
Learning to Guess the Distance

A learned heuristic for A* — and what it took to know whether it actually helped

A* finds shortest paths fast when its heuristic — its guess of the distance still to go — is good. Manhattan distance is the classic guess, but it's blind to walls. Could a model learn a better one? The surprising part of this project isn't the model. It's how much work it took to *know* whether the model helped at all — and how the answer kept changing as we looked closer.

THE AVERAGE LIES

Pooled over all mazes the model looked like a wash — until we stopped pooling.

ONE FEATURE FLIPS IT

A single global feature turned the wash into a clear, optimality-safe win.

IT WAS NEVER THE FEATURE

The real lever was the training distribution; the feature just told the model which world it was in.

THE MODEL FORGETS OFF-DISTRIBUTION

Train on one kind of maze, test on another, and it fails — in two opposite ways.

Did the model actually help?

It is an easy question to answer badly. Train a model to predict cost-to-go, plug it into A* as the heuristic, measure the nodes it expands, and report a number. We did that, and the first number said: barely. The honest work was everything that came after — because the single pooled number turned out to be hiding the truth, twice over.

Two axes, never one. A heuristic is good only if it cuts the work A* does (**nodes expanded**) *without* sending it down longer paths (the **optimality gap**). Collapsing those into one score is how you fool yourself. Every claim here is also checked *within* maze type, not just pooled — the same discipline that keeps an average from lying.

The turn	What it revealed
The average lies	Pooled, the model was a wash (~8% fewer nodes at a 4.8% optimality cost). Split by maze type it reversed: a clean win on corridors, a loss on open fields. A Simpson's-paradox trap.
One feature flips it	Adding global obstacle density: 17% fewer nodes at a 0.2% gap , 97% of mazes solved optimally.
It's the data, not the feature	Trained on open fields alone, the plain model already wins (34%). Global density is a <i>regime tag</i> that lets one model serve a mixed distribution — not a magic calibrator.
It forgets off-distribution	Train on one maze type, test on the other, and it fails in opposite ways: too timid (-2% nodes) one direction, too reckless (14% gap) the other.

1 · A heuristic worth learning

Of all the parts of A*, only one is a guess. The path is computed; the cost-so-far is known; but the heuristic — the estimate of remaining distance — is, by design, an approximation. That is exactly where machine learning belongs: predict the true cost-to-go from cheap features, and use the prediction as the heuristic. The true cost-to-go itself we get for free and exactly, by a single backward breadth-first sweep from each goal — perfect labels, so any error is the model's, not the data's.

The honest scaffolding matters as much as the model. We split the data by *whole maze*, never by cell, because cells in one maze are correlated — a random split would leak the test answers through their neighbours (and the code asserts the split is clean). The baseline isn't a strawman: it's the Manhattan heuristic A* already uses, which is admissible and good. And we judge on two axes — nodes expanded and optimality gap — reported within maze type. Two environments appear throughout: open fields of scattered obstacles, and structured corridor mazes.

Two scoping notes up front, in fairness to the reader. Everything here is simulation — two maze generators only (open scattered fields and perfect corridor mazes) on a 4-connected unit grid — so the findings describe this controlled world, not real road networks. And the model is deliberately simple: a gradient-boosted regressor on a handful of features. This is a deliberately basic ML project; its contribution is the evaluation discipline, not model sophistication.

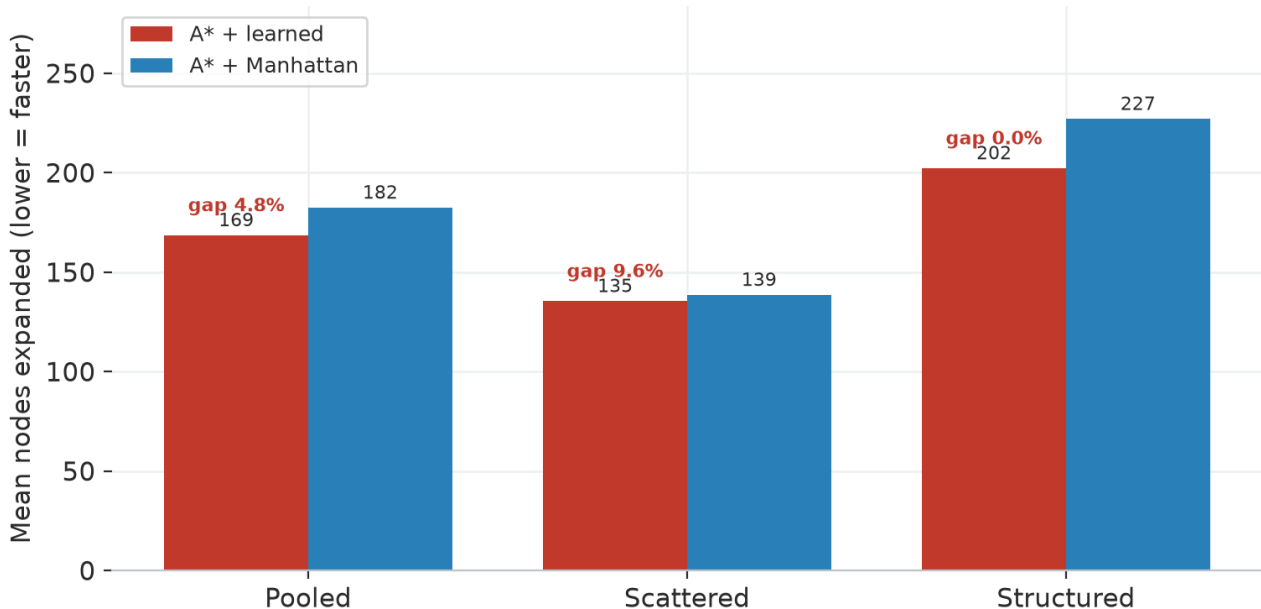
A heuristic is only worth learning if it beats a strong baseline on the metric you actually care about — and proves it on data it never saw.

2 · The average lies

The first end-to-end result was deflating. Pooled over every held-out maze, the learned heuristic expanded about 8% fewer nodes than Manhattan — but paid a 4.8% optimality gap for it. A wash with a cost. If we had stopped there, the verdict would have been: it didn't work.

The average hides two opposite stories

Six-feature model. Pooled looks like a wash; split by maze type it reverses



Six-feature model. Pooled, learned and Manhattan look similar; split by maze type, the story splits with them.

Then we refused to pool. Inside *open fields*, the model was actually worse — it gave up ~10% optimality for no real speed gain, because Manhattan is already nearly exact in open space. Inside *corridor mazes*, it was a clean win — about 11% fewer nodes at no optimality cost, because Manhattan is a poor guide where walls force long detours. Two opposite effects that cancel in the average.

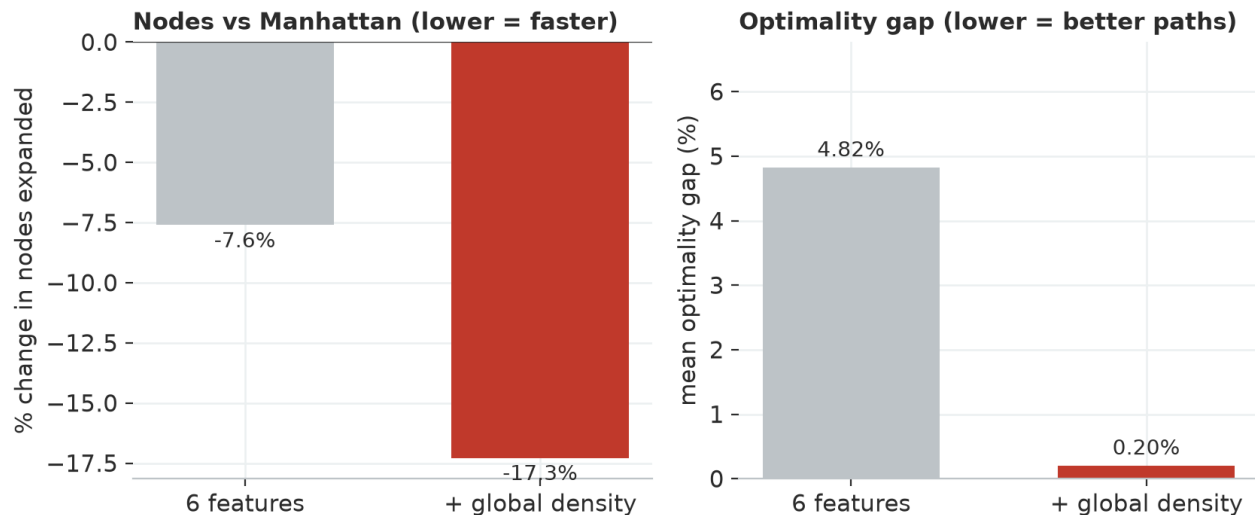
Pool two opposite effects and the average reports that nothing happened. The truth lives within the groups.

This is Simpson's paradox in miniature, and it is the spine of the whole report: a pooled metric can not only blur a result but invert it. Every number that follows is read within maze type first.

3 · One feature, and the verdict flips

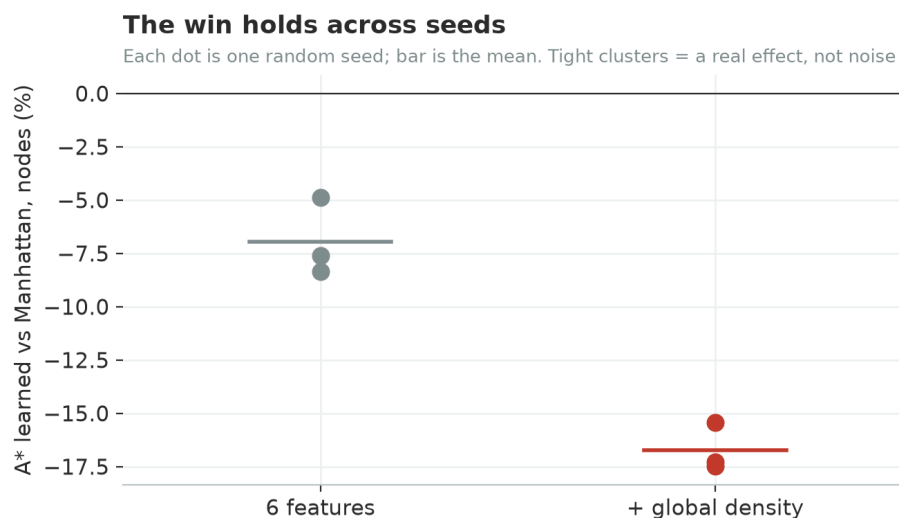
If the model helps where Manhattan is weak and hurts where it is already strong, the missing piece is something that tells it which situation it is in. We added one feature: the maze's overall obstacle density.

One feature turns a wash into a win



Adding global obstacle density: a much larger node reduction, and the optimality gap nearly vanishes.

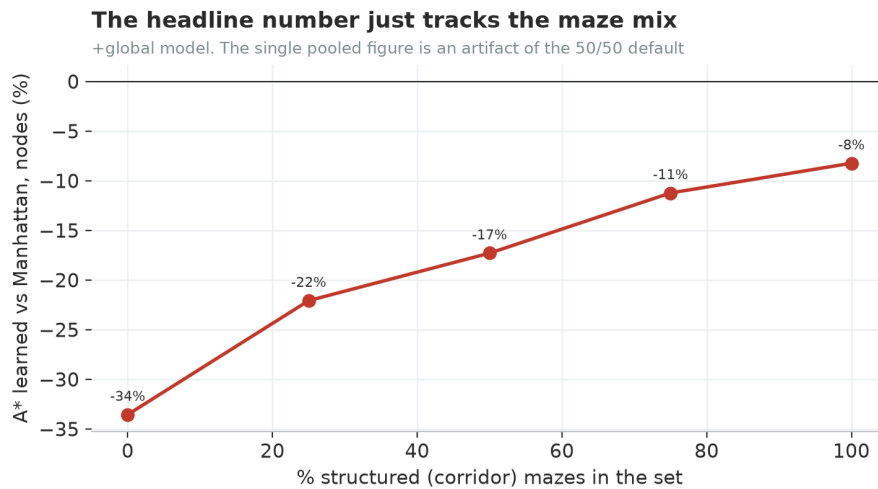
The verdict flipped. The learned heuristic now expands about **17% fewer nodes than Manhattan at a 0.2% optimality gap**, solving 97% of held-out mazes on the exact shortest path. The open-field regime, previously a loss, turned into a ~28% win — and that one is genuinely earned: open fields have many competing paths, so its near-zero gap is real. (The corridor 0% gap is partly free — perfect mazes have a single path, so any route found there is optimal — a point we return to in the limits.)



Each dot is one random seed; the win clusters tightly — not a one-seed fluke.

And it is not one lucky run. Across three random seeds the +global model expands between 15% and 17% fewer nodes, with the optimality gap never above 0.2%, and a 10,000-maze spot-check

agreed. The six-feature baseline is just as consistently a near-wash. The effect is the feature, not the seed.



The +global node-win as the maze mix shifts; the single headline is just the 50/50 point.

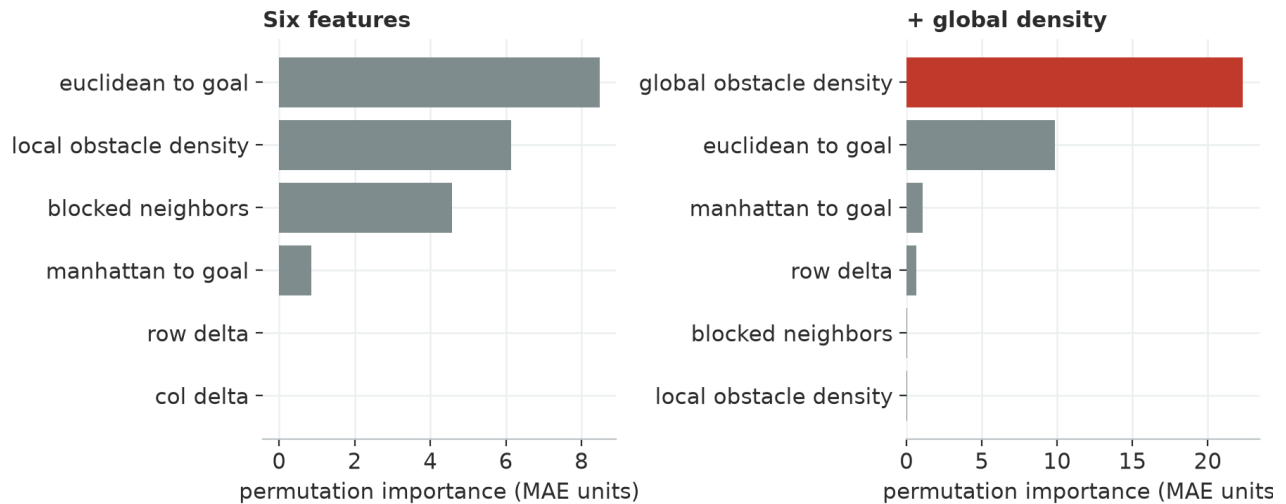
One honest qualifier on that headline: the ~17% is the 50/50 figure. Slide the proportion of corridor mazes and the pooled win slides with it — larger in open-heavy sets, smaller in corridor-heavy ones. The same lesson again: a single pooled number is an artifact of the mix, so we read every result within maze type.

A learned heuristic can beat a strong, admissible baseline on real search — given the one piece of context it was missing.

4 · It was never the feature — it was the data

The tidy story would end there: we found a good feature. But asking *why* it worked overturned that story — and this is the most important page in the report.

What the model leans on — and what global density makes redundant



Permutation importance. With global density present, the local-clutter features the model leaned on before collapse to near-zero — it subsumes them.

Two clues. First, importance: global density dominates everything, and the local-clutter features the six-feature model relied on drop to ~zero — it didn't *add* information so much as *replace* a noisy proxy for it. Second, and decisively: trained on open fields *alone*, the plain six-feature model already crushes them — about 34% fewer nodes, with a tiny error (MAE 3.9) and *no global density at all*. Trained on corridors alone it also does well (MAE 55). Global density adds almost nothing in either pure case.

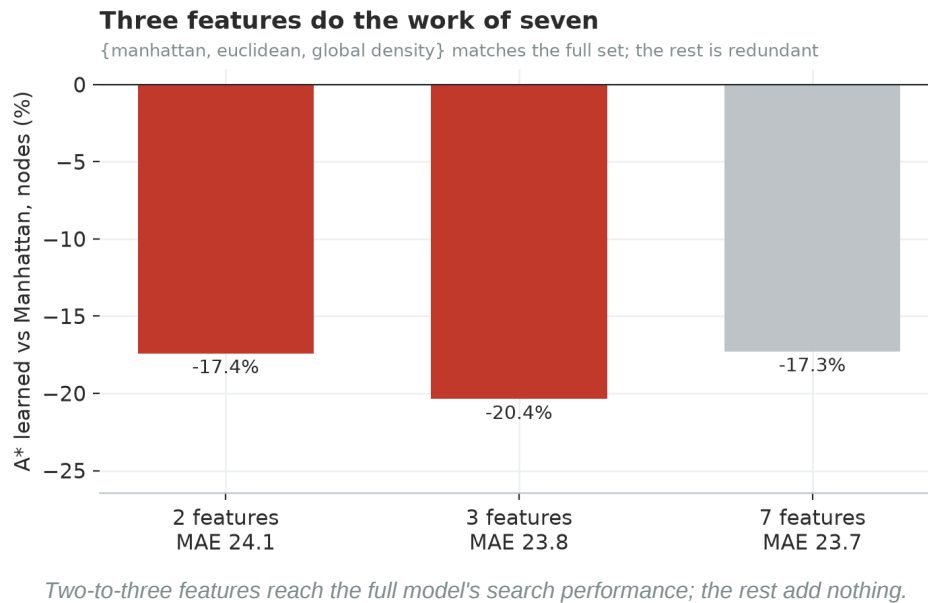
So global density is not an open-field calibrator. It is a **regime tag**. The earlier failure was never intrinsic to open fields — it happened only when one model was trained on a *mixture* of both maze types and had no way to tell them apart, so it compromised, and the open-field regime lost the compromise. Give it a feature that identifies the regime and the compromise dissolves.

The result was governed by the training distribution, not the model.

Heterogeneous data needs a group-identifying feature — or a model per group.

5 · How little it takes

If global density subsumes the clutter features, most of the seven should be removable. They are. A model with just three features — Manhattan and Euclidean distance plus global density — matches the full set (about 20% fewer nodes), and prediction error stops improving past three. The local-density and dead-end features were pure redundancy.

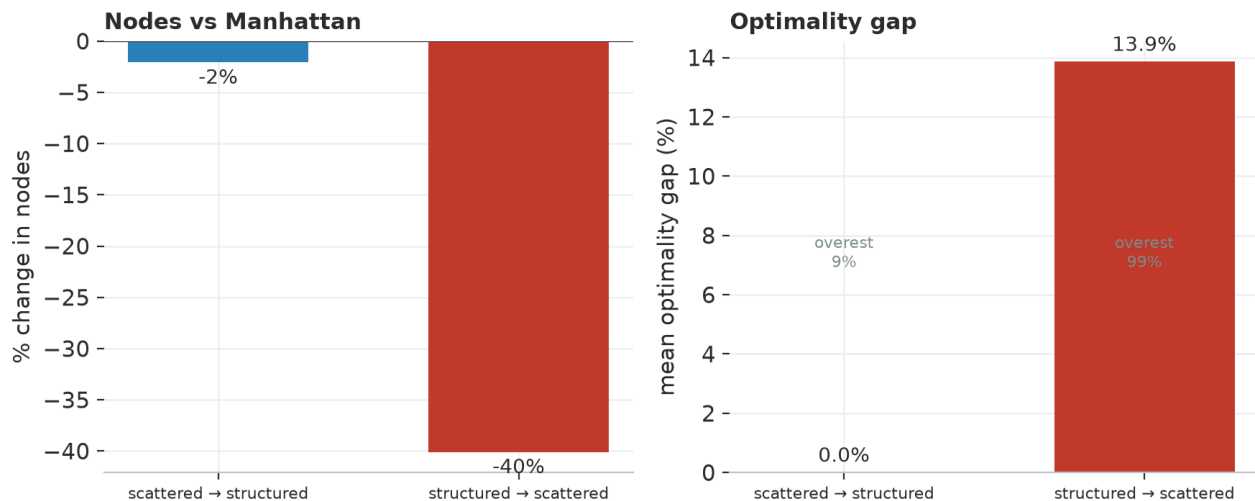


Parsimony, earned empirically: the smallest honest model is the one that says everything the big one did.

6 · The model only knows what it trained on

If the training distribution is the real lever, the sharpest test is to break it on purpose: train on one maze type and test on the other. It fails completely — and, tellingly, in two *opposite* ways.

Off its training distribution, the model fails — in opposite ways



Cross-distribution transfer. Trained on one regime, tested on the other, the heuristic either goes limp or goes reckless.

Trained on open fields and tested on corridors, it *under*-estimates the long detours (it overestimates only 9% of cells), so the heuristic goes limp — A* expands about as many nodes as plain Manhattan (-2%), drifting toward blind search, but stays optimal. Trained on corridors and tested on open fields, it *over*-estimates almost everywhere (99% of cells), turning A* greedy: 40% fewer nodes but a 14% optimality gap — fast and wrong. Same model, same features; only the training distribution changed.

A quiet signature confirms it: when tested off-distribution, the features' importance collapses toward zero — the model has stopped responding to its inputs, emitting a mis-scaled answer it learned elsewhere. (And global density doesn't rescue the transfer: the test density falls outside its training range, so it is extrapolating.) This is distribution shift made visible.

A learned heuristic is only as good as the distribution it learned. Move the world and it doesn't adapt — it mis-remembers.

What we didn't claim

A few honest limits, left open on purpose — each a real next step rather than a hidden flaw:

The corridor optimality is partly free. Our structured mazes are *perfect* mazes — exactly one path between any two cells — so any path found there is optimal regardless of the heuristic. The defensible corridor win is the node reduction; whether an inadmissible learned heuristic stays optimal once mazes have loops (braided mazes) is untested, and is the first experiment we'd run next.

The metric you optimise is not the metric you want. The model was trained to minimise prediction error, but A* cares about ranking, not absolute accuracy — which is why a model with a large MAE on corridors still guides search well, and why we always checked the search behaviour, not just the error. Two layers, never conflated.

Cheap features have a ceiling. A regional feature (walls on the straight line to the goal) added nothing once global density was present, and corridor routing is fundamentally a *global* property — whether a passage dead-ends depends on far-away structure a cheap feature can't see. Pushing past this likely means feeding the whole maze to the model (Neural-A*-style), which is a different, heavier project.

Admissibility is recoverable. The learned heuristic is inadmissible by default; the code already has a transform hook to scale predictions down. Whether an admissibility-constrained model keeps the speed win is an open, answerable question.

What the project is really about

Set the turns side by side and the subject isn't really pathfinding. It's the discipline of knowing whether a result is real: refusing to trust a pooled average (it inverted on us); beating a strong baseline rather than a strawman; separating the metric you can optimise from the objective you care about; and — the turn that mattered most — being willing to overturn your own explanation when the data demands it. “We found a good feature” became “the training distribution was the lever all along.”

That last lesson is the one worth carrying: a model is a compression of the distribution it was trained on, and it is only trustworthy inside that distribution. The most useful thing this project produced is not a faster heuristic — it is a small, reproducible apparatus for asking, honestly, whether a model helped at all.

The hard part was never training the model. It was earning the right to say it worked.

Principles in play: Simpson's paradox; baseline discipline; data leakage and group-wise splitting; permutation importance (and its correlated-feature caveat); mixture-of-distributions; distribution shift / out-of-distribution generalisation; admissibility (Hart, Nilsson & Raphael, 1968); metric-objective alignment. · Code, tests, and the full decision log: github.com/arda-basarici/pathfinding-ml