM is asked for the agent's element count on that domain and returns the closest it can reach; QUASI-STRUCTURED runs at its natural size; both are untangled with the same smoother as the agent's mesh. A marked vertex is one whose degree differs from every degree its position wants (Section 2).

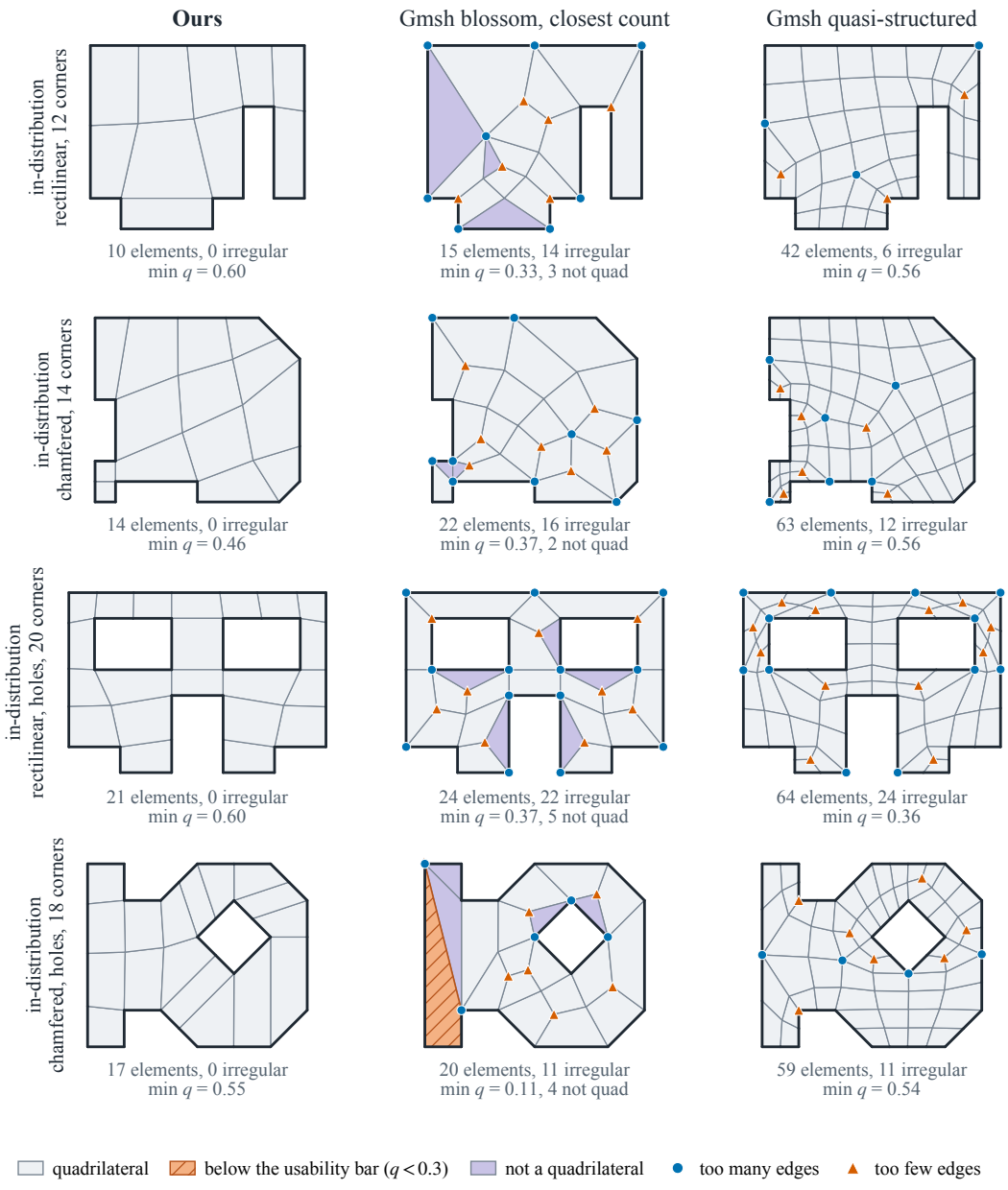


Figure 6: One domain per in-distribution family (8–24 corners).

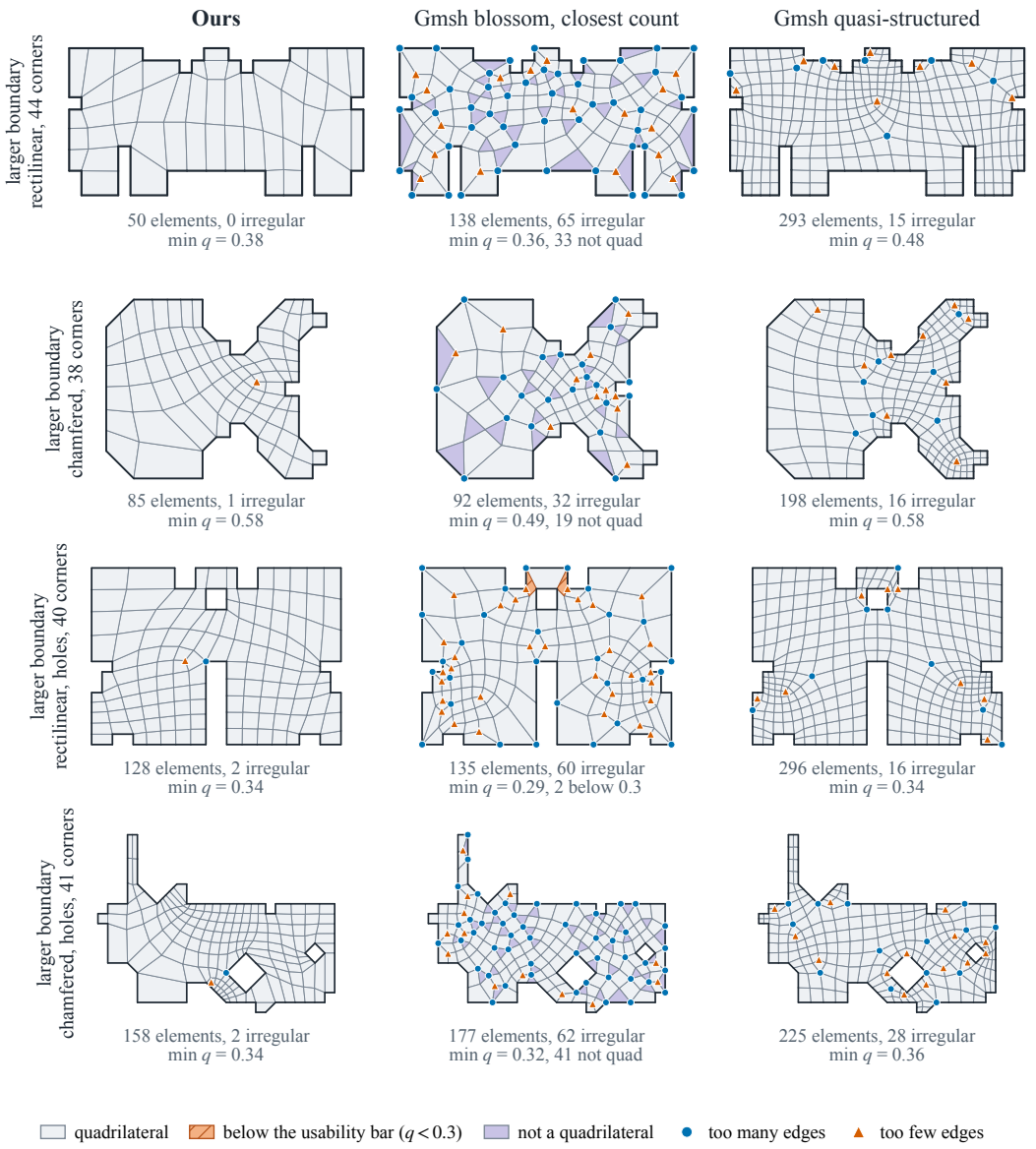


Figure 7: Larger-boundary domains (25–50 corners), beyond any seen in training: one per family.

Figure 8 shows one domain per curved suite of Section 6.4, meshed by the released agent with the repair search, chosen by the same rule among domains with at least one arc.

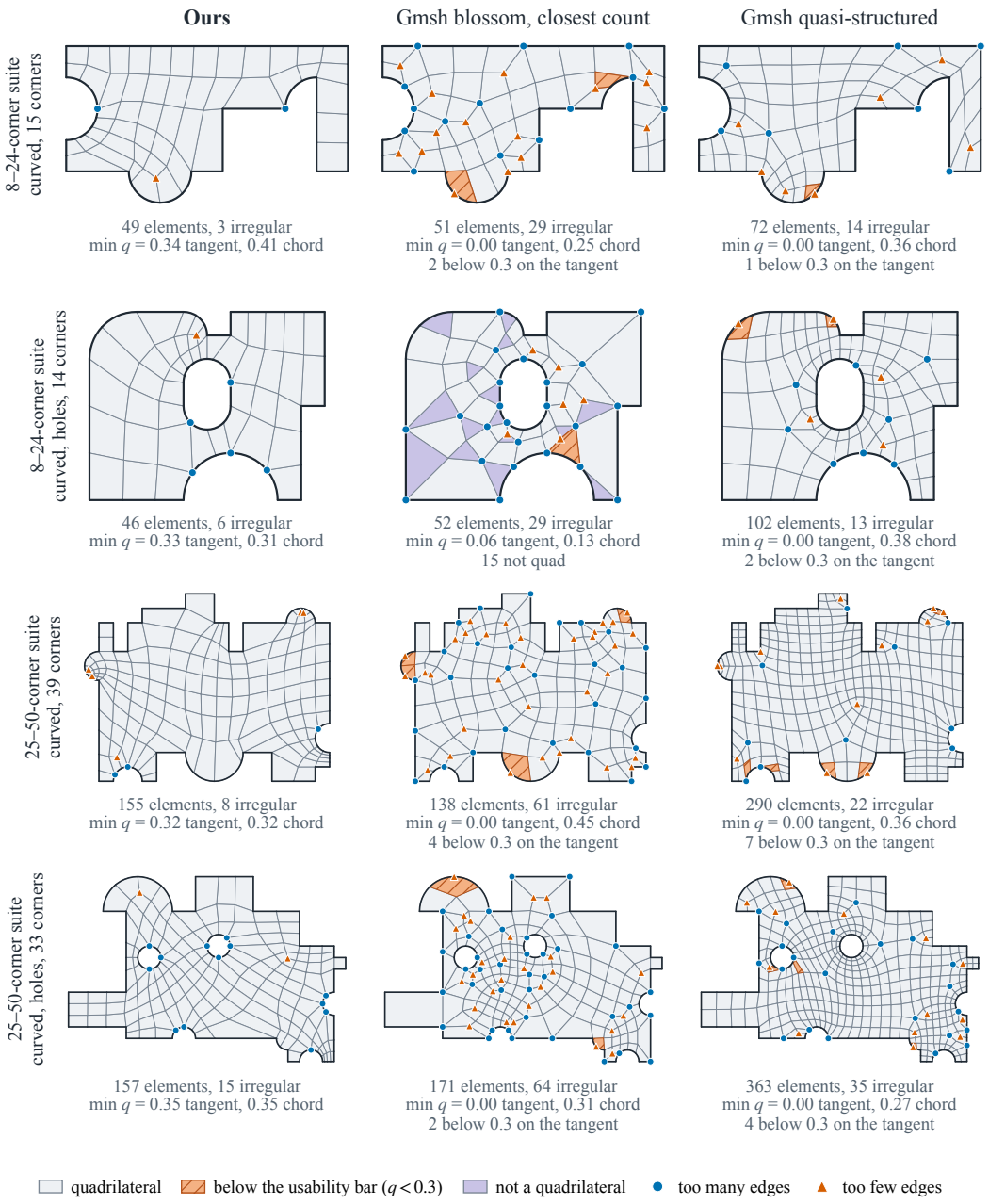


Figure 8: Curved domains, one per suite, meshed by the released agent, trained only on straight-sided domains, with the repair search, and by Gmsh as in Figure 1. A boundary edge that lies on an arc is drawn along the arc; ours has a boundary node every 45° of arc. Each panel gives the minimum element quality judged against the arc tangents and on the straight sides (Section 6.4); the hatching uses the tangent.