REASONING OVER SIMULATIONS: SYMBOLIC INTERMEDIARIES FOR LLM DESIGN AGENTS WITH A NUMERICAL ORACLE

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning to improve a mechanism from simulation requires connecting observed motion to changes in its design. For language agents, this is a representation problem: simulated trajectories provide numerical evidence, while design revisions are proposed through linguistic reasoning. We introduce symbolic intermediaries, compact analytical expressions fitted to simulated trajectories and supplied alongside the target description. They expose geometric structure that agents can compare, discuss, and use to formulate revisions. A design, simulation, critique, and memory loop accumulates this experience at inference time while the language model's parameters remain fixed. Our analysis characterizes the information needed for local revisions, conditions for their transfer, and boundaries of the feedback representation. We evaluate the approach on MSynth, covering six planar target-curve families and three language models. Critique supports sustained refinement across models, while symbolic feedback provides an additional benefit that depends on the model and target motion. With critique and memory matched, the share of runs improving after the first third of search rises from 44.4% to 55.2%, and the median share of total improvement achieved in that later phase rises from 5.4% to 15.6%. Critique transcripts show agents using the fitted expressions in geometric comparisons and calculations. These results connect inference-time learning to the representation of simulation experience, where analytical descriptions can help language agents use observed mechanism behaviour to guide subsequent design decisions.

1 INTRODUCTION

Designing a mechanism for a desired motion means connecting *structural choices* to *physical behaviour*. In a planar linkage mechanism, the arrangement of joints and links, together with their dimensions, determines the trajectory traced by an output point as the mechanism moves. While a language agent can propose such a design as an executable program, and a kinematic simulator reveals the motion it produces, improving the next mechanism proposal requires turning that motion into *useful feedback*. In this paper we focus on the question *how should simulated mechanism behaviour be represented so that language agents can learn to improve their designs through interaction?*

This is difficult as geometric discrepancies must be connected to changes in a coupled physical system, characterizing an inverse mechanism design problem. Adjusting one link can affect the entire output trajectory, and similar trajectory errors may require different changes in topology or dimensions. A distance score ranks candidates and names no defect, while properties such as curvature, symmetry, and the presence of a loop stay implicit in the simulated coordinates. Language models are documented to misread geometry delivered in that form (Mouselinos et al., 2024), while they contribute to problems as hard as olympiad construction once the same geometry is stated symbolically (Trinh et al., 2024).

Language agents can improve their outputs through critique and accumulated context (Shinn et al., 2023; Madaan et al., 2023), and agentic work already reads its verdict off a number a simulator returns, e.g., designing rewards in a physics engine (Ma et al., 2023), generating and scoring simulated tasks (Wang et al., 2024b), grounding plans against a robot simulator (Rana et al., 2023; ichter et al.,

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.

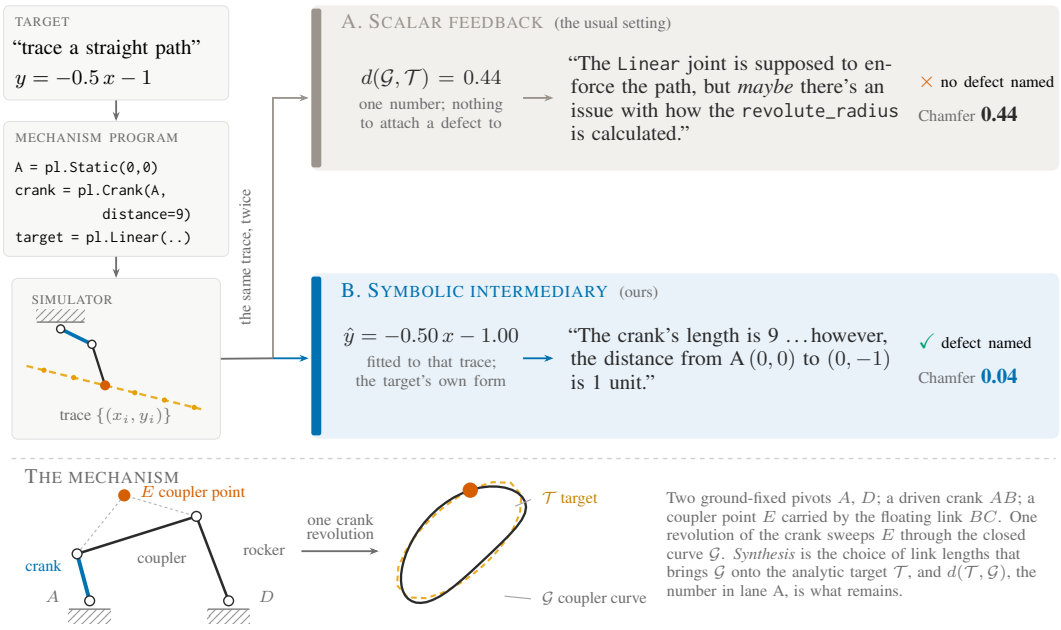

Figure 1: **Only the form of the feedback differs.** A design agent writes an executable mechanism program; the simulator returns a numeric trace. Collapsing that trace against the target gives a *scalar* (lane A), which ranks candidates but names no defect, so the critique can only hedge. We instead fit a compact analytical expression to the same trace by symbolic regression and set it beside the target's own equation (lane B, the *symbolic intermediary*); the critique then shifts to grounded geometric verification. Quotes and Chamfer distances are a matched Qwen 4B pair on a line target, identical in every other respect.

2023), closing a driving loop (Fu et al., 2024). In mechanism synthesis, connecting these capabilities requires translating numerical behaviour into feedback from which the agent can formulate design changes. We investigate this interface as a representation problem. Our hypothesis is that *exposing analytical structure in simulated motion supports more specific geometric critiques and more informed subsequent proposals*. This connects learning through simulation to the role of representation in making useful inferences accessible (Larkin & Simon, 1987).

In this context, we introduce *symbolic intermediaries*: compact analytical expressions fitted to simulated trajectories through symbolic regression (Cranmer, 2023; Udrescu & Tegmark, 2020). An intermediary places the observed motion and the target equation in a shared mathematical representation, so the agent can compare their functional forms and express discrepancies in natural language. For example, a fitted expression can expose curvature in a trajectory intended to follow a straight segment, giving the critique a concrete feature to examine alongside the mechanism's joints and dimensions. The intermediary enters the critique alone and never the objective, so the simulator remains the sole arbiter of success and a fitting error degrades guidance without corrupting correctness (§2.2).

This representation supports a repeated cycle of design, simulation, critique, and revision. Each simulation supplies new evidence, while the critique interprets it and conditions subsequent proposals. Executable designs accumulate in an external memory the next iteration retrieves from, so the agent adapts through its evolving context and memory while the language model's parameters remain fixed. We use *inference-time learning* for improvement supported by this accumulated experience, following Mitchell's operational criterion (Mitchell, 1997), with no gradients, no labels, and no fine-tuning (§2.3).

Planar mechanism synthesis makes this question experimentally accessible (Fig. 1), as it connects discrete topology and continuous dimensions to an observable trajectory, with simulation fitting the target geometry providing an external measure of design quality.

Our contributions are threefold. First, we present a novel synthesis algorithm for mechanical design, which integrates a symbolic intermediary as a representation of the simulator (oracle) output to Large Language Model (LLM) agents. A design agent proposes executable mechanism code, which is evaluated by a simulator, and a revision step identifies geometric deficiencies to be corrected and commits valid designs to an external memory. The result satisfies Mitchell's criterion for learning (Mitchell, 1997) with no gradients, no labels, and no fine-tuning (§3.1).

Second, we introduce symbolic intermediaries as the diagnostic channel, and show through systematic scoring of critique transcripts that the symbolic intermediaries change the agents' reasoning process. Engagement with discovered symbolic structure rises for every model, and in reasoning-trained models the critiques shift toward grounded geometric verification (§3.3).

Third, we release *MSynth*[1], a benchmark of six analytic planar curve families, and report the ablations that separate what critique feedback contributes from what the symbolic layer adds (§3.4, §3.5).

2 Symbolic Intermediaries

The method connects four representations of a design problem, namely: an executable mechanism, its simulated trajectory, a fitted analytical expression, and a natural-language critique. Each serves a different purpose. The program specifies joints, connections, and dimensions; the trajectory supplies evidence about the resulting motion; the expression exposes features of that motion; and the critique translates observed discrepancies into proposed design changes. Repeating this cycle lets the agent use simulation experience to refine its proposals at inference time while its language-model parameters remain fixed.

Given a target trajectory $\mathcal{T}$ specified by an analytic equation, find a mechanism $\mathcal{M}$ whose end-effector traces a path $\mathcal{G}_{\mathcal{M}}$ minimising

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \; d(\mathcal{T}, \mathcal{G}_{\mathcal{M}}) \,, \tag{1}$$

where designs are sampled from a pre-trained language model π_θ that is never updated. Fig. 2 shows one iteration of the resulting loop on a line target; Algorithm 1 gives the full procedure.

2.1 Representing designs and simulated motion

In planar linkage synthesis, the arrangement and dimensions of connected links determine the trajectory of a selected output point. A four-bar linkage, for example, has two ground-fixed pivots and a moving coupler whose motion depends on the link lengths and assembly configuration. Producing a desired curve requires coordinating these geometric choices with the mechanism's topology, a challenge studied extensively in dimensional and computational synthesis (García-Marina et al., 2020; Tsai, 1999; Romero et al., 2019; Vasiliu & Yannou, 2001; son, 2024).

We represent each candidate as an executable program using the simulator's API. This exposes topology and dimensions in one editable object, where the agent can change joint types and connections as well as their numerical parameters, within the constructions supported by the API. Executing the program in $\mathcal{S}im$ instantiates the linkage and produces an end-effector trace $\mathcal{G} = \{(x_i, y_i)\}_{i=1}^{n}$ over a crank rotation. The target $\mathcal{T}$ supplies both an analytic equation and sampled points. Candidate quality is measured by the Chamfer distance d between simulated and target point sets after rigid alignment by Iterative Closest Point (ICP) (App. G). Programs that fail to parse, execute, or produce a trajectory do not yield a scored candidate.

These representations support complementary operations: (i) *code* makes a proposed mechanism executable and revisable, (ii) *point sets* support geometric evaluation, (iii) *equations* make aspects of the target motion accessible to calculation and linguistic comparison. The missing connection is an analytical description of the motion the proposed mechanism actually produces. We construct that description from the simulated trace and supply it as feedback.

[1] anonymized

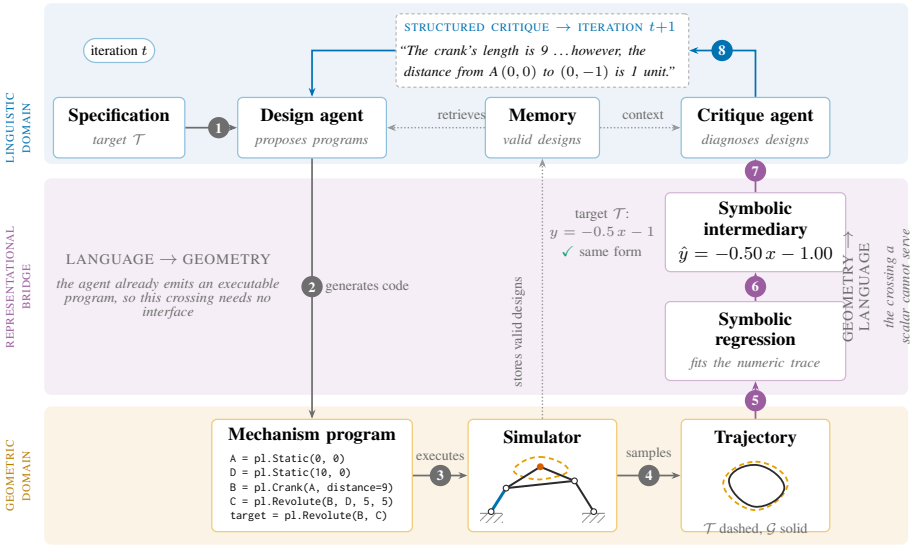

Figure 2: The linguistic–numerical interface in three layers. In the *linguistic domain* a design agent proposes mechanism programs and a critique agent evaluates them; in the *geometric domain* each program is assembled into a linkage that sweeps a trace. On the *representational bridge* between them a scalar names no defect, so symbolic regression fits the trace into a compact analytical expression, the *symbolic intermediary*, which the critique agent reads in the same notation as the target equation and can use to name a geometric defect.

2.2 From numerical trajectories to symbolic feedback

The trace and the distance fail in opposite directions. $\mathcal{G}$ retains the geometry of the produced motion but holds it implicitly, spread across hundreds of coordinates that name no property of the curve; d names a single number that ranks candidates and discards everything about how the two curves differ. Neither is expressed in the notation the target is written in, so neither supports the comparison the agent needs to make. The essential gap is a form that is compact enough to be interpreted and structured enough to compare, which is what a closed-form expression provides.

We therefore fit one. Given the trace $\mathcal{G}$, Symbolic Regression (SR) (Cranmer, 2023; Schmidt & Lipson, 2009; Petersen et al., 2021; Udrescu & Tegmark, 2020) searches a space $\mathbb{F}$ of expressions built from arithmetic and trigonometric primitives for

$$\hat{S} \;=\; \arg\min_{S \in \mathbb{F}} \; \frac{1}{n} \sum_{i=1}^{n} \big(S(x_i) - y_i\big)^2, \tag{2}$$

We call $\hat{S}$ the *symbolic intermediary*.

The representational change is from coordinate-level evidence to an expression that *exposes analytical structure*. With the fitted and target expressions in context, the agent can compare functional forms, calculate geometric quantities, and formulate a critique at a linguistic/symbolic level. In addition to the program's explicit joints and dimensions, this comparison supports hypotheses about which design changes could reduce the observed discrepancy. The intermediary therefore connects descriptions of behaviour to reasoning about the mechanism that generated it.

The fit is computed from the simulated trajectory after scoring and supplied in the prompts used for feedback and refinement. Analytical descriptions retained in context also accompany examples available to the design agent. They guide subsequent proposals, where candidate ranking continues to use the simulated point sets in Eq. 1. This keeps the experimental question fully on target, i.e. *does an analytical representation of observed motion help the agent use simulation feedback more effectively?*

2.3 THE MECHANISM FEEDBACK LOOP

Design. At iteration t the design agent receives the task specification, the simulator's API, retrieved examples, and the target equation/intermediary (App. I), and samples b candidate programs. Execution determines which proposals yield trajectories. Successful candidates are scored and their traces are fitted with an intermediary representation.

Critique. The critique agent receives the proposed program, simulator output, distance, and available symbolic feedback alongside the target specification. It assesses consistency with the task, identifies errors, considers structural complexity, and suggests refinements (App. J). This step turns the numerical and analytical observations into a linguistic/symbolic-level account of what should change and why.

Revision. The design agent uses this critique and the available feedback to produce a revised program (App. K). The next simulation evaluates the proposed change, adding evidence about whether the diagnosis led to a better mechanism. The loop terminates when

$$d(\mathcal{T}, \mathcal{G}_{\mathcal{M}_t}) \leq \epsilon \quad \text{or} \quad t \geq R_{\max}. \tag{3}$$

Memory. Successfully executed designs enter a persistent store. Retrieval selects the k stored designs whose trajectories are closest to the current target under d, providing executable examples with related motion (App. U). Memory thus preserves concrete design experience, while critique interprets the current candidate's behaviour. Both change the context conditioning subsequent proposals.

3 LEARNING FROM REPRESENTATIONS OF SIMULATED MOTION

We examine how simulation feedback becomes useful experience for mechanism design. The analysis separates three questions: (i) whether critique supports refinement, (ii) what a symbolic description of the simulated trajectory adds to that critique, and (iii) how the resulting gains depend on the model and target motion.

Experimental setup. *MSynth* is a benchmark which covers six analytic planar curve families: circles, ellipses, lines, parabolas, Lemniscates of Bernoulli (∞) (LB), and NACA four-digit airfoils (Jacobs et al., 1933; Abbott & Von Doenhoff, 2012), as targeted geometrical outputs. These families provide explicit target equations and varied geometric structures, including open segments, closed curves, and multiple lobes (App. H). We sample five instances per class and supply $n = 4$ target points per instance. Geometric fidelity is measured by *Chamfer distance* after ICP alignment of the generated and target trajectories, while execution success and iteration count describe feasibility and refinement (App. G). Enum+GA combines topology enumeration with dimensional optimisation and provides a numerical-search comparison at several simulator-evaluation budgets (App. S.1). Solver settings, agent hyperparameters, and prompts are described in App. Q.

Comparisons. We ran a factorial experiment over **Model** $\in \{\text{Llama}, \text{Gemma}, \text{Qwen}\}$, **Shape**, retrieved design examples (#Ex $\in \{2, 3\}$), **Memory** $\in \{0, 2\}$, **Critique feedback** $\in \{\text{No}, \text{Yes}\}$, and **Symbolic feedback (SFB)** $\in \{\text{No}, \text{Yes}\}$. SFB was only ever run at $Mem = 2$, leading to six conditions per (model, shape, #Ex) (App. S).

3.1 CRITIQUE SUPPORTS SUSTAINED DESIGN REFINEMENT

Critique feedback is the dominant channel for every model family. Mechanism (semantic) interpretation success improves for all three models when critique is enabled, by up to roughly 88% in relative terms, and execution validity rises between 9 and 24 percentage points. Critique also raises the number of refinement rounds while lowering Chamfer distance. Even though it prolongs the search with extra rounds, it also buys quality. Figure 3 shows the corresponding gains. Every model improves, and the largest uplift is Qwen3:4B's, from 16.4% to 32.4%, a gain of 16.1 points against 15.8 for Llama and 9.0 for Gemma.

The symbolic layer adds on top of critique, and its clearest signature is temporal. At matched critique and $Mem = 2$, 55.2% of runs holding an intermediary are still lowering their best design after the first third of the run, against 44.4% of runs without one, and the median share of a run's total gain that arrives in that late phase is 15.6% against 5.4%. The intermediary extends the phase over which

Model	Condition	Value	Imp. %
vs. 4-bar baseline (GA at 2× budget: Pop: 6 / Gen: 20)			
Gemma 12B	No FB, No SR	1.319	−31.9% ↓
	FB	1.219	−21.9% ↓
	FB + SR	1.185	−18.5% ↓
Llama 70B	No FB, No SR	0.742	+25.8% ↑
	FB	⋆ **0.634**	**+36.6%** ↑
	FB + SR	0.659	+34.1% ↑
Qwen 4B	No FB, No SR	1.107	−10.7% ↓
	FB	0.845	+15.5% ↑
	FB + SR	0.688	+31.2% ↑
vs. 6-bar baseline (GA at 2× budget: Pop: 6 / Gen: 20)			
Gemma 12B	No FB, No SR	1.232	−23.2% ↓
	FB	1.154	−15.4% ↓
	FB + SR	1.112	−11.2% ↓
Llama 70B	No FB, No SR	0.712	+28.8% ↑
	FB	⋆ **0.634**	**+36.6%** ↑
	FB + SR	0.648	+35.2% ↑
Qwen 4B	No FB, No SR	1.063	−6.3% ↓
	FB	0.829	+17.1% ↑
	FB + SR	0.698	+30.2% ↑

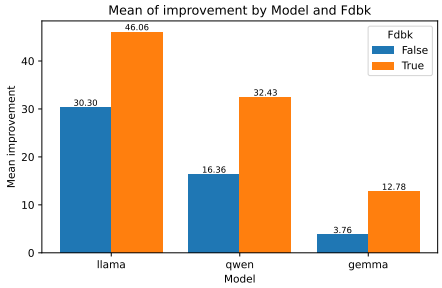

Figure 3: Average improvement in Chamfer distance with feedback (Fdbk) vs. without. Every model improves. The largest uplift is Qwen3:4B's (16.4% → 32.4%, +16.1 points); Llama3.3:70B gains 15.8 points and reaches the highest level (30.3% → 46.1%).

Table 1: Normalised Chamfer distance vs. an Enum+GA baseline given 120 evaluations, twice the agent's own budget of 60; matched-budget and 300× results are in App. S. Symbolic feedback lowers the ratio for Gemma and Qwen; Llama is the exception and does best under critique alone. **Value** = mean over shapes of $\frac{\text{LLM median}}{\text{GA median}}$; values < 1 (teal) mean the LLM beats the GA, > 1 red. ⋆ marks the best per block.

refinement keeps paying. In aggregate, the improvement with symbolic intermediary reaches a mean outcome of 140.00 against 148.50 without it, and restricting to $Mem=2$ preserves most of that gap. It is also uneven across models. In Table 1 the symbolic layer lowers normalised Chamfer distance for Gemma and Qwen against both baselines, most sharply for Qwen ($0.845 \rightarrow 0.688$ against the 4-bar baseline).

3.2 Symbolic feedback extends late-stage improvement

The symbolic intermediary describes an already simulated trajectory, i.e. it supplies no additional simulator observation. Its proposed benefit is to make geometric structure easier for the agent to interpret and connect to design changes. This motivates a prediction: the value of symbolic feedback should depend on how effectively the model can reason with the resulting expression.

The strongest improvement occurs on parabolas, the target family with the highest initially observed Chamfer error. With critique enabled and memory fixed ($Mem = 2$), symbolic feedback reduces the compact reasoning-trained Qwen 4B model's mean Chamfer distance by 10.3%, from 816.2 to 732.1. This mean is below the lowest parabola condition mean achieved by the roughly 17× larger Llama 70B model (815.2). Under the same matched comparison, Gemma improves by 2.1%, while Llama changes marginally. On simpler target classes, Llama achieves lower errors, showing that the performance ordering depends on the target motion (App. S).

The parabola improvement for Qwen remains substantial at both example counts: 7.3% with two examples and 13.3% with three. The corresponding reductions for Llama are +0.3% and −0.6%, indicating little change, while Gemma improves by 3.9% and 0.3%. Thus, the larger benefit for Qwen occurs under both example settings.

Across target families, the reported absolute reduction in Chamfer distance is largest on parabolas ($\Delta \approx 33.78$) and smallest on NACA airfoils (0.11). Their baseline errors differ by roughly three orders of magnitude, so these absolute reductions must be interpreted relative to their starting error

levels. The matched comparisons within the parabola family provide the clearer evidence that models benefit differently from the same symbolic feedback.

The central finding is therefore a model-dependent benefit, where parameter count alone does not explain which model gains most from the intermediary. This pattern is consistent with the hypothesis that analytical representations help when a linguistic/symbolic agent can use them to formulate effective revisions. Establishing the specific contribution of reasoning training is beyond the scope of this work.

3.3 Fitted expressions become objects of geometric critique

Paired transcripts show the analytical form substituting an explicit numerical check for a structural assertion. On the parabola, the canonical case, the critique holding an intermediary recovers the curve class from $y = (x-7)^2 + 3$ and applies the law of cosines ($40^2 = 30^2 + 20^2 - 2 \cdot 30 \cdot 20 \cos\theta \Rightarrow \theta \approx 104.5°$) to test whether a proposed joint is reachable, while the matched scalar-feedback critique observes that "a linear slider can only move along a straight path" and stops there. The substitution recurs across classes: on a NACA airfoil the intermediary-holding critique computes an explicit residual ($\sqrt{2.25+1} \approx 1.8$ against a target of 2.5) where the matched run offers "the four-bar linkage is a common choice" (Fig. 8; full transcripts in App. N).

Scoring all the critique entries against an eight-category taxonomy shows the shift is one of grounding and not of frequency (App. N.4). Engagement with discovered symbolic structure rises with the symbolic intermediary for every model, and grounded analytical reasoning rises most for the reasoning-trained Qwen 4B. Grounding thus requires both sides: an expression to reason over, and a model whose properties affords reasoning over that representation.

3.4 The benefit depends on the model and target motion

Parabolas, the target family with the highest observed error, provide a clear contrast between the models. With critique and $Mem = 2$, adding SFB lowers the mean of each run's best Chamfer distance for Qwen 4B from 816.2 to 732.1, a 10.3% reduction averaged over the two example counts. The reduction occurs with both two and three examples (7.3% and 13.3%, respectively). The corresponding mean reduction for Gemma is 2.1%, while Llama changes little (App. S).

This interaction makes the representational question concrete. The same type of analytical feedback can be productive for one model–target combination and offer little benefit for another. Qwen's improvement on parabolas coexists with Llama's stronger average performance on circles, ellipses, and lines (App. T). Parameter count alone therefore does not order performance in this setting. The empirical result is that the value of a representation depends jointly on the target motion and the model's use of the supplied expression.

3.5 Strong designs emerge under limited simulation budgets

Each agent run uses at most 60 simulator evaluations: three candidates per round over at most 20 rounds. We compare Enum+GA (a standard method used in mechanism design problems) at the same budget, then increase its allowance while keeping the agent budget fixed. Enum+GA searches separately within the 4-bar and 6-bar linkage classes, with each search receiving the full evaluation budget. The agent searches mechanisms expressible through the available API (§2.1), so the same agent results are compared against both baselines.

At the matched budget of 60 evaluations, all three models achieve lower mean normalised Chamfer error with critique and symbolic feedback than either baseline. Relative to the 4-bar baseline, the reductions are 19.6% for Gemma, 52.5% for Qwen, and 53.3% for Llama. Against the 6-bar baseline, the corresponding reductions are 4.5%, 39.9%, and 43.5% (App. S).

Table 1 doubles the Enum+GA budget to 120 evaluations while keeping the agent budget at 60. Llama and Qwen retain their advantage over the 4-bar baseline, with error reductions of 34.1% and 31.2%, respectively. For Qwen, the reported reduction is 15.5% with critique alone and 31.2% with critique plus symbolic feedback. Gemma's error is now 18.5% higher than the baseline.

The results demonstrate strong synthesis performance under limited simulation budgets. At $18{,}000$ evaluations (300 times the agent's budget) Enum+GA outperforms every model under both topology restrictions. The agent's advantage therefore lies in finding good designs with relatively few simulator evaluations.

4 Theory: Representations for Mechanism Revision

Symbolic feedback describes how observed motion differs from the target, while mechanism knowledge and experience help determine which edits could reduce that difference. Our theory analysis explains why these mechanisms are complementary and when their combination can save simulations. Appendix B provides assumptions and proofs, and Fig. 4 connects the results.

Preserve the geometry that matters for revision. Let e denote the current behaviour residual and v_a the predicted change induced by edit a. Under the local loss $\ell_e(a) = \frac{1}{2}\|e + v_a\|^2$,

$$\ell_e(a) - \ell_e(b) = \langle e, v_a - v_b \rangle + \tfrac{1}{2}\big(\|v_a\|^2 - \|v_b\|^2\big). \tag{4}$$

Only residual components along differences in edit effects influence this comparison. Theorem B.2 characterizes the smallest linear representation preserving these decisions and bounds approximation regret. This gives compression a concrete objective: *retain the distinctions needed to choose between edits*. An explicit fit $y = f(x)$ can lose such distinctions by merging branches, while low average fitting error need not preserve geometric coverage (App. B.5). The local coordinates are connected to stable correspondence and alignment branches of the experimental objective in Appendix B.1.

Connect the discrepancy to the effects of an edit. A useful description does not by itself identify a useful intervention: mechanisms with identical current behaviour can respond oppositely to the same edit (Proposition B.2). Theorem B.3 therefore relates revision transfer to similarity in both residuals and edit responses, together with local response curvature. This clarifies the complementary roles of expressions, code, and memory. Expressions expose geometric discrepancies; code and experience supply context for predicting how design changes affect them. Retrieved designs support revision insofar as their responses are relevant to the current mechanism.

Their complementarity determines the value of a simulation. In an r-dimensional calibration model with known edit responses, coefficient feedback identifies a correction from one informative simulation, whereas score-only feedback requires order-r queries for a constant-factor reduction in worst-case expected error (Theorem B.4). Unknown responses restore an order-r requirement even with coefficient feedback (Theorem B.4), while memory can help through the response directions it covers (Corollary B.4). These results connect representation to inference-time learning, i.e. structured observations become more useful when combined with knowledge of how to act on them. The experiments examine the practical value of this combination in the implemented synthesis loop (§3).

5 Discussion and Synthesis

Representations determine which inferences become accessible. The symbolic intermediary changes the computational role of a simulation result. Symbolic regression adds no new physical observation, but extracts analytical structure that the agent would otherwise have to recover from numerical coordinates, making abstract design inferences accessible. Our theory sharpens what "useful" means: a representation must preserve geometric discrepancies relevant to available edits, while the agent also needs knowledge of how those edits affect motion (§4). Similar output errors can require different corrections in different mechanisms. Geometric fidelity alone does not make feedback actionable. The intermediary and mechanism knowledge supply complementary resources. While one exposes the discrepancy, the other connects it to a possible intervention. This makes the relationship between representation and reasoning the central design question.

Learning consists in making simulation experience reusable. With fixed model parameters, the adaptive system is the complete agent loop. Its changing state contains observed outcomes, critiques, and executable designs that can inform subsequent decisions. The deeper role of memory is therefore

to make relationships between design choices and physical consequences reusable. Our transfer analysis suggests organising retrieval around compatible responses to edits, alongside trajectory similarity (App. B.4). The budget results give this perspective a practical meaning: interpreting and reusing existing observations can reduce the simulations needed to find a good design, while sufficiently extensive numerical search eventually overtakes the agent. The broader opportunity is to allocate more computation to extracting useful consequences from each simulation, increasing the value of experience without updating the underlying model.

6 Related Work

Learning through critique, tools, and memory. Language agents propose a program, run it, and revise from what the run reports (Yao et al., 2023b; Shinn et al., 2023; Madaan et al., 2023), with variants that broaden the search or add external calls (Wang et al., 2023; Yao et al., 2023a; Li et al., 2023; Park et al., 2023; Wang et al., 2024a; Shang et al., 2025; Yang et al., 2025; Wu et al., 2024; Hu et al., 2025; Xiang et al., 2024; Schick et al., 2023; Qin et al., 2024; Gao et al., 2023). We share Self-Refine's architecture (Madaan et al., 2023) and depart from the self-repair class (Shinn et al., 2023; Olausson et al., 2024) in one assumption: it verifies against an interpreter, whose verdict is a failing assertion, while ours verifies against a simulator returning real numbers, so the diagnostic half must be constructed (Dziri et al., 2023). Where the oracle stays a quantity, prior loops drop diagnosis and resample at scale (Romera-Paredes et al., 2024) or extrapolate from a score history (Yang et al., 2024). In contrast, we construct it from an external trace, which unaided self-correction lacks (Huang et al., 2024).

Representing simulation feedback. Agents use a simulator to score generated tasks (Wang et al., 2024b), ground plans (Rana et al., 2023; ichter et al., 2023), or close a driving loop (Fu et al., 2024; Mouselinos et al., 2024). The simulator decides, and its trace reaches the agent as coordinates or a score, never in the terms the agent reasons in. Closest is (Ma et al., 2023), which summarises a simulator's reward statistics into text, a move (Yuksekgonul et al., 2025) generalises. We differ in the target language: a statistics template reports a magnitude, while our fitted expression is written in the target's own notation, so the two compare as objects.

Symbolic regression as a feedback representation. Symbolic regression recovers closed-form expressions from data (Schmidt & Lipson, 2009; Udrescu & Tegmark, 2020; Petersen et al., 2021; Kamienny et al., 2022; Shojaee et al., 2023; Meidani et al., 2024; Cava et al., 2021; Cranmer, 2023), where the expression is the deliverable and its fidelity the metric. LLM-SR (Shojaee et al., 2025) is closest and runs the direction we invert: it uses a language model to produce equations, whereas we hand an expression to the model as input, discarded each iteration and never scored, so a poor fit costs guidance and leaves correctness intact.

Computational synthesis of planar mechanisms. Coupler-curve synthesis is solved analytically (Bai & Angeles, 2015), searched by metaheuristics under hand-set penalties (Singh et al., 2024), or learned by generative models (Lee et al., 2024; nob, 2022; Nobari et al., 2024; Nurizada et al., 2024). Each returns a design and no account of why it fails. We report the deficiency in the terms the agent proposed the design and, to our knowledge, ours is the first language agent in a closed design, evaluation, and revision loop for planar linkage synthesis.

7 Conclusion

Symbolic intermediaries turn simulated trajectories into compact analytical descriptions that language agents can compare with the target motion. This shared representation supports geometric critique and design revision, while candidate quality remains evaluated directly through simulation. With model parameters fixed, the agent adapts through accumulated feedback and memory. The experiments show that symbolic feedback supports sustained refinement, and that the complete loop outperforms Enum+GA under matched simulation budgets. These findings connect inference-time learning to representation: exposing the structure of numerical observations helps agents use simulation experience to reason about design changes and improve subsequent proposals.

REFERENCES

LINKS: A Dataset of a Hundred Million Planar Linkage Mechanisms for Data-Driven Kinematic Design, volume Volume 3A: 48th Design Automation Conference (DAC) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2022. doi: 10.1115/DETC2022-89798. URL https://doi.org/10.1115/DETC2022-89798.

Machine Learning Applications for the Synthesis of Planar Mechanisms — A Comprehensive Methodical Literature Review, volume Volume 7: 48th Mechanisms and Robotics Conference (MR) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2024. doi: 10.1115/DETC2024-143690. URL https://doi.org/10.1115/DETC2024-143690.

Ira H Abbott and Albert E Von Doenhoff. *Theory of wing sections: including a summary of airfoil data*. Courier Corporation, 2012.

Shaoping Bai and Jorge Angeles. Coupler-curve synthesis of four-bar linkages via a novel formulation. *Mechanism and Machine Theory*, 94:177–187, 2015. ISSN 0094-114X. doi: https://doi.org/10.1016/j.mechmachtheory.2015.08.010. URL https://www.sciencedirect.com/science/article/pii/S0094114X15001950.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

W. L. Cava, Patryk Orzechowski, Bogdan Burlacu, Fabr'icio Olivetti de Francca, M. Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. *Advances in neural information processing systems*, 2021:1 – 16, 2021. URL https://api.semanticscholar.org/CorpusID:236635250.

Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl, 2023. URL https://arxiv.org/abs/2305.01582. This work is licensed under the Apache-2.0 license https://github.com/MilesCranmer/PySR/blob/master/LICENSE.

Richardos Drakoulis. Iterative closest point. https://github.com/richardos/icp, 2023. Accessed: 2025-05-19.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Fkckkr3ya8.

Hugo Farajallah. pylinkage: Python linkage builder and optimizer. https://github.com/HugoFara/pylinkage, 2024. GitHub repository (v0.6.0, released Oct 2 2024; accessed 2025-05-08); This work is licensed under the MIT License https://github.com/HugoFara/pylinkage/blob/main/LICENSE.

Daocheng Fu, Wenjie Lei, Licheng Wen, Pinlong Cai, Song Mao, Min Dou, Botian Shi, and Yu Qiao. Limsim++: A closed-loop platform for deploying multimodal llms in autonomous driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1084–1090, 2024. doi: 10.1109/IV55156.2024.10588848.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of*

the 40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, pp. 10764–10799. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

V. García-Marina, I. Fernández de Bustos, G. Urkullu, and R. Ansola. Optimum dimensional synthesis of planar mechanisms with geometric constraints. *Meccanica*, 55(11):2135–2158, Nov 2020. ISSN 1572-9648. doi: 10.1007/s11012-020-01250-x. URL https://doi.org/10.1007/s11012-020-01250-x.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30583–30598. Curran Associates, Inc., 2022. doi: 10.52202/068431-2217. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c529dba08a146ea8d6cf715ae8930cbe-Paper-Conference.pdf.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. HiAgent: Hierarchical working memory management for solving long-horizon agent tasks with large language model. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32779–32798, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1575. URL https://aclanthology.org/2025.acl-long.1575/.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

brian ichter, Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, Dmitry Kalashnikov, Sergey Levine, Yao Lu, Carolina Parada, Kanishka Rao, Pierre Sermanet, Alexander T Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Mengyuan Yan, Noah Brown, Michael Ahn, Omar Cortes, Nicolas Sievers, Clayton Tan, Sichun Xu, Diego Reyes, Jarek Rettinghouse, Jornell Quiambao, Peter Pastor, Linda Luu, Kuang-Huei Lee, Yuheng Kuang, Sally Jesmonth, Nikhil J. Joshi, Kyle Jeffrey, Rosario Jauregui Ruano, Jasmine Hsu, Keerthana Gopalakrishnan, Byron David, Andy Zeng, and Chuyuan Kelly Fu. Do as i can, not as i say: Grounding language in robotic affordances. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

E. N. Jacobs, K. E. Ward, and R. M. Pinkerton. The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. Technical Report Report No. 460, National Advisory Committee for Aeronautics (NACA), 1933. URL https://ntrs.nasa.gov/citations/19930091108.

Pierre-alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and Francois Charton. End-to-end symbolic regression with transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 10269–10281. Curran Associates, Inc., 2022. doi: 10.52202/068431-0746. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/42eb37cdbefd7abae0835f4b67548c39-Paper-Conference.pdf.

Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987. ISSN 0364-0213. doi: https://doi.org/10.1016/S0364-0213(87)80026-5. URL https://www.sciencedirect.com/science/article/pii/S0364021387800265.

Sumin Lee, Jihoon Kim, and Namwoo Kang. Deep generative model-based synthesis framework of four-bar linkage mechanisms with target conditions. *Journal of Computational Design and Engineering*, 11(5):318–332, 10 2024. ISSN 2288-5048. doi: 10.1093/jcde/qwae084. URL https://doi.org/10.1093/jcde/qwae084.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 51991–52008. Curran Associates, Inc., 2023. doi: 10.52202/075280-2264. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a3621ee907def47c1b952ade25c67698-Paper-Conference.pdf.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *Second Agent Learning in Open-Endedness Workshop*, 2023. URL https://openreview.net/forum?id=RFUiBPyGYF.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Kazem Meidani, Parshin Shojaee, Chandan K. Reddy, and Amir Barati Farimani. SNIP: Bridging mathematical symbolic and numeric realms with unified pre-training. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KZSEgJGPxu.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

Tom M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill, New York, 1997. ISBN 0070428077.

Spyridon Mouselinos, Henryk Michalewski, and Mateusz Malinowski. Beyond lines and circles: Unveiling the geometric reasoning gap in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6192–6222, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.360. URL https://aclanthology.org/2024.findings-emnlp.360/.

Amin Heyrani Nobari, Akash Srivastava, Dan Gutfreund, Kai Xu, and Faez Ahmed. LInk: Learning joint representations of design and performance spaces through contrastive learning for mechanism synthesis. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a1MRjOL6WJ.

Anar Nurizada, Rohit Dhaipule, Zhijie Lyu, and Anurag Purwar. A dataset of 3m single-dof planar 4-, 6-, and 8-bar linkage mechanisms with open and closed coupler curves for machine learning-driven path synthesis. *Journal of Mechanical Design*, 147(4):041702, 11 2024. ISSN 1050-0472. doi: 10.1115/1.4067014. URL https://doi.org/10.1115/1.4067014.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=y0GJXRungR.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Brenden K Petersen, Mikel Landajuela Larma, Terrell N. Mundhenk, Claudio Prata Santiago, Soo Kyung Kim, and Joanne Taery Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=m5Qsh0kBQG.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/rana23a.html.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, Jan 2024. ISSN 1476-4687. doi: 10.1038/s41586-023-06924-6. URL https://doi.org/10.1038/s41586-023-06924-6.

Neider Nadid Romero, Alexandre Campos, Daniel Martins, and Rodrigo S. Vieira. A new approach for the optimal synthesis of four-bar path generator linkages. *SN Applied Sciences*, 1(11):1504, Oct 2019. ISSN 2523-3971. doi: 10.1007/s42452-019-1511-3. URL https://doi.org/10.1007/s42452-019-1511-3.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Yacmpz84TH.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009. doi: 10.1126/science.1165893. URL https://www.science.org/doi/abs/10.1126/science.1165893.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic LLM agent search in modular design space. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mPdmDYIQ7f.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan K. Reddy. Transformer-based planning for symbolic regression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0rVXQEeFEL.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. LLM-SR: Scientific equation discovery via programming with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=m2nmp8P5in.

Ramanpreet Singh, Vimal Kumar Pathak, Ashish Kumar Srivastava, Rakesh Kumar, and Abhishek Sharma. A new metaphor-less optimization algorithm for synthesis of mechanisms. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 18(4):2371–2391, May 2024. ISSN 1955-2505. doi: 10.1007/s12008-023-01502-6. URL https://doi.org/10.1007/s12008-023-01502-6.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, Jan 2024. ISSN 1476-4687. doi: 10.1038/s41586-023-06747-5. URL https://doi.org/10.1038/s41586-023-06747-5.

L. W. Tsai. Systematic enumeration of parallel manipulators. In C. R. Boër, L. Molinari-Tosatti, and K. S. Smith (eds.), *Parallel Kinematic Machines*, pp. 33–49, London, 1999. Springer London. ISBN 978-1-4471-0885-6.

Silviu-Marian Udrescu and Max Tegmark. AI feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Adrian Vasiliu and Bernard Yannou. Dimensional synthesis of planar mechanisms using neural networks: application to path generator linkages. *Mechanism and Machine Theory*, 36(2):299–310, 2001. ISSN 0094-114X. doi: https://doi.org/10.1016/S0094-114X(00)00037-9. URL https://www.sciencedirect.com/science/article/pii/S0094114X00000379.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, Mar 2024a. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL https://doi.org/10.1007/s11704-024-40231-1.

Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 4890–4924, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/143ea4a156ef64f32d4d905206cf32e1-Paper-Conference.pdf.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yue Wu, Yewen Fan, So Yeon Min, Shrimai Prabhumoye, Stephen Marcus McAleer, Ruslan Salakhutdinov, Yonatan Bisk, Yuanzhi Li, and Tom Mitchell. Agentkit: Structured LLM reasoning with dynamic graphs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=PKfAq8N4fK.

Yufei Xiang, Yiqun Shen, Yeqin Zhang, and Cam-Tu Nguyen. Retrospex: Language agent meets offline reinforcement learning critic. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4650–4666, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.268. URL https://aclanthology.org/2024.emnlp-main.268/.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bb4VGOWELI.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for LLM-based web agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oWdzUpOlkX.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023a. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, Mar 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08661-4. URL https://doi.org/10.1038/s41586-025-08661-4.

A Generative AI Use Disclosure

In this work, we used generative AI tools to draft parts of the manuscript, to edit the text for readability, and to create and edit software code, and to assist with drafting informal proof sketches under manual supervision. The authors reviewed and revised all such material.

B Theory of Representations for Mechanism Revision

The main question is what a simulated observation must preserve to help choose the next design. A description of motion identifies a discrepancy; knowledge of edit effects connects it to a possible correction. We first characterize the information needed for local edit comparisons, then establish conditions for transferring revisions. A pair of query results separates the cost of observing a discrepancy from the cost of identifying its response to edits. Finally, explicit counterexamples and recovery results describe which geometric evidence the current symbolic fitting family can preserve. Each result is accompanied by its interpretation for the feedback loop in §2.

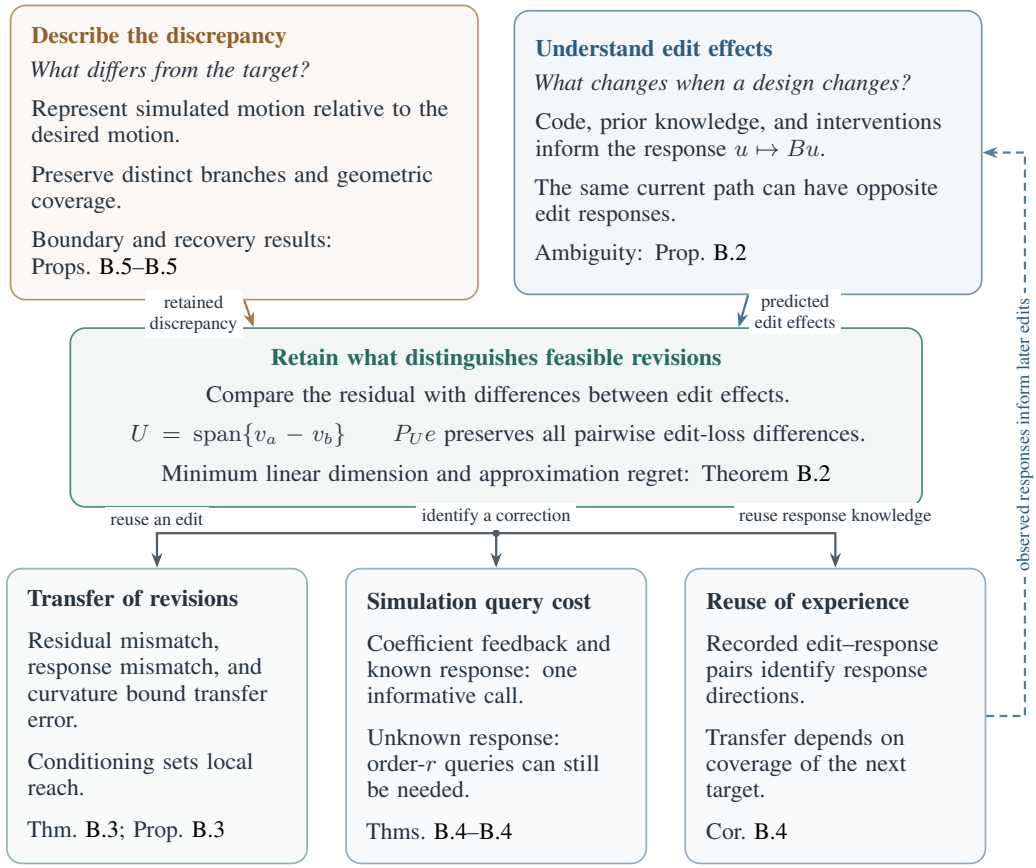

Figure 4: **How the theory connects.** The upper panels distinguish geometric evidence from knowledge of edit effects. Their meeting point is the information needed to compare feasible revisions. The lower panels organize the transfer, query, and memory results around this distinction. Solid arrows show conceptual information requirements; the dashed arrow shows reuse of observed responses in the memory model. Each result uses its stated assumptions: local behaviour maps for revision and transfer, affine feature models for query complexity, and explicit regression models for geometric fidelity.

Figure 4 can be read from top to bottom. First ask whether the intermediary retains the discrepancy that matters, then whether the effects of candidate edits are known. Together these determine which parts of the residual distinguish revisions. The remaining results examine how this information can be transferred, acquired through simulation, and retained for later tasks.

B.1 Behaviour coordinates and the experimental objective

Within a fixed mechanism topology and a valid assembly branch, write $F(\theta) \in \mathbb{R}^d$ for a differentiable vector of behaviour coordinates, $y_\star$ for its target, and $e = F(\theta) - y_\star$ for the current residual. A small parameter edit u changes the behaviour by $Bu + o(\|u\|)$, where $B = DF(\theta)$; the remainder is $O(\|u\|^2)$ when the Jacobian is locally Lipschitz. Coordinates may describe corresponding trajectory samples or coefficients in a fixed analytical family. The local loss is $\frac{1}{2}\|e\|^2$. The edit comparisons below concern an explicitly specified response model; the query results specify separate affine simulator families.

There is a direct local connection to the squared symmetric Chamfer score in Appendix G. Let $x_i(\theta)$, $i = 1, \ldots, m$, be generated points and q_j, $j = 1, \ldots, n$, target points. On a neighbourhood where

nearest-neighbour assignments $\alpha(i)$ and $\beta(j)$ remain fixed, stack the residuals

$$\mathcal{R}(\theta) = \operatorname{col}\left(\left\{ \frac{Rx_i(\theta) + t - q_{\alpha(i)}}{\sqrt{m}} \right\}_{i=1}^{m}, \left\{ \frac{Rx_{\beta(j)}(\theta) + t - q_j}{\sqrt{n}} \right\}_{j=1}^{n} \right). \tag{5}$$

Then $\|\mathcal{R}(\theta)\|^2$ is exactly that score at alignment (R, t); the local half-squared loss is one half of the score and has the same minimizers. Strict nearest-neighbour gaps ensure local assignment stability. Alignment can be fixed in this local analysis, or can follow a differentiable branch $(R(\theta), t(\theta))$, in which case its derivatives are included in $D\mathcal{R}$. The local results apply to this residual map; changes of correspondence, alignment branch, topology, or assembly require a new local model. A general coefficient representation needs its own connection to geometric error. We do not treat squared Chamfer as a metric or assume that ICP finds a global optimum.

The representation and query statements also distinguish the information supplied through a feedback channel from all information available to an agent. The implemented agent sees code, the target, and simulation feedback. Behaviour-only ambiguity concerns an abstraction that omits response information; the black-box query models stipulate which simulator quantities are unknown. Code and previous interactions can supply such quantities in practice. These results explain what information can support revision without imposing a particular reasoning procedure on the language model.

The mathematical perspective builds on decision-oriented prediction and value-preserving models (**??**). Our analysis makes the connection to mechanism revision explicit through edit-response spaces, transfer mismatch, and geometric fidelity. The affine query results specialize established questions about the information available through optimization oracles (**??**).

B.2 What information is needed to choose a revision?

A useful description need not preserve every detail of a trajectory. It must preserve the differences that distinguish the revisions under consideration. We first make this distinction exact for a finite set of candidate edits, then show why a description of current behaviour cannot supply their responses by itself. Throughout this appendix, vector norms are Euclidean and matrix norms are the induced operator norms.

Let $e \in \mathbb{R}^m$ be the current behaviour minus the target in the local coordinates of Section 4. For a finite action set $\mathcal{A}$, let $v_a \in \mathbb{R}^m$ denote the behaviour change associated with edit a. These can be exact displacements or predictions from a local response model. Define

$$Q_a(e) = \tfrac{1}{2}\|e + v_a\|^2, \qquad U = \operatorname{span}\{v_a - v_b : a, b \in \mathcal{A}\}, \qquad D = \max_{a,b\in\mathcal{A}} \|v_a - v_b\|. \tag{6}$$

The subspace U contains the behaviour differences that the available edits can produce relative to one another.

[Information sufficient for comparing revisions] All pairwise differences $Q_a(e) - Q_b(e)$ are determined by the orthogonal projection $P_U e$. A linear encoding $Re \in \mathbb{R}^k$ determines these differences for every $e \in \mathbb{R}^m$ if and only if $U \subseteq \operatorname{row}(R)$. Consequently, the minimum linear encoding dimension is $\dim U$.

Suppose $\widehat{e} \in U$ satisfies $\|\widehat{e} - P_U e\| \le \delta$, and a chosen edit $\widehat{a}$ is within $\xi \ge 0$ of minimizing $Q_a(\widehat{e})$. Then

$$Q_{\widehat{a}}(e) - \min_{a\in\mathcal{A}} Q_a(e) \le D\delta + \xi. \tag{7}$$

If the actual post-edit loss ℓ_a obeys $|\ell_a - Q_a(e)| \le \beta$ for every a, its excess over the best available edit is at most $D\delta + \xi + 2\beta$.

Expanding the squared norms gives

$$Q_a(e) - Q_b(e) = \langle P_U e, v_a - v_b \rangle + \tfrac{1}{2}(\|v_a\|^2 - \|v_b\|^2).$$

If $U \subseteq \operatorname{row}(R)$, each inner product is a linear function of Re. Conversely, if this inclusion fails, some $w \in \ker R$ and some pair a, b satisfy $\langle w, v_a - v_b \rangle \neq 0$. The residuals $e + tw$ have the same encoding for every t, while their pairwise loss difference varies with t. Thus R cannot determine that difference. The dimension statement follows by taking coordinates in an orthonormal basis of U.

Let a^* minimize $Q_a(e)$. Approximate optimality at $\widehat{e}$ and the same expansion yield

$$Q_{\widehat{a}}(e) - Q_{a^*}(e) \le \xi + \langle P_U e - \widehat{e}, v_{\widehat{a}} - v_{a^*} \rangle \le \xi + D\delta.$$

Adding the loss approximation error once for the selected edit and once for a true optimal edit gives the final bound.

Interpretation. The relevant compression dimension is set jointly by the task and the available revisions. Geometry orthogonal to U changes all model losses by the same amount and therefore does not help select an edit. Equation 7 separates three quantities: error in the retained discrepancy (δ), error in using it to choose an edit (ξ), and error in predicting what edits do (β). For example, if an actual displacement differs from v_a by at most κ and $\|e + v_a\| \le M$ for all a, then $\beta = M\kappa + \kappa^2/2$ follows by expanding its squared loss. The theorem characterizes a representation requirement; it does not identify expression-tree length with linear encoding dimension.

The response vectors in this result are substantive information. Even a perfect equation for the current trajectory does not determine them.

[Current behaviour does not identify edit response] Fix $a > 0$. Consider two scalar behaviour maps $F_s(u) = z_0 + su$, $s \in \{-1, +1\}$, with target $z_0 - a$, edits $u \in [-a, a]$, and loss $\ell_s(u) = \frac{1}{2}|F_s(u) - (z_0 - a)|^2$. A policy observing only their common current behaviour at $u = 0$, its target, and its loss cannot guarantee a strict improvement in expected loss for both maps. Indeed, its worst-case expected excess loss over the best edit is at least $a^2/2$, and this lower bound is attained by making no edit. One additional observation at any nonzero edit identifies s.

The policy has the same distribution of edits V for both maps. The average of their expected losses is

$$\tfrac{1}{4}\mathbb{E}\big[(a+V)^2 + (a-V)^2\big] = \tfrac{1}{2}a^2 + \tfrac{1}{2}\mathbb{E}[V^2] \ge \tfrac{1}{2}a^2.$$

Each map has zero optimal loss at $u = -sa$. Hence at least one expected excess loss is $a^2/2$ or greater; $V = 0$ achieves equality for both. A nonzero probe identifies s from $(F_s(u) - z_0)/u$.

Interpretation. The two maps have a simple mechanical realization. Mount an adjustable pen extension of length $a_0 + u$ rigidly on a rotating arm, with $0 < a < a_0 < R$. An outward mounting on an arm of length $R - a_0$ traces a circle of radius $R + u$; an inward mounting on an arm of length $R + a_0$ traces radius $R - u$. Both start with the same circle, but increasing the same positive physical extension has opposite effects. The proposition concerns behaviour-only policies: the mounting information in code distinguishes the designs. Our agent receives code and memory alongside the fitted expression, so these representations can supply complementary information about what happened and how it can change.

B.3 When can a revision transfer between mechanisms?

Retrieving a design with similar motion provides a plausible starting point. Transferring its revisions additionally requires compatible responses. The next result quantifies this requirement and separates response mismatch from nonlinear effects over a finite edit radius.

For $i \in \{A, B\}$, let F_i be a differentiable behaviour map near a design θ_i, with target y_i, residual $e_i = F_i(\theta_i) - y_i$, and Jacobian $B_i = DF_i(\theta_i)$. Let P be a linear map from edits of mechanism A to edits of B. Assume their Jacobians are Lipschitz with constants L_A, L_B on convex neighbourhoods containing all segments to the edits considered. Write

$$\ell_A(u) = \tfrac{1}{2}\|F_A(\theta_A + u) - y_A\|^2, \qquad \ell_B(Pu) = \tfrac{1}{2}\|F_B(\theta_B + Pu) - y_B\|^2.$$

[Transfer under compatible responses] Let $\mathcal{V}$ be a finite set of source edits with $\|u\| \le r$ and feasible recipient edits Pu. Define

$$\begin{aligned} d_r &= \|e_B - e_A\| + r\|B_B P - B_A\| + \tfrac{1}{2}(L_A + L_B\|P\|^2)r^2, \\ M_r &= \|e_A\| + \|B_A\|r + \tfrac{1}{2}L_A r^2, \qquad \Delta_r = M_r d_r + \tfrac{1}{2}d_r^2. \end{aligned} \tag{8}$$

Then $|\ell_B(Pu) - \ell_A(u)| \le \Delta_r$ for every $u \in \mathcal{V}$. If u_A is within ξ of the best source edit in $\mathcal{V}$, its transferred excess loss satisfies

$$\ell_B(Pu_A) - \min_{u \in \mathcal{V}} \ell_B(Pu) \le \xi + 2\Delta_r. \tag{9}$$

If also $e_A = e_B$, a source improvement $g = \ell_A(0) - \ell_A(u_A)$ transfers to an improvement of at least $g - \Delta_r$ on B.

The integral remainder formula and Lipschitz Jacobians give

$$F_A(\theta_A + u) - y_A = e_A + B_A u + R_A(u), \qquad \|R_A(u)\| \le L_A\|u\|^2/2,$$

and the corresponding recipient remainder is at most $L_B\|Pu\|^2/2$. Subtraction bounds the residual difference by d_r, while the source residual norm is at most M_r. For any vectors x, h with $\|x\| \le M_r$ and $\|h\| \le d_r$,

$$\left|\tfrac{1}{2}\|x+h\|^2 - \tfrac{1}{2}\|x\|^2\right| \le M_r d_r + \tfrac{1}{2}d_r^2.$$

This proves the uniform loss bound. If u_B minimizes the recipient loss over the mapped edits, then

$$\ell_B(Pu_A) \le \ell_A(u_A) + \Delta_r \le \ell_A(u_B) + \xi + \Delta_r \le \ell_B(Pu_B) + \xi + 2\Delta_r.$$

Finally, equal initial residuals imply equal initial losses, so only the post-edit discrepancy is subtracted from the source gain.

Interpretation. With matched initial residuals, response mismatch contributes at order r, whereas curvature contributes at order r^2. Thus reducing the edit size controls nonlinear error but does not establish compatibility of the leading response. Similar current curves alone place no bound on $\|B_B P - B_A\|$. Code, paired design–outcome examples, or deliberate perturbations can provide evidence for this missing relation. The conclusion compares the transferred edit with the same mapped candidate set, rather than all possible recipient designs.

[A radius for local target transfer] Let $F : \Omega \to \mathbb{R}^m$ be continuously differentiable on an open valid-design domain $\Omega \subseteq \mathbb{R}^p$. Suppose $B = DF(\theta)$ has full row rank and smallest singular value $\sigma > 0$, and its Jacobian is L-Lipschitz, $L > 0$, on the closed ball of radius $\sigma/(2L)$ about θ, contained in Ω. Every target y satisfying

$$\varepsilon = \|F(\theta) - y\| \le \frac{\sigma^2}{4L} \tag{10}$$

is attained by an edit $u \in \mathrm{range}(B^\top)$ with $\|u\| \le 2\varepsilon/\sigma$.

The case $\varepsilon = 0$ uses $u = 0$. Otherwise set $\rho = 2\varepsilon/\sigma \le \sigma/(2L)$, $e = F(\theta) - y$, and $R(u) = F(\theta + u) - F(\theta) - Bu$. On the radius-$\rho$ ball in $\mathrm{range}(B^\top)$, consider

$$T(u) = -B^\dagger e - B^\dagger R(u).$$

Here $\|B^\dagger\| = 1/\sigma$, so $\|T(u)\| \le \varepsilon/\sigma + L\rho^2/(2\sigma) \le 3\rho/4$. Moreover, $DR(u) = DF(\theta + u) - B$ implies that T is Lipschitz with constant at most $L\rho/\sigma \le 1/2$. It therefore has a fixed point in this closed ball. Since $BB^\dagger = I$, its fixed-point equation gives $Bu = -e - R(u)$, hence $F(\theta + u) = y$.

Interpretation. Within task-relevant behaviour coordinates, well-conditioned responses permit reuse for nearby targets. Matching these coordinates need not match an entire sampled trajectory. The guaranteed neighbourhood shrinks quadratically with the weakest response singular value, and this order occurs in a physical example. A pen extension of length a mounted at a fixed obtuse angle ϕ on a rotating rigid arm of length b traces radius

$$r(a) = \sqrt{(a - a_*)^2 + c^2}, \qquad a_* = -b\cos\phi > 0, \quad c = b\sin\phi > 0.$$

At $a = a_* + t$, $t > 0$, the response is $\sigma = t/\sqrt{t^2 + c^2}$ and its derivative is Lipschitz with $L = 1/c$. The maximum attainable radius reduction is

$$\sqrt{t^2 + c^2} - c = \left(\tfrac{1}{2} + o(1)\right)\frac{\sigma^2}{L} \quad \text{as } t \downarrow 0.$$

At $a = a_*$ the response loses rank and every smaller radius is unreachable. This illustrative construction explains the local boundary; the proof supplies an existence result rather than a description of the language agent's revision algorithm.

B.4 When structural feedback saves simulations

A scalar score says how large a discrepancy is; an expression can reveal its components. This distinction can change the number of simulations needed to diagnose a design. Turning that diagnosis into a correction, however, also requires knowing how design edits affect those components. The following paired results separate these two sources of difficulty. Their connection to dimension-dependent value-oracle optimisation is deliberate (**??**): the question here is when a representation exposes information that an available edit model can use.

We use a canonical affine feature model. Let $\phi_1, \dots, \phi_r$ be known orthonormal functions in $L^2(\mu)$, so an expression $g(t) = \sum_j c_j \phi_j(t)$ has squared error $\|g\|^2_{L^2(\mu)} = \|c\|_2^2$ against the zero target. A simulation produces this expression's trajectory; the feedback channel delivers either its squared error or its coefficient vector. The analysis concerns this squared feature loss and a simulator accessed through queries. It isolates the feedback interface; the paper's complete linkage search uses the geometric objective in App. G, and its language agent supplies the revision policy.

[A query advantage from accessible residual structure] Fix $r \geq 1$. A design $a \in \mathbb{R}^r$ produces

$$g_{a,b}(t) = \sum_{j=1}^{r} (a_j - b_j)\phi_j(t), \qquad \mathcal{L}_b(a) = \|a - b\|_2^2,$$

where the target is the known zero function and $b \in S^{r-1}$ is an unknown simulator calibration. Every adaptive, possibly randomized policy receiving only $\mathcal{L}_b(a)$ at each query and returning $\widehat{a}_m$ after $m \leq r$ queries satisfies

$$\sup_{b \in S^{r-1}} \mathbb{E}\, \mathcal{L}_b(\widehat{a}_m) \;\geq\; 1 - \frac{m}{r}. \tag{11}$$

In contrast, receiving the coefficient vector $c = a - b$ at one query identifies a zero-loss design $a - c$. If the received vector $\widehat{c}$ satisfies $\|\widehat{c} - c\|_2 \leq \eta$ and the implemented correction has additional error e with $\|e\|_2 \leq \delta$, then

$$\mathcal{L}_b(a - \widehat{c} + e) \leq (\eta + \delta)^2. \tag{12}$$

The exact correction requires one informative simulation and one further simulation if its performance must be verified.

Put the uniform prior on $b \in S^{r-1}$, independently of the policy's random seed. A score reveals exactly the linear measurement

$$\langle a, b \rangle = \tfrac{1}{2}\big(\|a\|_2^2 + 1 - \mathcal{L}_b(a)\big).$$

Condition on the seed. Orthogonalising each query against previous query directions shows that it reveals at most one fresh projection of b. We may grant the policy additional measurements to obtain m independent directions: doing so can only reduce its minimum achievable Bayes risk. Conditional on the history $\mathcal{H}_j$ of j such measurements, write $b = p_j + z_j$, where p_j is its known projection onto their span. Rotational invariance of the spherical prior implies that z_j is uniform on the sphere of radius ρ_j in the remaining $(r-j)$-dimensional space. This remains true for adaptive queries: the next orthogonalised direction is fixed conditional on $\mathcal{H}_j$, while the unexplored component is rotationally invariant.

The squared projection onto any such unit direction has conditional expectation $\rho_j^2/(r-j)$. Consequently,

$$\mathbb{E}[\rho_{j+1}^2 \mid \mathcal{H}_j] = \rho_j^2 \frac{r-j-1}{r-j}, \qquad \mathbb{E}\rho_m^2 = \frac{r-m}{r},$$

starting from $\rho_0^2 = 1$. The posterior mean p_m minimises conditional squared error, whose minimum is ρ_m^2. Thus the Bayes risk of any returned design is at least $(r-m)/r$. Averaging over the independent seed preserves this bound, and worst-case risk is at least prior-average risk, proving Eq. 11. Exact coefficient feedback gives $a - c = b$. With approximate feedback, the corrected residual is $c - \widehat{c} + e$; the triangle inequality gives Eq. 12.

The gain arises because one simulation exposes several independently correctable discrepancies. It is a comparison of delivered feedback channels: the coefficients are already determined by the simulated trajectory. The result also makes the required prior knowledge explicit—the policy knows

that changing a_j changes the corresponding coefficient by the same amount. The next result removes that knowledge while retaining complete coefficient feedback.

[Complete behavioural feedback does not identify edit responses] Let a design $a \in \mathbb{R}^r$ produce coefficient vector Ra, where $R \in O(r)$ is unknown, and let the known target have coefficient vector $u \in S^{r-1}$. Every query returns the full residual $Ra - u$. For every adaptive, possibly randomized policy returning $\widehat{a}_m$ after $m \le r$ queries,

$$\sup_{R \in O(r)} \mathbb{E}\|R\widehat{a}_m - u\|_2^2 \;\ge\; 1 - \frac{m}{r}. \tag{13}$$

Querying an input basis identifies R in r simulations and permits the exact correction $R^\top u$.

Give R the Haar probability distribution on $O(r)$, independently of the policy's seed, and condition on that seed. Since u is known, residual feedback reveals Ra. After j independent query directions have been orthonormalised to $v_1, \dots, v_j$, their images $w_i = Rv_i$ are known and orthonormal. Set $V_j = \operatorname{span}\{v_i\}_{i=1}^j$ and $W_j = RV_j$. Conditional on this history, the restriction of R from $V_j^\perp$ to $W_j^\perp$ remains a uniformly random orthogonal isometry. Adaptive choices do not change this conditional invariance, because each next query is a function of the observed history.

The optimal design $a^* = R^\top u$ therefore has known component

$$p_j = \sum_{i=1}^{j} v_i \langle w_i, u \rangle$$

and a conditionally uniform, mean-zero residual in $V_j^\perp$ of radius $\rho_j = \|P_{W_j^\perp} u\|_2$. Its conditional Bayes squared error is ρ_j^2. A new independent query has, after orthogonalisation, an image uniform on the unit sphere in $W_j^\perp$. Hence it reveals a component of u with expected squared magnitude $\rho_j^2/(r-j)$, giving the same recurrence and $\mathbb{E}\rho_m^2 = (r-m)/r$. As before, dependent queries can be supplemented with fresh ones without increasing minimum Bayes risk. Finally,

$$\|R\widehat{a}_m - u\|_2^2 = \|\widehat{a}_m - R^\top u\|_2^2,$$

so prior-average risk, and therefore worst-case risk, satisfies Eq. 13. This includes $j = r-1$: under $O(r)$ the remaining one-dimensional isometry still has two equiprobable signs. The upper bound follows by reading the columns of R from its responses to an input basis.

Even a complete description of the current motion leaves open how to change it when the response map is unknown. This explains the complementary roles of the intermediary and the agent's mechanism knowledge: the former exposes a discrepancy, while the latter can supply a useful relation between that discrepancy and an edit. Simulation experience can also establish that relation, leading to the following transfer statement.

[Transfer is governed by response coverage] In the orthogonal response model of Theorem B.4, suppose memory records exact response pairs spanning an input subspace V. Choose an orthonormal basis $v_1, \dots, v_k$ of V and recover their images $w_i = Rv_i$ from those pairs. Conditional on this memory, assume that the unobserved restriction from $V^\perp$ to $(RV)^\perp$ has the Haar distribution. For any fixed new target u, the minimum conditional Bayes risk is

$$\inf_{\widehat{a}} \mathbb{E}[\|R\widehat{a} - u\|_2^2 \mid \text{memory}, u] = \|P_{(RV)^\perp} u\|_2^2, \tag{14}$$

attained by $\widehat{a} = \sum_{i=1}^k v_i \langle w_i, u \rangle$. If new targets are independent of the remaining response uncertainty conditional on memory and have conditional second moment $\Sigma = \mathbb{E}[uu^\top \mid \text{memory}]$, their average risk is

$$\operatorname{tr}\big[(I - P_{RV})\Sigma\big].$$

Conditional on memory and the fixed target, the known component of $R^\top u$ is the stated estimate. Its remaining component has mean zero and squared norm $\|P_{(RV)^\perp} u\|_2^2$ under the conditional Haar model. The posterior-mean characterisation of squared-error risk proves Eq. 14. Averaging this quadratic form over independent new targets gives the trace expression.

The transferable resource in this model is coverage of response directions relevant to future targets. Repeated examples confined to the same response subspace leave the same uncertainty; examples that expand it can support new corrections. This provides a criterion for studying what design memory should retain beyond similarity of the recorded trajectories.

B.5 When does a symbolic description preserve trajectory geometry?

A useful description must retain the geometric distinctions relevant to the target. Two separate issues arise: whether the expression class can represent the path, and whether the fitting criterion gives sufficient weight to every relevant part of it. The following results isolate these issues for the explicit regression used in Eq. 2. They concern the information retained by the intermediary; the agent also receives the program and simulation feedback, and candidate scoring continues to use the original trace.

Averaging branches can remove the shape. When a path passes above and below the same horizontal coordinate, a single prediction can lie between its branches. More data can make this average more accurate without recovering either branch.

[Branch collapse under explicit regression] Suppose that, conditional on $X = x$, a trajectory observation satisfies $Y = m(x) + h(x)$ or $Y = m(x) - h(x)$ with equal probability, where $h \geq 0$ and the variables have finite second moments. For every square-integrable predictor f,

$$\mathbb{E}[(Y - f(X))^2] = \mathbb{E}[h(X)^2] + \mathbb{E}[(m(X) - f(X))^2]. \tag{15}$$

Thus the least-squares predictor is m, uniquely up to X-null sets, and does not identify h. In particular, uniformly phase-sampled ellipses $C_b = \{(\cos t, b \sin t) : 0 \leq t \leq 2\pi\}$, $b \geq 1$, all have predictor $f = 0$ and horizontal domain $[-1, 1]$, whereas

$$\inf_{R \in \mathrm{SE}(2)} d_H(RC_b, C_{b'}) \geq |b - b'|, \tag{16}$$

where d_H is Hausdorff distance and $\mathrm{SE}(2)$ denotes planar rigid motions.

Write $Y = m(X) + Z$, with $\mathbb{E}[Z \mid X] = 0$ and $\mathbb{E}[Z^2 \mid X] = h(X)^2$. Expanding the square and taking conditional expectations proves Eq. 15. For the ellipses, the two ordinates at an interior abscissa are $\pm b\sqrt{1 - x^2}$ with equal conditional probability, so their conditional mean is zero. Their diameters are $2b$ and $2b'$. For compact sets A, B, matching points within Hausdorff distance gives $|\operatorname{diam}(A) - \operatorname{diam}(B)| \leq 2d_H(A, B)$. Rigid motions preserve diameter, proving Eq. 16.

This is a representation boundary, rather than a search-compute boundary: arbitrarily different ellipses have the same optimal expression, even when its input interval is retained. The ellipse conclusion also holds with the expression-size penalty when the zero constant has minimum complexity. The missing information has a simple constructive description.

[Recovering balanced branches and its stability] For the balanced two-branch model, let $v(x) = \mathrm{Var}(Y \mid X = x)$. Its branches are exactly the zero set, over their horizontal domain I, of

$$(y - m(x))^2 - v(x) = 0. \tag{17}$$

Suppose I is compact and $m, v, \hat{m}, \hat{v}$ are continuous, with $v, \hat{v} \geq 0$, $\|m - \hat{m}\|_\infty \leq \eta_m$ and $\|v - \hat{v}\|_\infty \leq \eta_v$. The corresponding two-branch sets $C, \widehat{C}$ satisfy

$$d_H(C, \widehat{C}) \leq \eta_m + \sqrt{\eta_v}. \tag{18}$$

If additionally $v \geq \rho^2 > 0$ on I, the bound improves to $d_H(C, \widehat{C}) \leq \eta_m + \eta_v/\rho$.

The conditional variance is h^2, so Eq. 17 recovers $m \pm h$. Pair points at the same x and with the same branch sign. Their distance is at most $\eta_m + |\sqrt{v(x)} - \sqrt{\hat{v}(x)}|$ in both directions. For nonnegative a, b, $|\sqrt{a} - \sqrt{b}| \leq \sqrt{|a - b|}$. If $a \geq \rho^2$, the identity $|\sqrt{a} - \sqrt{b}| = |a - b|/(\sqrt{a} + \sqrt{b})$ instead gives the bound $|a - b|/\rho$.

The square-root dependence is sharp without separation: $m = \hat{m} = 0$, $v = 0$, and $\hat{v} = \eta_v$ give Hausdorff distance $\sqrt{\eta_v}$. This instability concerns reconstruction through the chosen Cartesian branches; it need not be an instability of the physical curve under an alternative parametrization. Equation 17 is a constructive alternative, not an additional component of the evaluated pipeline. Equal conditional branch weights are essential to this particular mean–variance reconstruction.

Average accuracy and geometric coverage are different. Even a small regression error can conceal a part of the path that matters to the target. To state this precisely, treat traces as *sample lists*:

repeated locations retain their multiplicity in the averages. Write

$$d_{\mathrm{Ch}}(P,T) = \frac{1}{|P|}\sum_{p\in P}\min_{q\in T}\|p-q\|^2 + \frac{1}{|T|}\sum_{q\in T}\min_{p\in P}\|q-p\|^2,$$

the squared symmetric Chamfer score used in App. G.

[Vanishing fit error need not preserve the task score] Let $T = ((0,0),(0,1),(0,2),(0,3))$. For $n \geq 4$, let P_n contain $n-3$ copies of $(0,0)$ and one copy of each remaining point in T. An optimal explicit least-squares fit has $f_n(0) = 6/n$, with mean squared error

$$E_n = \frac{14}{n} - \frac{36}{n^2} \longrightarrow 0.$$

Nevertheless, $d_{\mathrm{Ch}}(P_n, T) = 0$, while its fitted sample list $\widehat{P}_n = ((x_i, f_n(x_i)))_{i=1}^n$ satisfies

$$d_{\mathrm{Ch}}(R\widehat{P}_n, T) \geq \frac{5}{4} \qquad \text{for every } R \in \mathrm{SE}(2).$$

Every input is zero, so minimizing squared error reduces to estimating the sample mean of the ordinates, $6/n$. Their mean square is $14/n$, giving the stated error. The original and target lists have identical supports, so both nearest-neighbour terms vanish. The fitted list has a single location; after any rigid motion call it z. Its target-to-trace term is

$$\frac{1}{4}\sum_{j=0}^{3}\|(0,j)-z\|^2 = \frac{5}{4} + \|z-(0,3/2)\|^2 \geq \frac{5}{4}.$$

The other term is nonnegative.

Thus no bound tending to zero with average fitting error alone can guarantee preservation of this geometric score, even with four target samples and rigid alignment. No assumption about global optimality of ICP is involved. The example isolates an averaging effect: rarely visited locations contribute little to regression error but remain part of the target's support. With constants of minimum expression complexity, the constant fit is also optimal under the nonnegative size penalty in Eq. 2.

Coverage supplies a positive guarantee. The preceding gap closes when an error peak cannot be hidden in a negligibly sampled interval. The next result makes that requirement explicit and separates it from the fitting error itself.

[Coverage converts mean error into geometric fidelity] Let $\gamma, \hat{\gamma} : [0,1] \to \mathbb{R}^2$ be continuous, and suppose $e(t) = \|\gamma(t) - \hat{\gamma}(t)\|$ is K-Lipschitz, $K \geq 0$. Let the sampling distribution have a density $p(t) \geq c > 0$, and set $E = \int_0^1 e(t)^2 p(t)\,dt$. Then

$$\|e\|_\infty \leq \delta := \max\left\{2\sqrt{E/c},\ (8KE/c)^{1/3}\right\}. \tag{19}$$

Consequently $d_H(\gamma([0,1]), \hat{\gamma}([0,1])) \leq \delta$. For paired sample lists $P = (\gamma(t_i))_{i=1}^n$ and $Q = (\hat{\gamma}(t_i))_{i=1}^n$ and a finite nonempty target T, if $P \cup Q \cup T$ lies in a set of diameter at most D, then, in this common coordinate frame,

$$|d_{\mathrm{Ch}}(P,T) - d_{\mathrm{Ch}}(Q,T)| \leq 4D\delta. \tag{20}$$

Let $A = \|e\|_\infty$, attained at t_0. The case $A = 0$ is immediate. If $K = 0$, $E = A^2$, and Eq. 19 follows since $c \leq 1$. Otherwise put $r = \min\{1, A/(2K)\}$. The intersection of $[t_0 - r, t_0 + r]$ with $[0,1]$ has length at least r, and $e \geq A/2$ throughout it. Hence $E \geq crA^2/4$. If $r = A/(2K)$, this gives $A \leq (8KE/c)^{1/3}$; if $r = 1$, it gives $A \leq 2\sqrt{E/c}$. Pairing equal parameter values proves the Hausdorff bound.

For the score bound, paired samples are within δ. Their distances to T therefore differ by at most δ. Conversely, the distance of each target point to the support of P differs from its distance to the support of Q by at most δ. All these distances are at most D, so each squared-distance difference is at most $2D\delta$. Averaging the two directed terms proves Eq. 20.

This guarantee requires population error, sampling coverage, and residual regularity; finite empirical fit error alone does not establish them. The score bound concerns a common frame, including a common fixed rigid transformation when the distance bound holds, rather than separately fitted ICP transformations. Together, the results identify what an informative intermediary must preserve: distinct branches and the parts of a trajectory needed to cover the target.

C LIMITATIONS

Planar targets. We evaluate on 2D curves. This is the most favourable case for the construction and not a neutral one: a planar trace admits a short readable expression, which is condition (iv) of §2.1, and 3D paths and time-varying fields are where that condition first comes under strain. The reported effect should therefore be read as an upper bound on what transfers, and we treat 3D as an open extension.

D ETHICAL STATEMENT

This work explores AI-driven methods for automating engineering design tasks. While the results are promising, the proposed method is exploratory and should not replace expert engineering judgement.

E EXAMPLE

Figure 5 shows a materialisation of the neuro-symbolic synthesis method for planar mechanism design. **(1)** Designer Agent generate hypotheses, **(2)** simulator simulates and Critique Agent critiques and prescribe revisions, and **(3)** Designer Agent performs the revision then repeat.

1. Designer Agent. This agent takes a target specification, either an analytic curve or sampled points, together with high-level constraints such as naming conventions, plus memory of previously validated designs, and produces executable mechanism hypotheses. It translates abstract requirements into concrete structural proposals by exploring topology and parameter spaces while enforcing hard constraints. The output is an executable representation of a mechanism (topology and parameter values) expressed in the system's canonical design language. Emphasis is on generating a diverse set of simulation-ready candidates, biasing toward promising motifs learned from memory and avoiding known dead-ends.

2. Simulation and Critique. Each candidate is validated in the simulation to confirm physical executability and to convert numeric behaviour into compact, interpretable summaries. The simulation produces time-series traces of the end-effector in R^2, execution diagnostics (for example, success or failure), and a symbolic surrogate, a compact analytic expression that approximates the numeric trace. This stage gates validity by rejecting non-executable or unstable designs, and it exposes algebraic and functional motifs in the motion through the symbolic summary.

3. Revision. The Revision step consumes the critique entries into structured, actionable modifications. It produces a new revised synthesises hypothesis that implements those changes. The goal is to recommend minimal, high-impact edits that improve fit and robustness while reducing unnecessary complexity, using memory to prevent repeating known failures and to suggest effective design motifs.

1. The Designer proposes one or more simulation-ready mechanism hypotheses.
2. The Simulator simulates each candidate and produces numeric traces, diagnostics, and symbolic surrogates.
3. The Critic analyses.
4. The Designer ingests the directives and memory items, revises or generates new hypotheses, and the cycle repeats.

F METHOD ALGORITHM

In this section, we provide a detailed description of our dual-agent iterative design loop for synthesising planar mechanisms (see Algorithm 1). Starting with an initial set of parameters, the $\mathbb{D}a$ proposes a candidate mechanism, which is then simulated on $\mathcal{S}im$. Successful simulations trigger a distance-metric evaluation against the target trajectory $\mathcal{T}$, and SR feedback. The validated design is then stored in memory. The $\mathbb{C}a$ then assesses performance and provides feedback on how to refine the design, prompting the $\mathbb{D}a$ to generate updated parameters for a second simulation and re-evaluation.

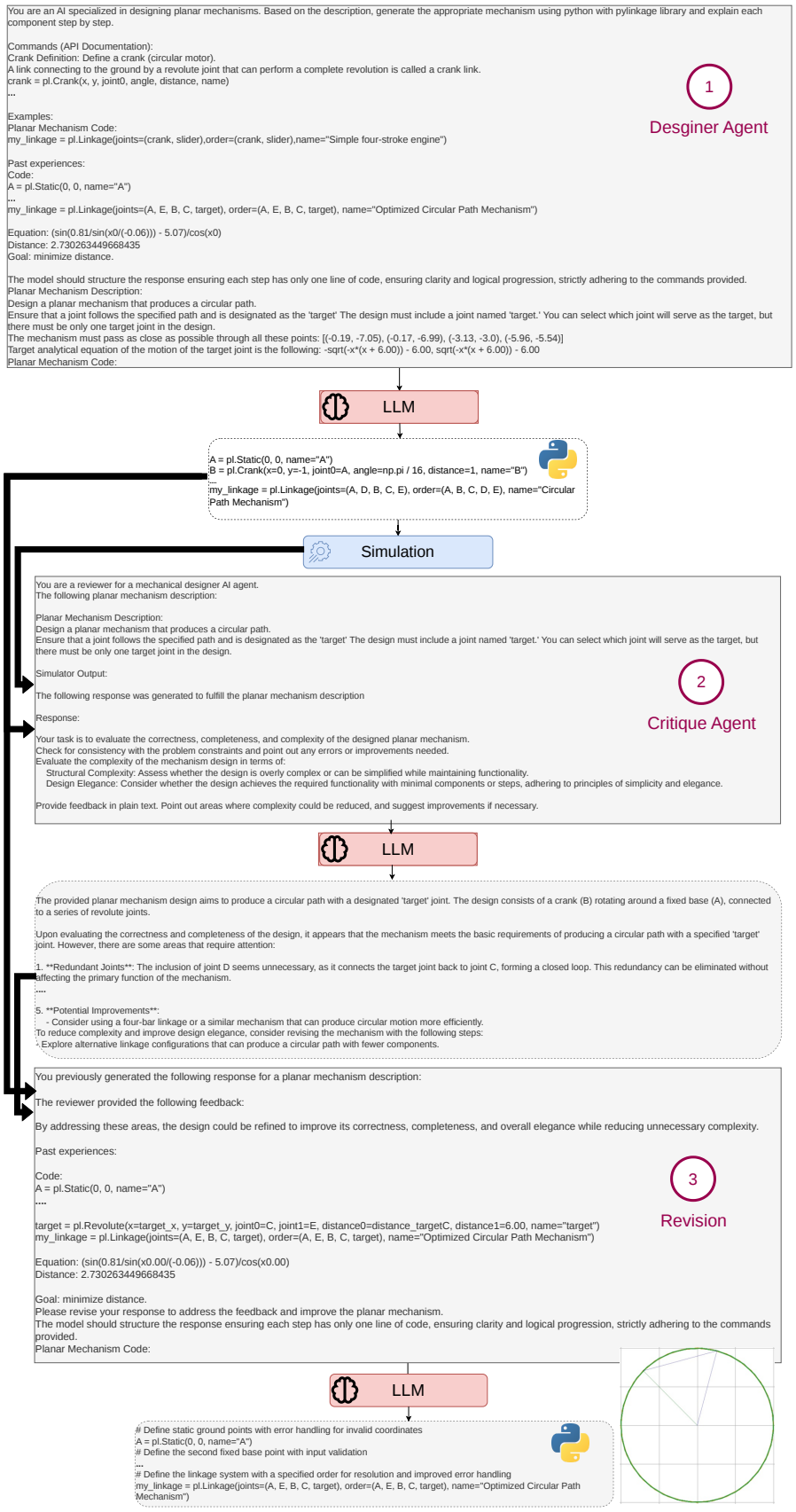
You are an AI specialized in designing planar mechanisms. Based on the description, generate the appropriate mechanism using python with pylinkage library and explain each component step by step.
Commands (API Documentation):
Crank Definition: Define a crank (circular motor).
A link connecting to the ground by a revolute joint that can perform a complete revolution is called a crank link.
crank = pl.Crank(x, y, joint0, angle, distance, name)
...
Examples:
Planar Mechanism Code:
my_linkage = pl.Linkage(joints=(crank, slider),order=(crank, slider),name="Simple four-stroke engine")
Past experiences:
Code:
A = pl.Static(0, 0, name="A")
...
my_linkage = pl.Linkage(joints=(A, E, B, C, target), order=(A, E, B, C, target), name="Optimized Circular Path Mechanism")
Equation: (sin(0.81/sin(x0/(-0.06))) - 5.07)/cos(x0)
Distance: 2.730263449668435
Goal: minimize distance.
The model should structure the response ensuring each step has only one line of code, ensuring clarity and logical progression, strictly adhering to the commands provided.
Planar Mechanism Description:
Design a planar mechanism that produces a circular path.
Ensure that a joint follows the specified path and is designated as the 'target' The design must include a joint named 'target.' You can select which joint will serve as the target, but there must be only one target joint in the design.
The mechanism must pass as close as possible through all these points: [(-0.19, -7.05), (-0.17, -6.99), (-3.13, -3.0), (-5.96, -5.54)]
Target analytical equation of the motion of the target joint is the following: -sqrt(-x*(x + 6.00)) - 6.00, sqrt(-x*(x + 6.00)) - 6.00
Planar Mechanism Code:
1
Desginer Agent
LLM
A = pl.Static(0, 0, name="A")
B = pl.Crank(x=0, y=-1, joint0=A, angle=np.pi / 16, distance=1, name="B")
...
my_linkage = pl.Linkage(joints=(A, D, B, C, E), order=(A, B, C, D, E), name="Circular Path Mechanism")
Simulation
You are a reviewer for a mechanical designer AI agent.
The following planar mechanism description:
Planar Mechanism Description:
Design a planar mechanism that produces a circular path.
Ensure that a joint follows the specified path and is designated as the 'target' The design must include a joint named 'target.' You can select which joint will serve as the target, but there must be only one target joint in the design.
Simulator Output:
The following response was generated to fulfill the planar mechanism description
Response:
Your task is to evaluate the correctness, completeness, and complexity of the designed planar mechanism.
Check for consistency with the problem constraints and point out any errors or improvements needed.
Evaluate the complexity of the mechanism design in terms of:
Structural Complexity: Assess whether the design is overly complex or can be simplified while maintaining functionality.
Design Elegance: Consider whether the design achieves the required functionality with minimal components or steps, adhering to principles of simplicity and elegance.
Provide feedback in plain text. Point out areas where complexity could be reduced, and suggest improvements if necessary.
2
Critique Agent
LLM
The provided planar mechanism design aims to produce a circular path with a designated 'target' joint. The design consists of a crank (B) rotating around a fixed base (A), connected to a series of revolute joints.
Upon evaluating the correctness and completeness of the design, it appears that the mechanism meets the basic requirements of producing a circular path with a specified 'target' joint. However, there are some areas that require attention:
1. **Redundant Joints**: The inclusion of joint D seems unnecessary, as it connects the target joint back to joint C, forming a closed loop. This redundancy can be eliminated without affecting the primary function of the mechanism.
....
5. **Potential Improvements**:
- Consider using a four-bar linkage or a similar mechanism that can produce circular motion more efficiently.
To reduce complexity and improve design elegance, consider revising the mechanism with the following steps:
- Explore alternative linkage configurations that can produce a circular path with fewer components.
You previously generated the following response for a planar mechanism description:
The reviewer provided the following feedback:
By addressing these areas, the design could be refined to improve its correctness, completeness, and overall elegance while reducing unnecessary complexity.
Past experiences:
Code:
A = pl.Static(0, 0, name="A")
....
target = pl.Revolute(x=target_x, y=target_y, joint0=C, joint1=E, distance0=distance_targetC, distance1=6.00, name="target")
my_linkage = pl.Linkage(joints=(A, E, B, C, target), order=(A, E, B, C, target), name="Optimized Circular Path Mechanism")
Equation: (sin(0.81/sin(x0.00/(-0.06))) - 5.07)/cos(x0.00)
Distance: 2.730263449668435
Goal: minimize distance.
Please revise your response to address the feedback and improve the planar mechanism.
The model should structure the response ensuring each step has only one line of code, ensuring clarity and logical progression, strictly adhering to the commands provided.
Planar Mechanism Code:
3
Revision
LLM
# Define static ground points with error handling for invalid coordinates
A = pl.Static(0, 0, name="A")
# Define the second fixed base point with input validation
...
# Define the linkage system with a specified order for resolution and improved error handling
my_linkage = pl.Linkage(joints=(A, E, B, C, target), order=(A, E, B, C, target), name="Optimized Circular Path Mechanism")

Figure 5: Neuro-symbolic closed-loop for planar mechanism synthesis: (1) Designer Agent generates executable mechanism hypotheses from the specification and memory; (2) Executor runs candidates and Critique Agent critiques and issues structured revision directives; (3) Designer Agent performs the revision, then repeats. The circle in the picture represents the path traced by the mechanism as simulated by the simulator.

This cycle of design, simulation, evaluation and refinement repeats until the mechanism meets the predefined convergence or performance criteria.

Algorithm 1 Iterative Planar Mechanism Design with Dual Agent, Simulation, and Evaluation

Input: Initial design parameters, Simulator on $\mathcal{S}im$, evaluation functions, target trajectory $\mathcal{T}$
Output: Optimized planar mechanism design $\mathcal{M}(\theta)$
while Design objective not met **do**
    Design: Generate a new planar mechanism $\mathcal{M}(\theta)$ using the $\mathbb{D}a$.
    First Simulation: Simulate $\mathcal{M}(\theta)$ on $\mathcal{S}im$
    if Simulation successful **then**
        Compute distance metric $d(\mathcal{M}(\theta), \mathcal{T})$
        Discover SR equation
        Store $\mathcal{M}(\theta)$ in memory
    end if
    - $\mathbb{C}a$ evaluates the mechanism and provides feedback

    Refinement: Update the design strategy using feedback from the evaluation
    Design (Refinement): Generate a refined mechanism $\mathcal{M}'(\theta)$ using the updated strategy.
    Simulation: Simulate the refined mechanism $\mathcal{M}'(\theta)$ on $\mathcal{S}im$
    if Simulation successful **then**
        Store $\mathcal{M}'(\theta)$ in memory
    else
        Discard $\mathcal{M}'(\theta)$
        continue to the next iteration
    end if
    Evaluation:
    - Re-compute the distance metric $d(\mathcal{M}'(\theta), \mathcal{T})$
    - Re-assess SR feedback

end while
return Best mechanism $\mathcal{M}(\theta)$ or refined mechanism $\mathcal{M}'(\theta)$ meeting design criteria

G Evaluation Metrics

Point Set Alignment. For a generated mechanism m_t that yields an end effector trajectory $GeneratedPoints$, we first perform point set registration via the ICP algorithm to establish the correspondence with the target motion profile. The algorithm aligns the generated point set with the target point set through rigid transformations, which compensates for the differences in the orientation of the coordinate systems.

Chamfer distance The Chamfer distance between two point sets P and Q measures their similarity by summing, for each $p \in P$, the distance to its nearest neighbour in Q, and doing the same from Q to P. After trajectory alignment, the mechanism's performance is quantified using the Chamfer distance, which provides this bidirectional measure of spatial proximity between the two point sets: Formally, one common formulation is 6.

$$d_{Chamfer}(P,Q) = \frac{\sum_{p\in P} \min_{q\in Q} \|p-q\|^2}{|P|} + \frac{\sum_{q\in Q} \min_{p\in P} \|q-p\|^2}{|Q|}$$

Semantic Success Rate We define *Semantic success* as a binary indicator of whether a generated program yields an executable. Formally, given a program P, we denote its execution in the simulator as $T_{\text{pred}} = \text{Sim}(P)$. Then

$$S(P) = \begin{cases} 1, & \text{if } P \text{ parses and executes without error,} \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Implementation notes: any parse, compilation, runtime error, timeout, or early termination that prevents a simulation from being produced is treated as a failure (i.e., $S = 0$). All Python code generated by the LLM was automatically formatted with `black` to ensure a consistent style across implementations. Using `black` produces deterministic, idempotent formatting that reduces stylistic variation introduced by code generation and simplifies code inspection and review.

H Dataset Details

Let

$$\mathcal{S} = \{\, \mathcal{C}(r, x_1, y_1),\ \mathcal{E}(a, b, x_1, y_1),\ \mathcal{L}(x_1, y_1, x_2, y_2),\ \mathcal{P}(a, h, k),\ \mathcal{B}(a),\ \mathcal{N}(\text{series}) \},$$

where $r, a, b > 0$, $(x_1, y_1), (x_2, y_2), (h, k) \in \mathbb{R}^2$, $a \neq 0$, and "series" ranges over the standard NACA four-digit codes $\{2000, \ldots, 3000\}$. Each shape is defined on 7.

For each $S \in \{\text{circle}, \text{ellipse}, \text{line}, \text{parabola}, \text{LB}, \text{NACA}\}$, we sample 5 independent instances by drawing all shape parameters (centres, radius, semi-axes, vertex offsets, scales, chord lengths, and series codes) uniformly from their prescribed domains. We then choose $n \in \{4\}$ uniformly at random and sample each curve at n parameter values to obtain $\{(x_i, y_i)\}_{i=1}^{n}$, which serve as the target profiles.

Shape eligibility criteria. The six target classes were selected against four criteria. **(i) Analytic ground truth.** Each shape admits a closed-form equation, so a coupler path is correct or incorrect by construction. This is the property the symbolic interface relies on: SR can recover an analytical surrogate and compare it to a known target, yielding verifiable rather than subjective feedback. **(ii) Curve-family diversity.** The set deliberately samples qualitatively different families, linear (line), conic (circle, ellipse, parabola), transcendental quartic (LB), and an engineered piecewise-analytic profile (NACA), together covering open and closed, bounded and unbounded, single- and multi-lobe,

$$
\begin{aligned}
\mathcal{C}(r, x_1, y_1) &= \{(x, y) \mid (x - x_1)^2 + (y - y_1)^2 = r^2\}, \\
\mathcal{E}(a, b, x_1, y_1) &= \{(x, y) \mid \tfrac{(x-x_1)^2}{a^2} + \tfrac{(y-y_1)^2}{b^2} = 1\}, \\
\mathcal{L}(x_1, y_1, x_2, y_2) &= \left\{(x, y) \mid y - y_1 = \tfrac{y_2-y_1}{x_2-x_1}(x - x_1)\right\}, \\
\mathcal{P}(a, h, k) &= \{(x, y) \mid y = a\,(x - h)^2 + k\}, \\
\mathcal{B}(a) &= \{(x, y) \mid (x^2 + y^2)^2 = 2\,a^2\,(x^2 - y^2)\}, \\
\mathcal{N}(\text{series}) &= \{(x, y) \mid (x, y) \text{ satisfies the standard NACA upper/lower formulas}\}.
\end{aligned}
$$

and symmetric and asymmetric curves. **(iii) Emergent, kinematic difficulty.** We did not order the shapes by algebraic degree; the difficulty is an emergent property of kinematic feasibility. Table 2 orders mean Chamfer distance as NACA $<$ ellipse $<$ line $<$ circle $<$ LB $<$ parabola, an ordering that algebraic degree does not predict: the parabola is a degree-two conic yet is hardest by two orders of magnitude, because it is open and unbounded and cannot be matched by the closed curve a coupler point traces (the corresponding critique reasoning is shown in App. N.1), whereas the most engineered profile, the NACA airfoil, is smooth, bounded, and easiest. Shapes without a closed-form expression, such as free-hand or purely data-defined curves, were therefore excluded: they provide no analytic ground truth for SR to recover and no rigorous correctness signal, which is exactly the property the method depends on.

Lemniscate of Bernoulli The lemniscate of Bernoulli is a figure-eight-shaped curve defined by the quartic equation $(x^2 + y^2)^2 = 2\,a^2\,(x^2 - y^2)$, where the parameter a controls its overall size. It is symmetric about both axes and consists of two lobes meeting at the origin, exemplifying a simple rational algebraic curve.

NACA Four-Digit Airfoils Each NACA four-digit airfoil is specified by a code (e.g., "2412") indicating maximum camber, camber position, and thickness relative to chord length. Upper and lower surface coordinates are generated by piecewise analytic formulas that ensure smooth curvature and typical aerodynamic properties, making them standard in preliminary wing design.

I Designer Agent ($\mathbb{D}a$) Prompt

The prompt below (30) serves as the template for the $\mathbb{D}a$ Prompt. It provides the structure and context that guides the $\mathbb{D}a$ in understanding how to approach and execute design of a planar mechanism.

```
You are an AI specialized in designing planar mechanisms. Based on the description,
generate the appropriate mechanism using Python with the pylinkage library and
explain each component step by step.
Commands (API Documentation): {api_doc}
Examples:
# Example 1
import pylinkage as pl
crank = pl.Crank(x=0, y=0, joint0=(0, 0), distance=1, angle=0.1, name="Crank")
slider = pl.Linear(x=2, y=0, joint0=crank, joint1=(0, 0), joint2=(1, 0),
revolute_radius=1.5, name="target")
{memory}
The analytical equation describing the motion of the target joint in the above
code is given by: {equation} The Chamfer distance of the target equation in the
above code is: {score}
Our goal is to minimise the distance. Therefore, the greater the distance, the
more it is not following the target motion and deviating from the intended path.
```

```
Planar Mechanism Description:
{description}
The mechanism must pass as close as possible through all these points: {points}
Target analytical equation of the motion of the target joint: {target_equation}
Planar Mechanism Code:
```

Legend for Colour Coding:

- **Brown**: Agent Role.
- **Violet**: Chain-of-Thoughts (CoT).
- **Red**: API Documentation and Mechanism Descriptions.
- **Blue**: Examples and code snippets.
- **Cyan**: Memory: previously retained knowledge.
- **Green**: Key points or constraints (e.g., points through which the mechanism must pass).
- **Orange**: Target analytical equation defining ideal behavior.
- **Purple**: Final mechanism code output.

J CRITIQUE AGENT ($\mathbb{C}a$) PROMPT

The prompt below (30) acts as the template for the $\mathbb{C}a$ Prompt. It outlines the format and expectations that help the $\mathbb{C}a$ deliver clear, structured, and constructive critiques to support design improvement and decision-making.

```
You are a reviewer for a mechanical designer AI agent.
The following planar mechanism description: {description}
Simulator Output: {simulator_output}
{memory}
The analytical equation describing the motion of the target joint in the above
code is given by: {equation} The Chamfer distance of the target equation in the
above code is: {score}
Our goal is to minimise the distance. Therefore, the greater the distance, the
more it is not following the target motion and deviating from the intended path.
The following response was generated to fulfill the planar mechanism description:
Response: {designer_response}
Your task is to evaluate the correctness, completeness, and complexity of the
designed planar mechanism.
Check for consistency with the problem constraints and point out any errors or
improvements needed.
Evaluate the complexity of the mechanism design in terms of: Structural Complexity:
Assess whether the design is overly complex or can be simplified while maintaining
functionality.
  Structural Complexity: Assess whether the design is overly complex or can be
simplified while maintaining functionality.
Design Elegance: Consider whether the design achieves the required functionality
with minimal components or steps, adhering to principles of simplicity and
elegance.
Provide feedback in plain text. Point out areas where complexity could be reduced,
and suggest improvements if necessary.
```

Legend for Colour Coding:

- **Brown**: Agent's Role.
- **Red**: Planar mechanism description.
- **Orange**: Simulator output.
- **Cyan**: Memory: previously retained knowledge.
- **Purple**: Designer's response.
- **Teal**: Agent's task.

K Designer Agent ($\mathbb{D}a$) Revision Prompt

The prompt below (31) serves as the template for the $\mathbb{D}a$ Revision Prompt. It provides the structure and context that guides the $\mathbb{D}a$ in understanding how to approach and execute design revisions effectively.

```
You previously generated the following response for a planar mechanism description: {designer_response}
The reviewer provided the following feedback: {critique_response}
Simulator Output: {simulator_output}
{memory}
The analytical equation describing the motion of the target joint in the above code is given by: {equation} The Chamfer distance of the target equation in the above code is: {score}
Our goal is to minimise the distance. Therefore, the greater the distance, the more it is not following the target motion and deviating from the intended path. Please revise your response to address the feedback and improve the planar mechanism.
The model should structure the response ensuring each step has only one line of code, ensuring clarity and logical progression, strictly adhering to the commands provided.
Planar Mechanism Code:
```

Legend for Colour Coding:

- **Purple**: Previously generated response (initial planar mechanism code).
- **Magenta**: Reviewer feedback.
- **Orange**: Simulator output.
- **Cyan**: Memory: previously retained knowledge.
- **Teal**: Agent's task.

L Shape Complexity

In this section, we present metrics for quantifying shape complexity and compare them to our ground truth and agent-generated forms (Table 2). This concise analysis illustrates how fidelity deteriorates (or remains stable) as geometric intricacy increases.

M Symbolic Intermediary

Protocol. We refit every intermediary produced in the SR-enabled runs under the settings used in the loop: SR over $+, -, \times, \div, \wedge, \sin, \cos$ with a maximum expression size of 30 and 2500 search iterations, fitting $y \sim f(x)$ on the trace of the end-effector joint.

Shape	Mean Chamfer distance
LB	8.5654 ± 0.1632
Circle	7.4564 ± 0.5147
Ellipse	5.4398 ± 0.1789
Line	7.1046 ± 0.5392
NACA	1.1684 ± 0.0659
Parabola	853.2922 ± 6.7232

Table 2: The mean Chamfer distances ($\pm$ standard error) are shown for each shape category, calculated from the motion of the contours generated by the planar mechanism and the ground-truth contours.

N SR-Enabled vs. SR-Absent Critique Comparison

The following paired excerpts illustrate how the symbolic regression bridge changes the character of the critique agent's reasoning. In each pair, the same model and target shape are used; only the availability of SR-fitted equations differs. Excerpts are lightly abridged for space; full transcripts are available in the supplementary material.

N.1 Parabola - Qwen (4B), iteration 4

SR-enabled critique (SR equations available; 15 accumulated expressions):

> The equation given is $y = (x - 7)^2 + 3$, which is a parabola. The points provided (like $(32.56, 656.14)$) lie on this curve. The mechanism should be designed to approximate this path. [. . .] Now, I need to verify if this setup can produce a parabolic path. The movement of joint B (the crank) is circular around A. As B rotates, C moves in a circle around D. The target E is connected to both B and C, so its position is determined by the relative positions of B and C.
>
> Does this configuration result in E tracing a parabola? The parabola is a quadratic curve, which is a second-order curve. A four-bar linkage can approximate conic sections, but the exact path depends on the link lengths. [. . .] The distance between B and D is $\sqrt{40^2 + 20^2} = \sqrt{2000} \approx 44.72$. Using the law of cosines: $40^2 = 30^2 + 20^2 - 2 \cdot 30 \cdot 20 \cos\theta \Rightarrow \cos\theta = -0.25 \Rightarrow \theta \approx 104.48^\circ$. This is possible, so C can be positioned accordingly.

SR-absent critique (no SR equations):

> The code defines a crank (B) rotating around point A (7,3), and a linear slider (C) connected to B and fixed points A and D (14,3). [. . .] A linear joint is typically a slider that moves along a straight line. If A and D are at $(7, 3)$ and $(14, 3)$, the line AD is horizontal. So the slider C would move along this horizontal line. However, the parabola equation is a curve, not a straight line. This seems contradictory. [. . .] The design is incorrect. The linear joint can't produce a parabola. [. . .] The mechanism doesn't meet the requirements.

Key difference. The SR-enabled critique identifies the curve class (parabola), computes whether the four-bar linkage can kinematically realise it via the law of cosines, and evaluates link-length compatibility, all geometric reasoning mediated by the symbolic representation. The SR-absent critique detects a topological error (linear slider vs. curved path) but does not engage analytically with the curve geometry or propose quantitative corrections.

N.2 NACA Airfoil - Qwen (4B), iteration 3

SR-enabled critique (18 accumulated SR equations):



Figure 8: Paired critique comparison showing how symbolic regression changes the nature of reasoning. **Left**: SR-enabled critiques identify curve classes, apply kinematic calculations (law of cosines), and compute explicit distances. **Right**: matched SR-absent critiques default to structural heuristics and generic improvement advice. Coloured tags indicate the reasoning types present.

> The static points A and D are at $(0, 0)$ and $(3, 0)$. [. . .] Joint C is at $(3, 2)$, with distances of 3 from B and 1.5 from D. If B is at $(0, 1)$ and C is at $(3, 2)$, the distance between B and C would be $\sqrt{(3-0)^2 + (2-1)^2} = \sqrt{10} \approx 3.16$, which is close to the specified 3 units. [. . .] The analytical equation is complex and might not be directly compatible with the linkage's kinematic constraints. [. . .] The distances between joints (especially between C and E) could be adjusted to better approximate the desired path.

SR-absent critique (no SR equations):

> The four-bar linkage is a common choice for approximating curved paths, especially when combined with specific link lengths and angles. [. . .] The code uses fixed link lengths (distance=1, 2, etc.), but without dynamic adjustments, the linkage might not be able to precisely follow the curve. [. . .] The positions of the fixed points and link lengths should be optimized to align with these points.

Key difference. The SR-enabled critique performs explicit distance verification ($\sqrt{10} \approx 3.16$ vs. specified 3), flags incompatibility between the SR-fitted analytical equation and the linkage topology, and localises the source of error to a specific joint pair (C–E). The SR-absent critique offers generic optimisation advice without quantitative grounding.

N.3 Circle - Qwen (4B), iteration 0

SR-enabled critique:

> The crank is connected to this centre, and the target is at the end of the crank. As the crank rotates, the target should move in a circle around the centre. The crank's length is 1, which matches the radius. [. . .] But the link between the crank and B is defined with a distance of 13.45, which is the distance from A to B. If the crank is rotating around A, and the link is fixed between the crank's end and B, then the target would move in an ellipse, not a circle.

SR-absent critique:

> The equation given is a circle centred at $(3, -1)$ with radius 1. The crank is connected to this centre, and the target is at the end of the crank. [. . .] For $(3.6, -1.8)$: $(3.6-3)^2 + (-1.8+1)^2 = 0.36 + 0.64 = 1.0 \rightarrow$ exactly on the circle.

Key difference. In both conditions, the critique agent engages with the circle equation. However, the SR-absent run happened to receive a near-optimal design from the first iteration (Chamfer $= 0.003$), so the critique merely verifies correctness. The SR-enabled critique detects a subtler design flaw, a distance mismatch that would produce an ellipse rather than a circle, demonstrating that symbolic feedback supports *error diagnosis* even on simple shapes.

N.4 Quantitative reasoning taxonomy

To complement the paired excerpts, we scored every critique entry by pattern matching against an eight-category taxonomy, splitting by model and SR condition. Six categories record analytical engagement: curve-class identification, kinematic feasibility calculation, point-residual verification, explicit distance calculation, quantitative parameter reasoning, and engagement with a discovered SR equation. Two record non-analytical engagement: structural/topological commentary and generic improvement advice. Two roll-ups summarise them: an entry is *analytical* when it matches any of the six analytical categories, and *heuristic-only* when it carries structural or generic commentary alone.

The robust cross-model signal. Engagement with discovered symbolic structure rises with SR for *every* model (Gemma $0\% \rightarrow 2\%$, Llama $1\% \rightarrow 3\%$, Qwen $16\% \rightarrow 26\%$): the interface measurably changes *what* critiques reason over, independently of architecture.

Grounding, not frequency: a compositional caveat. A naïve aggregate reading of Table **??** suggests SR-absent critiques are *more* analytical (explicit distance calculation in 11.6% of SR-off vs. 10.6% of SR-on entries, and a higher overall analytical share, 39.5% vs. 28.5%). This aggregate pattern is a *compositional* (Simpson) effect: the SR-off pool is dominated by Llama 70B, which emits analytical-sounding language at high frequency regardless of whether it connects to the target geometry (its grounded-analytical share is only 25–32%). Conditioning on model removes the paradox: the reasoning-trained compact model Qwen 4B is precisely where SR raises grounded

analytical reasoning ($96\% \rightarrow 98\%$), while the larger dense models do not increase it. The effect is therefore one of *grounding* anchored in discovered equations, an architecture-dependent shift (consistent with §3.4), which aggregate metrics alone cannot capture.

O EMERGENT CURRICULUM

Section 5 claims the critique loop induces an *emergent curriculum*: structural commentary dominates early and gives way to parametric refinement. We test this directly. Over the SR- and feedback-enabled experiments, the share of sentences flagged *structural* (topology, joint changes, redesign) falls from 5.69% at iteration 0 to 3.39% at iteration 19, while *parametric* sentences (link-length, position, angle adjustments) stay essentially flat ($\approx 4.9\%$); the structural-to-parametric ratio declines from 1.08 to 0.66. A χ^2 test of structural sentence frequency, early vs. late iterations, is highly significant ($\chi^2 = 278.67$, $p \approx 1.5 \times 10^{-62}$), whereas the parametric change is only marginal ($\chi^2 = 4.23$, $p = 0.04$): difficulty shifts from *what topology* to *which parameters* without any explicit curriculum design. Representative early-structural/late-parametric pairs from the same run:

> **Gemma 12B, LB.** *Iter 4 (structural):* "the simulation output indicates a critical issue: `UnbuildableError` [...] the underlying mechanism design is fundamentally flawed." $\rightarrow$ *Iter 16 (parametric):* "Re-evaluate joint positions & link lengths: the x, y coordinates of joints C and D, and the `distance0`, `distance1` values, need to be carefully calculated."
>
> **Gemma 12B, circle.** *Iter 2 (structural):* "Prioritise a four-bar linkage: the most significant improvement would be to restructure the mechanism." $\rightarrow$ *Iter 17 (parametric):* "Optimise link lengths: incorporate an optimisation routine to adjust the link lengths."

P QUALITATIVE RESULTS

Figures 5,7,9,11,13, and 15, shows the ground-truth targets, while Figures 6, 8,10, 12, 14, and 16 shows the corresponding outputs produced by our agent. Close inspection reveals that many of the generated results closely align with their targets, accurately capturing both the global structure and fine-scale details. This strong visual correspondence highlights the effectiveness of our method in approximating the desired outputs across a range of diverse test cases.

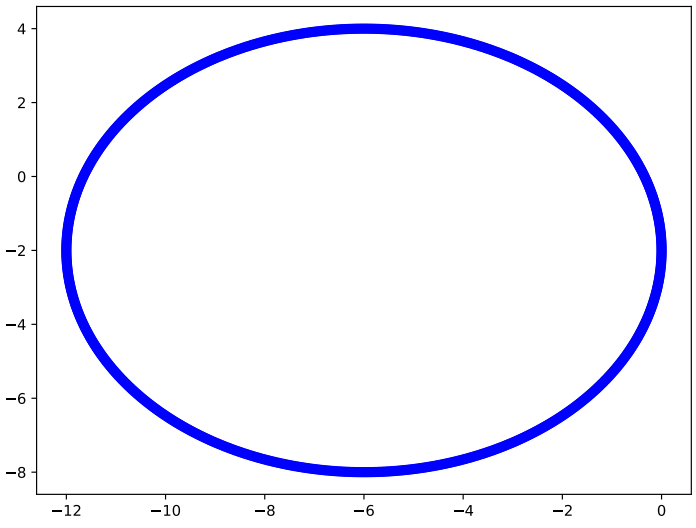

Figure 5: Ground truth Ellipse: the original target Ellipse used for evaluation.

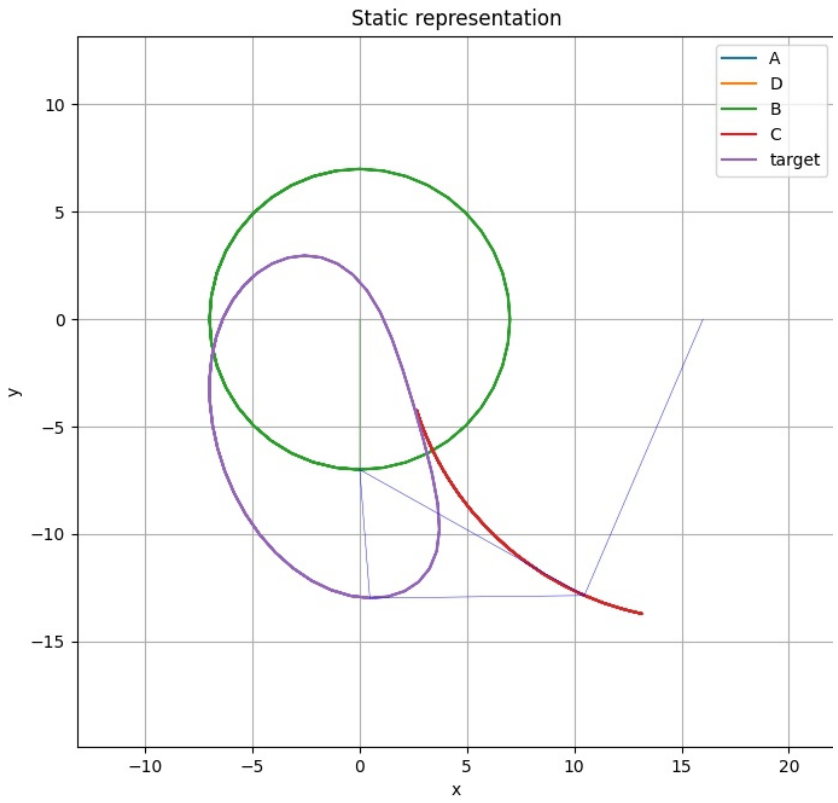

Figure 6: Generated Ellipse: the Ellipse synthesised by our agent.

Q Implementation and Hardware Configuration

At each iteration the design agent generates 3 candidate mechanisms at sampling temperature 0.8; we set the maximum number of refinement rounds to $R_{\max} = 20$ and the convergence threshold to $\epsilon = 0.05$, so a single run consumes at most 60 simulator evaluations. Simulation uses PyLinkage (Farajallah, 2024), symbolic regression uses PySR (Cranmer, 2023), and the iterative closest point algorithm (Drakoulis, 2023; **?**) aligns simulated and target point clouds before evaluation. Prompts for the design, critique, and revision stages are reproduced in App. I, J, and K.

We used a workstation equipped with five NVIDIA RTX 3090 graphics cards, each with 24 GB of VRAM and 72 GB of system RAM.

R Token Cost of the Loop

§Q states the loop in evaluations. This appendix states it in tokens, because the quantity a reader needs in order to reproduce or re-scope the method is the cost of one design iteration, and because that cost turns out to be set by which model runs the loop.

Protocol. Every call the pipeline made was logged with its full prompt and completion. We tokenised that corpus with each model's own tokeniser, taken from the published vocabularies for Llama-3.3, Gemma-3 and Qwen3. As a check, the Qwen3 counts were reproduced against the serving runtime's own reported prompt length on 50 sampled segments, with exact agreement on all 50. Counts cover the raw prompt and completion text; chat-template control tokens are excluded, and at an upper bound of 25 tokens per call they would change any figure below by under 0.5%. We report medians. An iteration is 3 calls in the open loop and 9 with the critique loop, so per-iteration figures

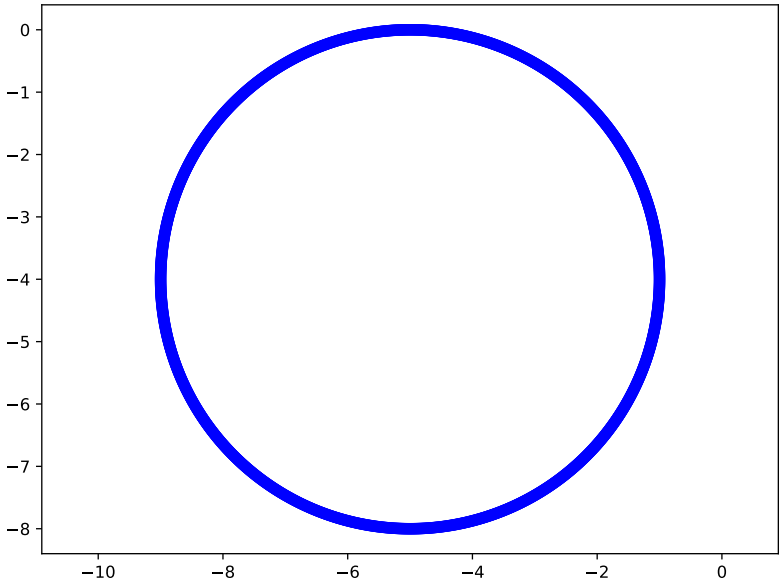

Figure 7: Ground truth Circle: the original target Circle used for evaluation.

Model	Loop	In	Out	Think	×
Llama	open	7.0	1.2	0.0	1.00
	critique	17.8	4.2	0.0	2.71
Gemma	open	6.5	0.9	0.0	0.97
	critique	19.7	8.0	0.0	3.37
Qwen	open	6.8	19.4	17.5	3.24
	critique	55.9	28.3	21.2	10.37

Table 3: Median tokens per design iteration, in thousands. **Loop** is *open* (no critique) or *critique*. **Think** is the part of **Out** emitted inside `<think>` blocks. × is the total cost of one iteration relative to Llama in the open loop. Cost is set by the model and the loop: the six target shapes span only 11% between cheapest and dearest.

divide by that; dividing the exactly-conforming runs by a fixed 20 refinement rounds instead moves every figure by under 3%.

The loop costs about 8k tokens per iteration, and the critique triples it. Table 3 gives the per-iteration cost. In the open loop Llama and Gemma agree to within 3%, at 8.2k and 8.0k tokens per iteration, which is what one expects when three candidates are drawn from a fixed prompt template. Turning on the critique loop multiplies that by 2.71 for Llama, 3.46 for Gemma and 3.20 for Qwen, tracking the 3× increase in calls. The extra input is the mechanism under review and, at the revision stage, the critique as well.

The target curve is not what costs. Across the six shape classes the median cost of an iteration spans 22.2k to 24.6k tokens, a range of 11%, against a 5.7× spread between models. Difficulty in the sense of §L does not translate into token spend: the budget is a property of the loop and the model, and a practitioner scoping this method should size it that way.

The models are charged for what they emit. Table 4 separates a single call by stage. All three receive design prompts within 7% of each other, at 2.2k to 2.4k tokens, confirming that the input side

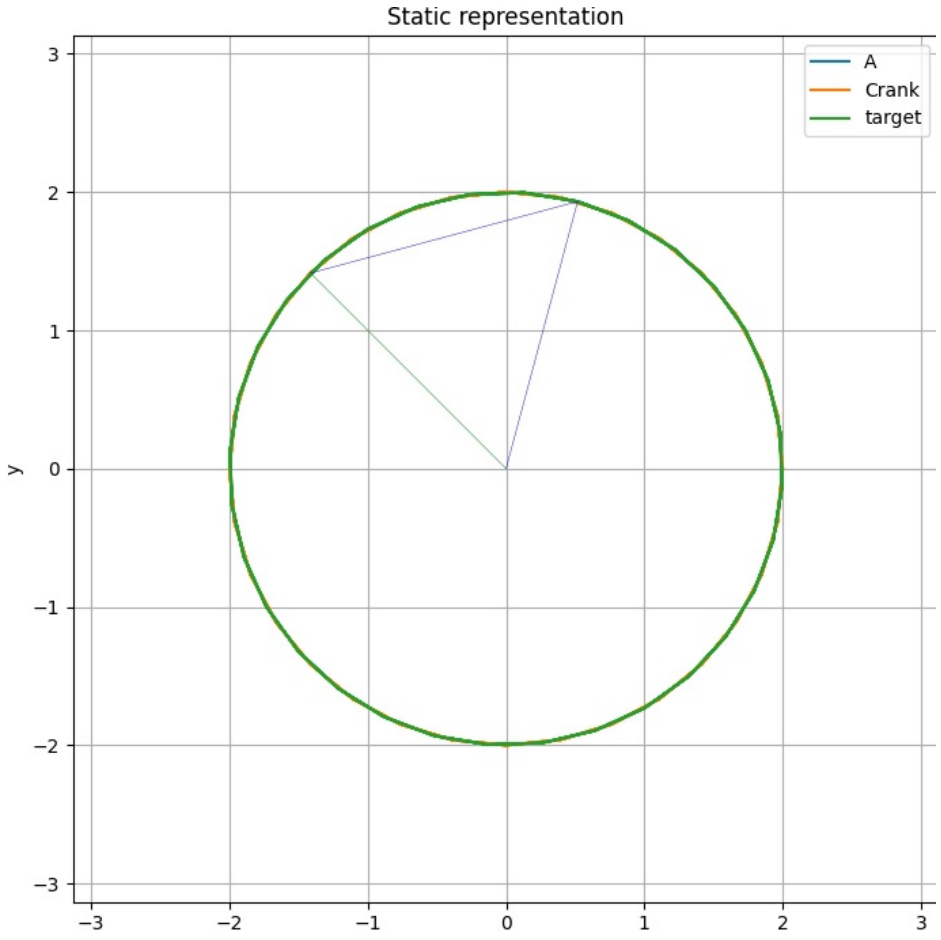

Figure 8: Generated Circle: the Circle synthesised by our agent.

Model	Stage	In	Out	Think
Llama	design	2,365	411	0
	critique	1,547	580	0
	revision	1,978	405	0
Gemma	design	2,220	351	0
	critique	1,380	1,261	0
	revision	2,780	381	0
Qwen	design	2,262	6,177	5,393
	critique	7,401	1,574	735
	revision	8,880	1,617	933

Table 4: Median tokens in a single call, by pipeline stage. The three models receive design prompts of near-identical size, because the prompt is a fixed template; they diverge on what they emit, and Qwen's critique and revision prompts inflate because those prompts quote its own previous response back to it.

is fixed by our template. The divergence is on the output side. Llama and Gemma answer a design prompt in roughly 350 to 400 tokens. Qwen answers in 6,177, of which 5,393, or 87%, sit inside `<think>` blocks that are discarded before the program is extracted. That difference then feeds back into the input side: because the critique and revision prompts quote the previous response, Qwen's prompts at those stages reach 7,401 and 8,880 tokens, three to five times what the other two models

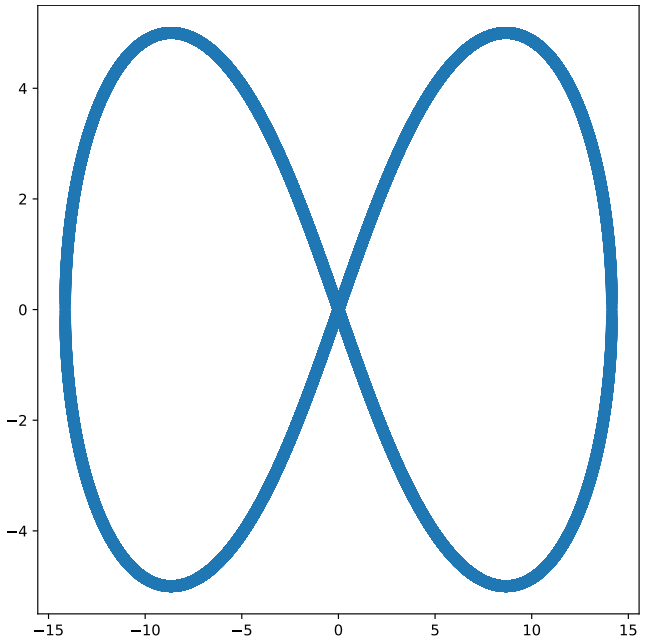

Figure 9: Ground truth LB: the original target LB used for evaluation.

see for the same work. One iteration of the full loop costs $10.4\times$ more with Qwen than with Llama in the open loop.

This bears on how the trace analyses should be read. Qwen is the only one of the three models that emits a visible reasoning channel, and 82% of everything it generates across the campaign sits in that channel. The taxonomy of §N.4 and the grounding analysis therefore read a surface that exists for one model and is absent for the other two, which is a reason to treat those comparisons as within-model evidence.

S Extended Quantitative Result

In this section, we present the outcomes of our complete set of experiments, summarising performance across all six shape categories.

S.1 Full Enum+GA Baseline Comparison

Table 5 compares the three LLM agents (with Feedback enabled) against all six Enum+GA baseline configurations spanning exhaustive topology enumeration over two linkage types (4-bar, 6-bar) and three GA parameter-tuning budgets (Pop 3/Gen 20, Pop 6/Gen 20, Pop 60/Gen 300). Values are per-shape normalised Chamfer ratios (LLM median / GA median, averaged over shapes); values below 1 indicate the LLM outperforms the GA.

Why the Enum+GA baseline. We chose exhaustive topology enumeration with genetic-algorithm parameter tuning for four reasons. **(i) Established standard.** Enumeration of kinematic topologies followed by population-based dimensional optimisation is the canonical approach to path synthesis in computational kinematics (García-Marina et al., 2020; Tsai, 1999; Romero et al., 2019; Vasiliu & Yannou, 2001). **(ii) Strong, complete comparator.** Enumerating both 4-bar and 6-bar topologies covers a large search space, so the comparison is against a thorough search. **(iii) Budget control.** The GA budget is tunable, which lets us match the agent's evaluation budget exactly (Pop 3 / Gen 20, 60 evaluations), double it (Pop 6 / Gen 20, 120 evaluations, the setting reported in Table 1), and report a $300\times$ upper bound (Pop 60 / Gen 300, 18,000 evaluations); this isolates reasoning efficiency from

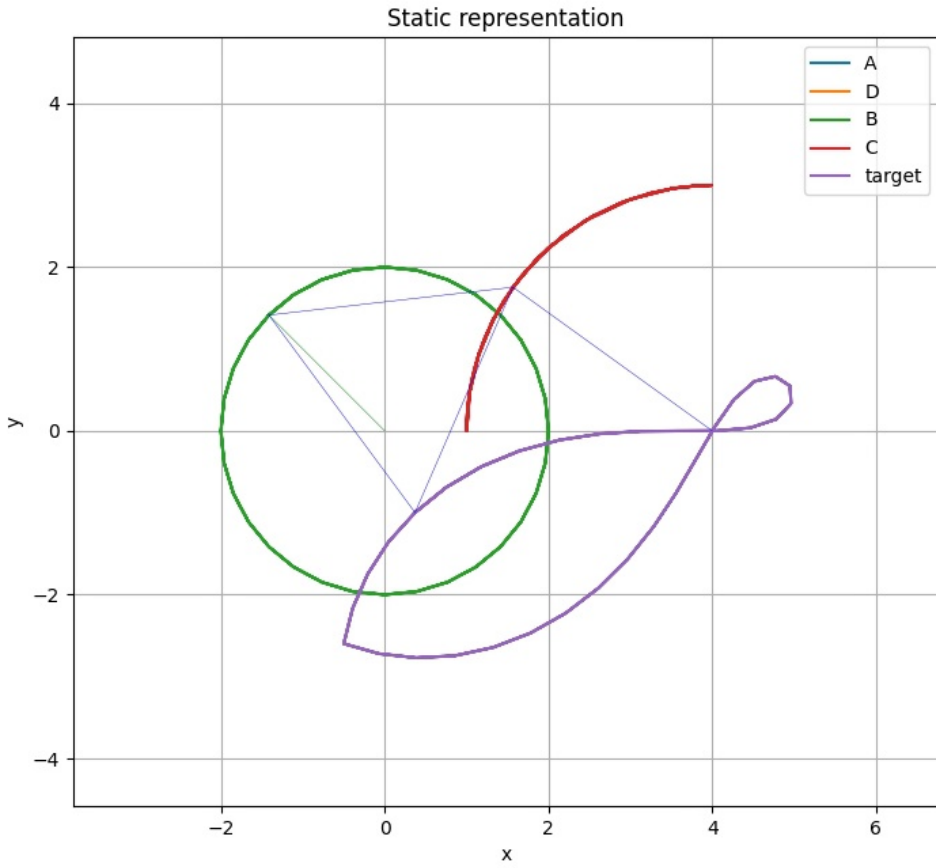

Figure 10: Generated LB: the LB synthesised by our agent.

raw compute. **(iv) Verified near-optimal.** Convergence is confirmed for every topology and shape (App. S.2), so the baseline is a fair reference.

S.2 Enum+GA Convergence

Fig. 21 shows the best-so-far fitness (Chamfer distance, log scale) as a function of GA evaluations for the strong Enum+GA baseline (Pop 60 / Gen 300). Each column is a shape class; each row is a linkage topology. Thin lines show individual runs (5 per condition) and the bold line shows the median. In all 18 conditions the curves plateau well before the 18,000-evaluation budget is exhausted, confirming that the GA has converged.

S.3 Per-Experiment Metrics

Table 6 displays quantitative metrics, such as the average Chamfer distance and the number of convergence steps, for agent-generated outputs.

T Memory dynamics

The impact of memory and the number of retrieved design examples depends on both architecture and scale. Qwen, a compact model with step-by-step reasoning training, benefits consistently from increasing memory (Mem $0 \rightarrow 2$), with aggregate means falling by ≈ 11.6 at 2 examples ($150.498 \rightarrow 138.882$) and by ≈ 11.4 at 3 examples ($152.391 \rightarrow 141.034$), with the largest impact on

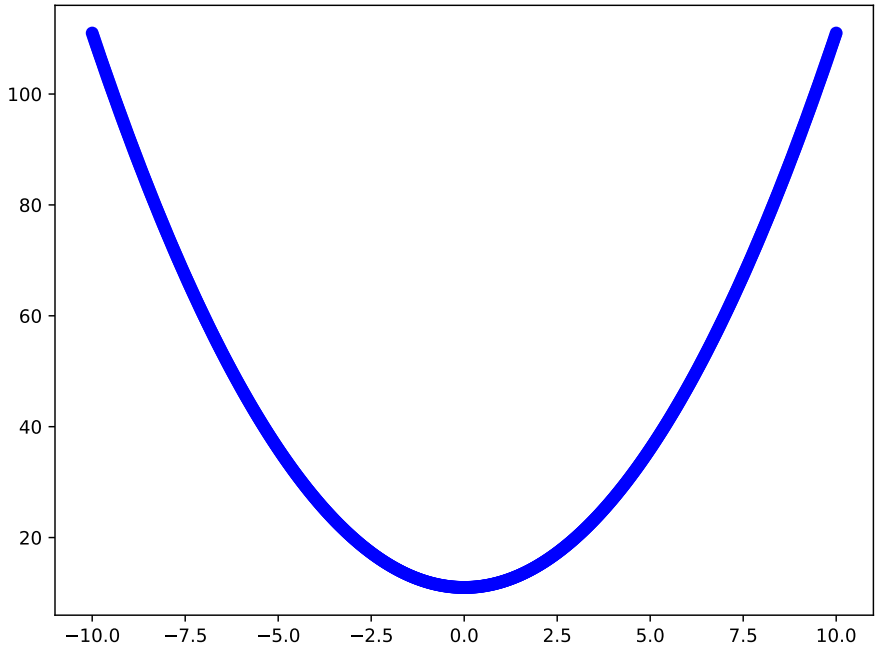

Figure 11: Ground truth Parabola: the original target Parabola used for evaluation.

the most challenging shape (Parabola). By contrast, Llama, a large dense model, shows a memory–examples interaction: memory helps at 2 examples ($147.819 \to 143.836$, $\Delta \approx -3.98$) but can harm at 3 examples ($142.515 \to 144.959$, $\Delta \approx +2.44$), suggesting that very large models are more sensitive to the volume of contextual examples (additional examples can induce interference). Gemma displays only modest, variable responses to memory and example count (e.g., 2 examples, Mem $0 \to 2$: $153.267 \to 153.038$, $\Delta \approx -0.23$).

Architecture and scale. Across the evaluated shapes, model family and training regime were more predictive of performance than raw parameter count. The Large Reasoning Model (LRM)-style model Qwen achieved the best aggregate minimisation, largely driven by its superior performance on the Parabola task, whereas the dense Llama excelled on the other shapes (Circle, Ellipse, Line). Averaged over conditions, Parabola means were Qwen $= 833.09$, Llama $= 845.99$, Gemma $= 883.79$; for the other shapes Llama attained the lowest means (Circle $= 5.00$, Ellipse $= 4.06$, Line $= 3.54$). On NACA and LB, the models were close (≈ 1.13 and ≈ 8.1–8.2 for Qwen and Llama), with Gemma uniformly higher. All models benefit from the combination of SFB and external feedback, but the magnitude varies: Qwen exhibits the largest absolute improvements under SFB+feedback, Llama moderate gains, and Gemma the smallest — consistent with the view that step-by-step reasoning training enables better exploitation of structured symbolic feedback.

U MEMORY — FORMAL DESCRIPTION AND LEARNING DYNAMICS

This appendix expands the memory component summarised in Section 2.

Formal description. Let $M_t = \{m_1, m_2, \ldots, m_n\}$ denote the set of stored mechanisms at iteration t. A mechanism m_i is stored only if it satisfies the execution criterion $\mathcal{V}(m_i) = 1$. Retrieval follows $m_{\text{retrieved}} = \arg\min_{m_i \in M_t} d(m_i, m_{\text{target}})$, where $d(\cdot, \cdot)$ is a predefined distance metric. Memory consists exclusively of mechanisms that executed successfully, which preserves validity across the

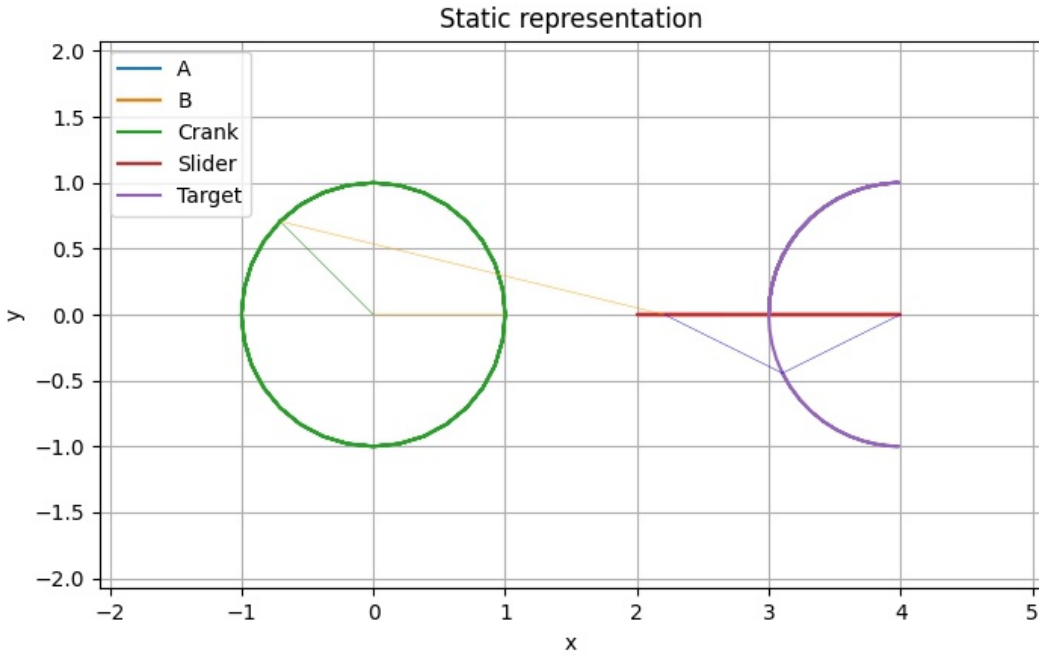

Figure 12: Generated Parabola: the Parabola synthesised by our agent.

repository, eliminates invalid designs from future context, and favours convergence by retaining proven solutions.

Retrieval methodology. Memory retrieval implements a proximity-based selection strategy: (1) mechanisms are prioritised by closeness to the current design objectives; (2) when multiple examples are required, the top-k closest mechanisms are returned.

Progressive specialisation. As iterations proceed, the memory accumulates designs that successfully navigate the specific constraints of the current problem. Early iterations contribute generic mechanisms satisfying basic kinematic requirements; later iterations add designs that handle edge cases, optimise for specific trajectory features, or employ domain-specific tricks discovered through the critique feedback loop.

Pattern transfer. The retrieval mechanism ($\arg\min_{m_i \in M_t} d(m_i, m_{\text{target}})$) ensures that the LLM encounters designs geometrically similar to current objectives — e.g. for a quarter-circle trajectory, it retrieves past designs for circular or arc-shaped paths — enabling transfer of geometric patterns (e.g. link-length ratios for a given curvature) without explicit feature engineering.

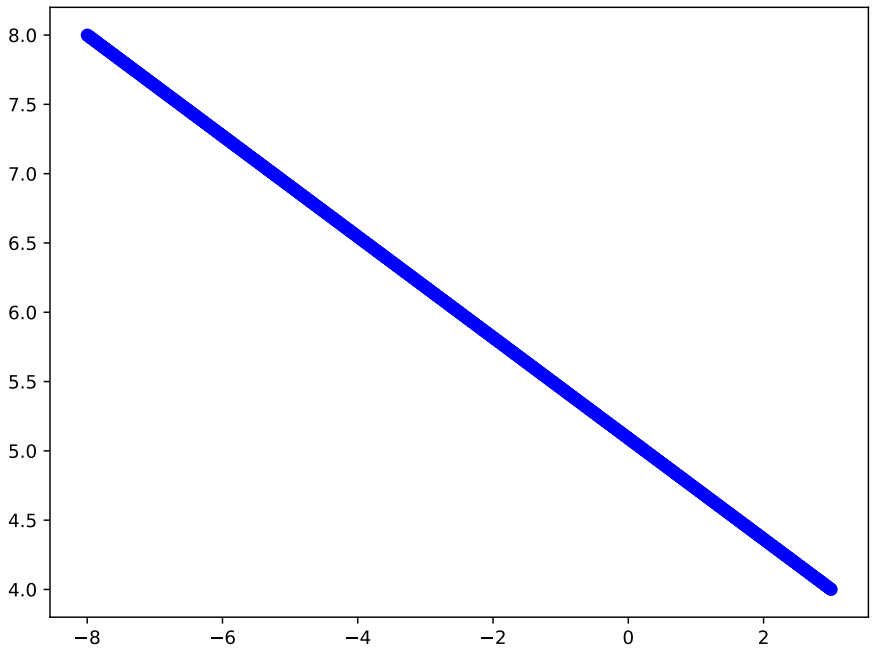

Figure 13: Ground truth Line: the original target Line used for evaluation.

Failure avoidance. The selective storage criterion ($\mathcal{V}(m_i) = 1$) prevents the LLM from re-exploring regions that violate physical constraints (disconnected link chains, impossible joint configurations), implicitly learning the valid subspace of the design space.

Formal learning objective. Memory-augmented learning can be cast as minimising expected trajectory error over the task distribution: $\mathbb{E}_{\mathcal{T}\sim p(\text{tasks})}[\min_{t\leq T} d(\mathcal{M}_t, \mathcal{T}) \mid M_t]$, where $\mathcal{M}_t$ is the best mechanism at iteration t and M_t the memory state. Memory acts as a sufficient statistic for task-relevant knowledge, yielding lower expected error with fewer iterations as $|M_t|$ grows.

The paradigm differs from supervised learning (no ground-truth labels), reinforcement learning (no learnable policy parameters), and evolutionary algorithms (no population-based crossover): it is a form of *linguistic program synthesis with geometric grounding* in which memory retrieval and symbolic feedback guide systematic exploration without parameter updates.

V Broader connections

The symbolic intermediary connects to several active research threads, and this section records the deltas against each.

Paradigms of adaptation without parameter updates. The loop of §2.3 occupies a distinctive position. It departs from standard *in-context learning* (Brown et al., 2020; Min et al., 2022; Garg et al., 2022) in that its examples are generated dynamically through simulation interaction instead of being fixed in advance, each iteration adding design–outcome pairs that reshape the context. It departs from *meta-learning* (**?**) in having no outer-loop optimisation over a task distribution, with adaptation coming from accumulated memory and critique. It departs from *heuristic search* in that the operators are not fixed, since the model adapts its proposals from the symbolic analysis of past failures.

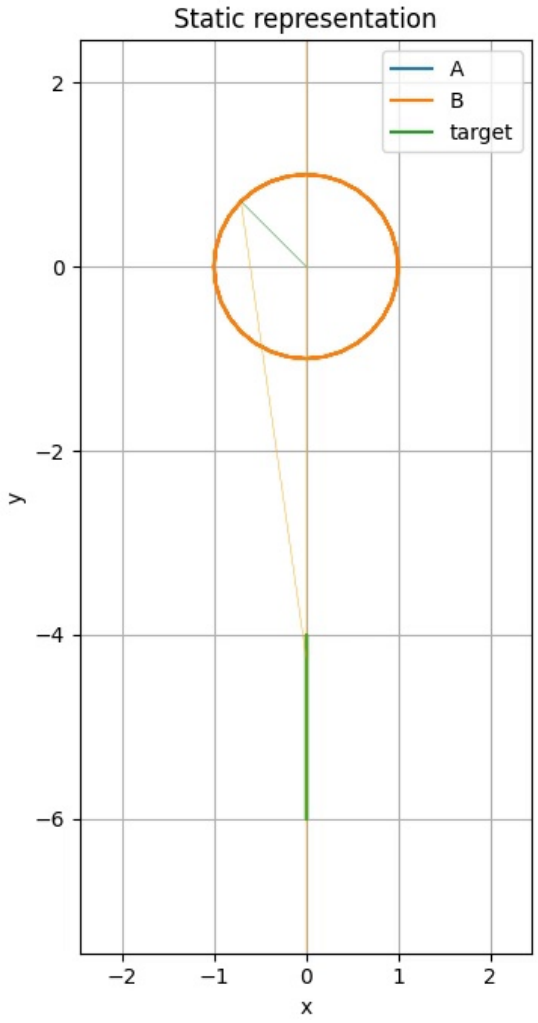

Figure 14: Generated Line: the Line synthesised by our agent.

Methodological connections. The construction extends the *tool-augmented language model* paradigm by showing that tools, here a simulator and symbolic regression, can serve as components that translate between representational domains and not only as information sources, which expands the class of problems amenable to linguistic reasoning. It instantiates a form of *neuro-symbolic integration* in which the symbolic component is discovered from data instead of hard-coded, giving a layer that adapts to each design problem. Its memory relates to *retrieval-augmented generation* (Lewis et al., 2020) with one difference: the retrieved content is dynamically accumulated problem-solving experience instead of static knowledge, which makes the method a form of experience-augmented reasoning.

Future work: other domains. Instantiating the same cycle elsewhere requires a simulator that returns a numeric trace and a target with an analytic description. Four domains meet those two conditions on paper and would be natural first tests: molecular geometry optimisation, where bond lengths, angles and dihedral angles define continuous configurations a simulator produces numerically; robotic trajectory planning, where end-effector paths through 3D space bear direct structural similarity to the 2D trajectories studied here; structural topology optimisation, where stress and deformation fields from FEM solvers admit symbolic surrogates; and metamaterial design, where unit-cell geometry determines continuous wave-propagation properties. We leave that question to future work.

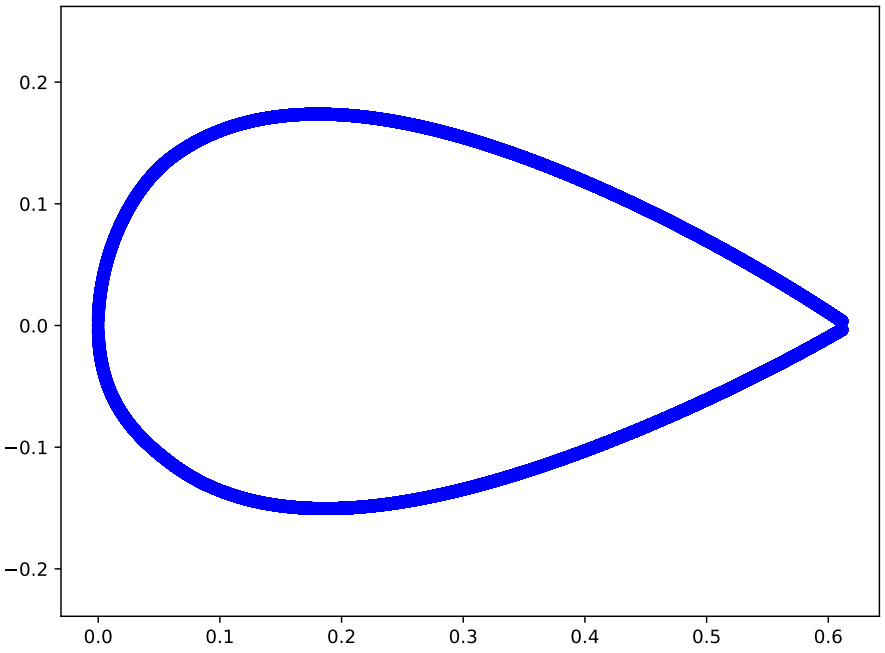

Figure 15: Ground truth NACA airfoil: the original target NACA airfoil used for evaluation.

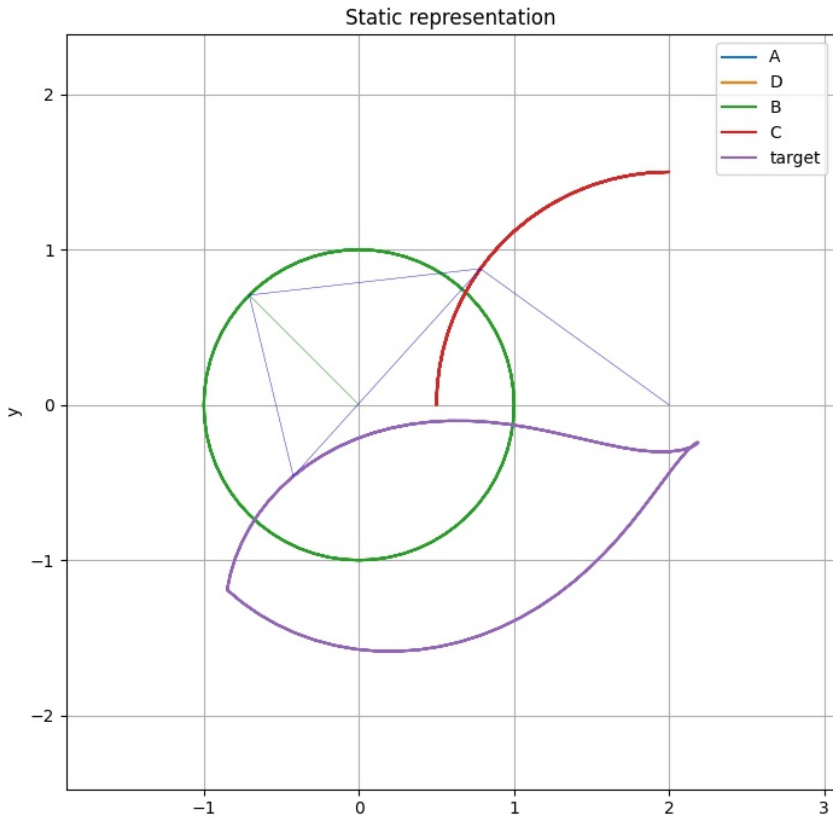

Figure 16: Generated NACA airfoil: the NACA airfoil synthesised by our agent.

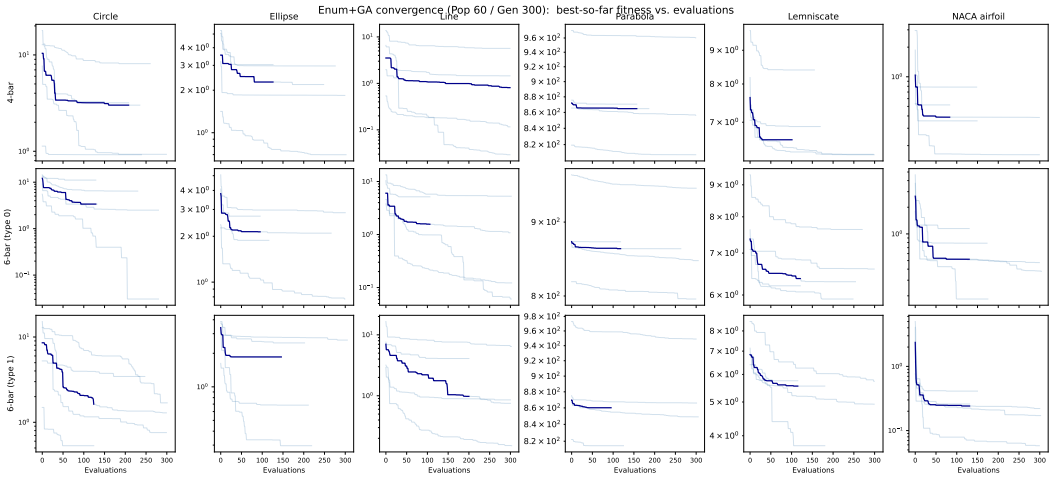

Figure 21: Enum+GA convergence curves (Pop 60 / Gen 300). Best-so-far Chamfer distance (log scale) vs. number of evaluations for three linkage topologies (rows) and six shape classes (columns). Thin blue lines: individual runs; bold line: median across 5 runs. All conditions plateau, confirming GA convergence.

Baseline	SFB	Model	Value	Δ (vs Base)	Imp. %
4-bar Pop: 3 / Gen: 20	×	Gemma	**0.818**	−0.182 ↑	+18.2% ↑
		LLaMA	**0.461**	−0.539 ↑	+53.9% ↑
		Qwen	**0.582**	−0.418 ↑	+41.8% ↑
	✓	Gemma	**0.804**	−0.196 ↑	+19.6% ↑
		LLaMA	**0.467**	−0.533 ↑	+53.3% ↑
		Qwen	**0.475**	−0.525 ↑	+52.5% ↑
4-bar Pop: 6 / Gen: 20	×	Gemma	**1.219**	+0.219 ↓	−21.9% ↓
		LLaMA	**0.634**	−0.366 ↑	+36.6% ↑
		Qwen	**0.845**	−0.155 ↑	+15.5% ↑
	✓	Gemma	**1.185**	+0.185 ↓	−18.5% ↓
		LLaMA	**0.659**	−0.341 ↑	+34.1% ↑
		Qwen	**0.688**	−0.312 ↑	+31.2% ↑
4-bar Pop: 60 / Gen: 300	×	Gemma	**3.043**	+2.043 ↓	−204.3% ↓
		LLaMA	**1.247**	+0.247 ↓	−24.7% ↓
		Qwen	**2.097**	+1.097 ↓	−109.7% ↓
	✓	Gemma	**3.072**	+2.072 ↓	−207.2% ↓
		LLaMA	**1.351**	+0.351 ↓	−35.1% ↓
		Qwen	**1.320**	+0.320 ↓	−32.0% ↓
6-bar Pop: 3 / Gen: 20	×	Gemma	**0.980**	−0.020 ↑	+2.0% ↑
		LLaMA	**0.558**	−0.442 ↑	+44.2% ↑
		Qwen	**0.710**	−0.290 ↑	+29.0% ↑
	✓	Gemma	**0.955**	−0.045 ↑	+4.5% ↑
		LLaMA	**0.565**	−0.435 ↑	+43.5% ↑
		Qwen	**0.601**	−0.399 ↑	+39.9% ↑
6-bar Pop: 6 / Gen: 20	×	Gemma	**1.154**	+0.154 ↓	−15.4% ↓
		LLaMA	**0.634**	−0.366 ↑	+36.6% ↑
		Qwen	**0.829**	−0.171 ↑	+17.1% ↑
	✓	Gemma	**1.112**	+0.112 ↓	−11.2% ↓
		LLaMA	**0.648**	−0.352 ↑	+35.2% ↑
		Qwen	**0.698**	−0.302 ↑	+30.2% ↑
6-bar Pop: 60 / Gen: 300	×	Gemma	**3.039**	+2.039 ↓	−203.9% ↓
		LLaMA	**1.290**	+0.290 ↓	−29.0% ↓
		Qwen	**2.088**	+1.088 ↓	−108.8% ↓
	✓	Gemma	**3.092**	+2.092 ↓	−209.2% ↓
		LLaMA	**1.367**	+0.367 ↓	−36.7% ↓
		Qwen	**1.420**	+0.420 ↓	−42.0% ↓

Table 5: Comparison of models vs. GA baselines (Feedback: True). SFB ✓ = symbolic feedback enabled; × = disabled. Δ and Imp. % are relative to the corresponding GA baseline. Teal = improvement over baseline; red = degradation.

Model	Shape	#Ex	Fdbk	SFB	Mem	Best chamf.	Fcham	Steps	Fstep	% Imp.	% Semantic
Gemma	LB	2	No	No	0	8.847 ± 0.493	9.078 ± 0.720	7.333 ± 2.963	7.000 ± 3.215	2.203	0.101
	LB	2	No	No	2	8.893 ± 0.851	8.893 ± 0.851	9.000 ± 4.163	5.000 ± 3.055	0.000	0.082
	LB	2	No	Yes	2	8.474 ± 1.036	8.474 ± 1.036	9.667 ± 4.055	9.667 ± 4.055	0.000	0.168
	LB	2	Yes	No	0	8.700 ± 0.375	9.043 ± 0.395	11.000 ± 2.608	6.400 ± 1.600	3.731	0.076
	LB	2	Yes	No	2	9.132 ± 0.853	9.965 ± 0.772	14.400 ± 1.720	9.600 ± 2.315	8.171	0.423
	LB	2	Yes	Yes	2	7.982 ± 0.729	9.374 ± 0.444	18.400 ± 0.245	6.200 ± 1.497	15.422	0.492
	LB	3	No	No	0	10.767 ± 0.985	11.951 ± 1.432	15.000 ± 0.577	11.000 ± 4.509	8.424	0.044
	LB	3	No	No	2	10.529 ± 0.675	10.529 ± 0.675	5.600 ± 1.720	5.600 ± 1.720	0.000	0.103
	LB	3	No	Yes	2	13.662 ± 1.679	14.552 ± 1.257	10.333 ± 4.807	5.667 ± 3.712	6.332	0.143
	LB	3	Yes	No	0	8.916 ± 0.327	10.266 ± 1.146	8.200 ± 1.744	4.400 ± 1.833	10.763	0.079
	LB	3	Yes	No	2	9.236 ± 0.561	9.612 ± 0.533	15.400 ± 1.691	6.600 ± 3.265	3.883	0.483
	LB	3	Yes	Yes	2	9.021 ± 0.656	9.763 ± 0.593	16.200 ± 2.577	6.000 ± 3.082	7.682	0.545
	Circle	2	No	No	0	8.490 ± 2.885	8.503 ± 2.874	13.750 ± 2.720	11.500 ± 2.754	1.059	0.042
	Circle	2	No	No	2	11.302 ± 3.330	11.356 ± 3.288	7.800 ± 2.010	7.400 ± 2.227	4.829	0.208
	Circle	2	No	Yes	2	8.227 ± 3.201	8.242 ± 3.189	13.750 ± 3.276	13.000 ± 3.109	1.643	0.089
	Circle	2	Yes	No	0	11.924 ± 3.486	14.248 ± 3.615	10.400 ± 2.977	5.800 ± 2.035	14.806	0.053
	Circle	2	Yes	No	2	10.899 ± 3.318	12.040 ± 3.422	15.200 ± 1.881	7.600 ± 1.939	9.651	0.527
	Circle	2	Yes	Yes	2	9.919 ± 3.420	11.597 ± 3.365	10.600 ± 2.768	1.600 ± 0.400	23.521	0.649
	Circle	3	No	No	0	11.490 ± 2.669	11.509 ± 2.654	6.000 ± 1.225	4.200 ± 1.068	0.611	0.050
	Circle	3	No	No	2	9.019 ± 2.241	11.203 ± 3.348	11.400 ± 2.768	11.000 ± 2.950	10.285	0.162
	Circle	3	No	Yes	2	6.288 ± 1.658	8.826 ± 2.147	13.600 ± 2.638	10.000 ± 2.387	20.266	0.187
	Circle	3	Yes	No	0	10.074 ± 2.260	11.837 ± 3.345	11.600 ± 2.159	8.000 ± 3.178	9.397	0.076
	Circle	3	Yes	No	2	7.351 ± 1.974	11.817 ± 3.059	13.000 ± 3.050	3.800 ± 1.356	29.604	0.595
	Circle	3	Yes	Yes	2	10.416 ± 3.413	11.744 ± 3.908	14.200 ± 2.289	3.400 ± 1.030	7.921	0.519
	Ellipse	2	No	No	0	6.677 ± 1.335	6.677 ± 1.335	9.000 ± 4.726	9.000 ± 4.726	0.000	0.029
	Ellipse	2	No	No	2	6.945 ± 2.819	6.945 ± 2.819	11.000 ± 9.000	11.000 ± 9.000	0.000	0.026
	Ellipse	2	No	Yes	2	6.568 ± 1.315	6.568 ± 1.315	3.667 ± 1.764	3.667 ± 1.764	0.000	0.176
	Ellipse	2	Yes	No	0	7.985 ± 1.375	8.928 ± 1.461	10.800 ± 1.828	7.400 ± 1.778	10.527	0.039
	Ellipse	2	Yes	No	2	6.355 ± 1.001	6.714 ± 1.133	11.600 ± 2.786	8.800 ± 2.596	4.053	0.442
	Ellipse	2	Yes	Yes	2	6.836 ± 0.991	7.348 ± 1.079	12.000 ± 2.429	10.400 ± 3.234	6.179	0.218
	Ellipse	3	No	No	0	6.722 ± 1.353	8.009 ± 1.415	6.800 ± 2.498	4.000 ± 1.265	17.586	0.211
	Ellipse	3	No	No	2	7.678 ± 1.267	7.678 ± 1.267	7.800 ± 2.709	7.800 ± 2.709	0.000	0.213
	Ellipse	3	No	Yes	2	7.372 ± 1.454	7.659 ± 1.276	7.600 ± 2.135	5.600 ± 2.272	6.428	0.371
	Ellipse	3	Yes	No	0	6.900 ± 1.443	7.777 ± 1.281	12.600 ± 2.542	3.800 ± 2.332	14.609	0.121
	Ellipse	3	Yes	No	2	6.647 ± 1.244	7.445 ± 1.396	13.600 ± 3.027	6.400 ± 2.713	11.320	0.472
	Ellipse	3	Yes	Yes	2	5.552 ± 0.819	6.665 ± 1.103	12.000 ± 4.324	2.000 ± 0.775	15.391	0.394

Model	Shape	#Ex	Fdbk	SFB	Mem	Best chamf.	Fcham	Steps	Fstep	% Imp.	% Semantic
Gemma	Line	2	No	No	0	13.640 ± 0.322	13.640 ± 0.322	11.000 ± 3.000	11.000 ± 3.000	0.000	0.002
	Line	2	No	No	2	13.643 ± 1.385	13.643 ± 1.385	10.667 ± 4.096	10.667 ± 4.096	0.000	0.136
	Line	2	No	Yes	2	12.076 ± 0.947	14.277 ± 1.254	8.500 ± 3.500	7.500 ± 2.500	14.172	0.778
	Line	2	Yes	No	0	11.150 ± 1.542	11.664 ± 1.139	10.400 ± 3.234	9.800 ± 3.484	6.050	0.032
	Line	2	Yes	No	2	10.827 ± 1.545	11.298 ± 1.214	12.200 ± 1.530	8.800 ± 2.059	6.025	0.241
	Line	2	Yes	Yes	2	10.186 ± 2.444	10.186 ± 2.444	10.600 ± 2.750	10.000 ± 2.449	0.000	0.151
	Line	3	No	No	0	8.619 ± 3.041	9.501 ± 3.392	6.750 ± 4.423	6.500 ± 4.518	6.019	0.038
	Line	3	No	No	2	11.917 ± 1.188	11.917 ± 1.188	9.000 ± 3.055	9.000 ± 3.055	0.000	0.221
	Line	3	No	Yes	2	12.071 ± 0.942	12.071 ± 0.942	4.500 ± 1.500	4.500 ± 1.500	0.000	0.319
	Line	3	Yes	No	0	9.715 ± 1.258	10.689 ± 0.921	6.800 ± 1.393	3.600 ± 1.208	10.022	0.056
	Line	3	Yes	No	2	10.043 ± 1.210	10.835 ± 1.468	10.200 ± 2.518	6.000 ± 2.387	6.166	0.552
	Line	3	Yes	Yes	2	11.035 ± 0.986	12.417 ± 1.243	14.800 ± 2.939	5.600 ± 1.806	10.570	0.371
	Naca	2	No	No	0	1.017 ± 0.436	1.017 ± 0.436	7.000 ± 1.000	7.000 ± 1.000	0.000	0.058
	Naca	2	No	No	2	1.104 ± 0.151	1.104 ± 0.151	6.333 ± 0.882	6.333 ± 0.882	0.000	0.041
	Naca	2	No	Yes	2	1.057 ± 0.416	1.057 ± 0.416	7.000 ± 0.000	7.000 ± 0.000	0.000	0.026
	Naca	2	Yes	No	0	1.082 ± 0.416	1.336 ± 0.306	12.600 ± 2.040	7.800 ± 1.715	27.086	0.078
	Naca	2	Yes	No	2	1.618 ± 0.555	1.714 ± 0.508	16.400 ± 1.364	11.800 ± 3.611	11.084	0.313
	Naca	2	Yes	Yes	2	1.074 ± 0.371	1.663 ± 0.523	12.000 ± 2.470	8.200 ± 2.709	30.463	0.522
	Naca	3	No	No	0	1.498 ± 0.499	1.628 ± 0.427	7.600 ± 2.694	4.600 ± 1.913	15.066	0.159
	Naca	3	No	No	2	1.245 ± 0.533	1.245 ± 0.533	7.000 ± 3.507	7.000 ± 3.507	0.000	0.209
	Naca	3	No	Yes	2	1.687 ± 0.518	1.883 ± 0.544	8.000 ± 1.975	5.200 ± 1.685	18.053	0.328
	Naca	3	Yes	No	0	1.347 ± 0.404	2.381 ± 0.462	6.600 ± 1.400	3.200 ± 1.428	46.926	0.146
	Naca	3	Yes	No	2	0.997 ± 0.307	2.021 ± 0.512	18.200 ± 0.860	5.000 ± 1.897	49.954	0.566
	Naca	3	Yes	Yes	2	0.796 ± 0.204	1.702 ± 0.439	13.800 ± 2.458	2.000 ± 0.548	42.740	0.854
	Parabola	2	No	No	0	869.205 ± 35.569	869.205 ± 35.569	9.500 ± 1.500	9.500 ± 1.500	0.000	0.036
	Parabola	2	No	No	2	861.442 ± 28.839	861.442 ± 28.839	6.000 ± 4.000	6.000 ± 4.000	0.000	0.352
	Parabola	2	No	Yes	2	906.927 ± 50.681	906.927 ± 50.681	10.667 ± 3.712	10.667 ± 3.712	0.000	0.019
	Parabola	2	Yes	No	0	890.160 ± 23.248	890.612 ± 23.162	13.800 ± 3.338	10.800 ± 3.513	0.052	0.023
	Parabola	2	Yes	No	2	892.928 ± 23.123	897.871 ± 27.055	15.000 ± 2.168	6.000 ± 2.145	0.502	0.304
	Parabola	2	Yes	Yes	2	858.509 ± 40.475	892.209 ± 23.039	12.600 ± 2.159	8.200 ± 1.744	3.784	0.300
	Parabola	3	No	No	0	880.197 ± 16.446	881.267 ± 16.420	8.600 ± 1.631	5.200 ± 0.860	0.121	0.073
	Parabola	3	No	No	2	897.018 ± 33.711	898.569 ± 33.387	7.600 ± 2.839	5.200 ± 2.223	0.177	0.505
	Parabola	3	No	Yes	2	871.551 ± 20.077	871.551 ± 20.077	5.000 ± 3.050	5.000 ± 3.050	0.000	0.497
	Parabola	3	Yes	No	0	889.653 ± 22.191	896.919 ± 24.107	11.600 ± 2.768	6.400 ± 1.288	0.788	0.072
	Parabola	3	Yes	No	2	895.262 ± 24.071	905.689 ± 33.476	10.200 ± 2.818	7.200 ± 2.746	1.013	0.396
	Parabola	3	Yes	Yes	2	892.568 ± 23.659	894.953 ± 22.691	10.400 ± 2.750	5.800 ± 2.354	0.278	0.513

Model	Shape	#Ex	Fdbk	SFB	Mem	Best chamf.	Fcham	Steps	Fstep	% Imp.	% Semantic
Llama	LB	2	No	No	0	7.415 ± 0.864	14.752 ± 2.231	14.750 ± 2.136	1.500 ± 0.289	45.126	0.231
	LB	2	No	No	2	7.730 ± 0.445	10.356 ± 1.944	3.600 ± 1.631	1.400 ± 0.245	18.120	0.786
	LB	2	No	Yes	2	8.264 ± 0.759	11.495 ± 1.094	5.800 ± 2.615	1.800 ± 0.490	25.670	0.816
	LB	2	Yes	No	0	7.029 ± 0.825	12.083 ± 1.603	10.400 ± 3.219	1.000 ± 0.000	35.867	0.375
	LB	2	Yes	No	2	7.539 ± 0.638	12.458 ± 2.201	9.250 ± 2.810	1.750 ± 0.479	34.418	0.899
	LB	2	Yes	Yes	2	7.986 ± 0.701	16.445 ± 2.855	11.000 ± 3.416	1.250 ± 0.250	45.705	0.801
	LB	3	No	No	0	7.861 ± 0.522	15.193 ± 2.333	10.600 ± 3.709	1.400 ± 0.245	43.505	0.324
	LB	3	No	No	2	10.945 ± 1.811	12.727 ± 3.415	3.750 ± 2.136	1.250 ± 0.250	8.388	0.796
	LB	3	No	Yes	2	11.656 ± 1.987	16.404 ± 2.378	4.600 ± 1.806	1.800 ± 0.583	25.118	0.576
	LB	3	Yes	No	0	6.610 ± 1.389	17.877 ± 3.803	14.667 ± 2.728	1.000 ± 0.000	57.215	0.259
	LB	3	Yes	No	2	8.017 ± 1.204	10.287 ± 1.723	4.500 ± 3.500	1.000 ± 0.000	21.837	0.969
	LB	3	Yes	Yes	2	7.337 ± 0.212	10.209 ± 1.356	17.333 ± 1.764	1.000 ± 0.000	25.852	0.855
	Circle	2	No	No	0	2.673 ± 0.714	12.537 ± 3.937	8.800 ± 3.023	2.200 ± 0.583	77.134	0.192
	Circle	2	No	No	2	11.047 ± 3.018	11.119 ± 2.962	4.000 ± 1.291	3.500 ± 1.500	1.730	0.550
	Circle	2	No	Yes	2	7.901 ± 3.435	11.478 ± 2.628	14.200 ± 2.518	6.200 ± 1.855	40.320	0.632
	Circle	2	Yes	No	0	1.466 ± 0.659	12.239 ± 3.669	17.000 ± 1.225	1.500 ± 0.289	90.609	0.238
	Circle	2	Yes	No	2	5.482 ± 2.233	12.932 ± 4.846	11.500 ± 3.069	1.750 ± 0.750	52.205	0.758
	Circle	2	Yes	Yes	2	1.930 ± 0.428	8.044 ± 3.258	17.500 ± 0.645	4.750 ± 3.119	58.285	0.676
	Circle	3	No	No	0	4.011 ± 2.143	6.372 ± 2.802	7.400 ± 3.385	2.000 ± 0.447	29.322	0.170
	Circle	3	No	No	2	7.237 ± 6.154	9.361 ± 7.705	9.667 ± 3.712	4.000 ± 3.000	22.474	0.723
	Circle	3	No	Yes	2	6.334 ± 3.305	11.281 ± 3.148	11.400 ± 2.159	2.400 ± 0.678	54.314	0.646
	Circle	3	Yes	No	0	2.181 ± 0.848	10.715 ± 4.386	7.200 ± 2.223	1.200 ± 0.200	73.424	0.387
	Circle	3	Yes	No	2	4.097 ± 2.085	11.403 ± 3.930	18.200 ± 1.562	1.400 ± 0.400	60.651	0.702
	Circle	3	Yes	Yes	2	5.660 ± 2.889	10.780 ± 3.395	11.600 ± 2.249	1.800 ± 0.800	54.820	0.826
	Ellipse	2	No	No	0	4.218 ± 0.505	7.903 ± 2.433	9.600 ± 2.542	1.600 ± 0.400	30.905	0.164
	Ellipse	2	No	No	2	4.523 ± 0.673	7.153 ± 1.232	10.200 ± 2.672	2.000 ± 0.316	34.718	0.837
	Ellipse	2	No	Yes	2	4.599 ± 0.977	6.798 ± 1.167	10.200 ± 1.828	2.400 ± 0.510	30.822	0.731
	Ellipse	2	Yes	No	0	4.113 ± 0.590	8.028 ± 1.123	14.400 ± 1.600	1.800 ± 0.374	45.818	0.252
	Ellipse	2	Yes	No	2	3.383 ± 0.391	6.356 ± 0.877	17.000 ± 1.732	2.750 ± 1.436	43.673	0.862
	Ellipse	2	Yes	Yes	2	3.371 ± 0.906	10.413 ± 5.775	12.000 ± 1.826	2.750 ± 1.750	52.252	0.737
	Ellipse	3	No	No	0	3.801 ± 0.663	8.619 ± 1.584	9.600 ± 2.713	2.000 ± 0.548	52.293	0.217
	Ellipse	3	No	No	2	5.523 ± 0.598	11.445 ± 2.579	3.800 ± 0.860	2.000 ± 0.632	43.739	0.826
	Ellipse	3	No	Yes	2	4.759 ± 0.820	7.138 ± 0.904	15.200 ± 2.267	2.600 ± 1.166	31.002	0.633
	Ellipse	3	Yes	No	0	4.048 ± 0.798	7.090 ± 1.704	4.000 ± 1.683	1.250 ± 0.250	41.283	0.303
	Ellipse	3	Yes	No	2	2.935 ± 0.621	8.870 ± 4.465	11.333 ± 1.202	2.333 ± 0.882	54.619	0.606
	Ellipse	3	Yes	Yes	2	3.429 ± 0.496	7.536 ± 2.523	8.667 ± 4.702	2.000 ± 0.577	41.305	0.785

Model	Shape	#Ex	Fdbk	SFB	Mem	Best chamf.	Fcham	Steps	Fstep	% Imp.	% Semantic
Llama	Line	2	No	No	0	3.479 ± 1.495	10.273 ± 1.666	12.400 ± 1.939	3.000 ± 0.894	65.454	0.113
	Line	2	No	No	2	5.999 ± 1.191	9.055 ± 1.468	9.200 ± 3.023	4.800 ± 2.577	30.440	0.541
	Line	2	No	Yes	2	5.089 ± 1.962	6.117 ± 1.809	11.000 ± 2.074	7.000 ± 2.121	18.628	0.536
	Line	2	Yes	No	0	2.935 ± 1.658	8.540 ± 2.595	8.000 ± 1.581	1.750 ± 0.750	66.011	0.138
	Line	2	Yes	No	2	2.024 ± 0.803	7.411 ± 1.205	9.750 ± 3.660	1.750 ± 0.750	62.358	0.653
	Line	2	Yes	Yes	2	4.203 ± 2.095	11.977 ± 1.247	16.000 ± 1.761	1.200 ± 0.200	66.938	0.930
	Line	3	No	No	0	1.892 ± 1.151	6.488 ± 1.696	13.000 ± 3.000	1.600 ± 0.400	70.869	0.167
	Line	3	No	No	2	2.824 ± 1.109	7.774 ± 2.180	5.000 ± 1.080	2.000 ± 0.408	47.051	0.794
	Line	3	No	Yes	2	6.741 ± 2.275	9.788 ± 1.316	9.000 ± 1.871	2.600 ± 0.980	35.589	0.661
	Line	3	Yes	No	0	0.904 ± 0.402	11.338 ± 3.167	7.250 ± 1.109	1.000 ± 0.000	89.959	0.178
	Line	3	Yes	No	2	3.281 ± 1.334	9.965 ± 3.968	14.000 ± 1.155	1.667 ± 0.333	51.229	0.675
	Line	3	Yes	Yes	2	3.090 ± 1.299	10.357 ± 2.300	14.000 ± 2.588	2.400 ± 0.600	68.001	0.758
	Naca	2	No	No	0	1.555 ± 0.635	2.577 ± 0.494	11.600 ± 3.043	5.400 ± 3.655	38.687	0.157
	Naca	2	No	No	2	1.247 ± 0.444	2.166 ± 0.604	4.000 ± 1.049	2.800 ± 1.114	46.449	0.624
	Naca	2	No	Yes	2	1.597 ± 0.532	1.714 ± 0.472	5.600 ± 3.187	4.800 ± 3.323	12.779	0.588
	Naca	2	Yes	No	0	0.624 ± 0.182	1.722 ± 0.494	10.400 ± 3.763	1.600 ± 0.400	57.032	0.309
	Naca	2	Yes	No	2	0.655 ± 0.209	2.469 ± 0.303	13.750 ± 1.652	1.500 ± 0.289	70.067	0.873
	Naca	2	Yes	Yes	2	0.922 ± 0.313	1.614 ± 0.701	9.667 ± 5.175	1.000 ± 0.000	29.433	0.888
	Naca	3	No	No	0	1.157 ± 0.373	2.021 ± 0.794	9.400 ± 2.542	1.400 ± 0.245	31.172	0.194
	Naca	3	No	No	2	1.315 ± 0.367	1.643 ± 0.630	4.250 ± 2.016	2.000 ± 0.707	11.631	0.828
	Naca	3	No	Yes	2	1.168 ± 0.367	2.324 ± 0.740	10.600 ± 2.337	4.600 ± 2.619	43.976	0.564
	Naca	3	Yes	No	0	0.943 ± 0.419	2.576 ± 0.817	5.750 ± 2.496	1.750 ± 0.479	66.220	0.381
	Naca	3	Yes	No	2	1.456 ± 0.571	2.291 ± 0.235	10.333 ± 4.631	2.000 ± 1.000	38.924	0.675
	Naca	3	Yes	Yes	2	1.016 ± 0.427	1.939 ± 0.472	17.500 ± 1.658	1.250 ± 0.250	49.899	0.776
	Parabola	2	No	No	0	871.702 ± 29.579	892.003 ± 24.567	13.400 ± 3.234	2.000 ± 0.548	2.319	0.224
	Parabola	2	No	No	2	872.853 ± 34.498	894.611 ± 30.983	9.800 ± 3.625	2.600 ± 0.748	2.478	0.748
	Parabola	2	No	Yes	2	851.307 ± 33.603	909.159 ± 20.136	8.000 ± 2.588	3.400 ± 2.159	6.395	0.775
	Parabola	2	Yes	No	0	866.615 ± 32.053	1125.509 ± 211.739	11.600 ± 2.379	1.400 ± 0.245	14.196	0.302
	Parabola	2	Yes	No	2	817.244 ± 83.559	900.522 ± 23.241	12.000 ± 2.775	1.200 ± 0.200	9.727	0.827
	Parabola	2	Yes	Yes	2	815.179 ± 84.612	896.679 ± 26.440	10.200 ± 2.764	1.200 ± 0.200	9.732	0.836
	Parabola	3	No	No	0	838.519 ± 33.608	879.870 ± 18.275	16.800 ± 1.985	3.200 ± 0.583	4.761	0.204
	Parabola	3	No	No	2	845.166 ± 45.083	858.242 ± 41.509	5.600 ± 1.122	2.000 ± 0.775	1.619	0.731
	Parabola	3	No	Yes	2	843.717 ± 36.534	902.770 ± 49.407	11.600 ± 2.977	1.200 ± 0.200	5.900	0.787
	Parabola	3	Yes	No	0	838.254 ± 37.670	896.905 ± 23.839	14.400 ± 2.064	1.600 ± 0.400	6.597	0.294
	Parabola	3	Yes	No	2	843.277 ± 49.965	898.882 ± 26.610	11.200 ± 2.223	1.000 ± 0.000	6.205	0.820
	Parabola	3	Yes	Yes	2	848.027 ± 35.591	898.265 ± 24.218	16.600 ± 0.678	1.000 ± 0.000	5.670	0.872

Model	Shape	#Ex	Fdbk	SFB	Mem	Best chamf.	Fcham	Steps	Fstep	% Imp.	% Semantic
Qwen	LB	2	No	No	0	8.589 ± 0.319	10.122 ± 0.512	7.400 ± 3.234	1.800 ± 0.374	13.760	0.367
	LB	2	No	No	2	8.530 ± 0.376	12.677 ± 1.963	9.600 ± 3.473	1.000 ± 0.000	26.693	0.512
	LB	2	No	Yes	2	9.129 ± 0.866	10.727 ± 1.141	11.200 ± 2.888	1.800 ± 0.374	13.545	0.611
	LB	2	Yes	No	0	8.047 ± 0.683	10.937 ± 1.311	13.200 ± 1.855	1.400 ± 0.245	25.157	0.288
	LB	2	Yes	No	2	8.763 ± 0.943	11.410 ± 1.425	6.800 ± 2.396	1.200 ± 0.200	22.070	0.471
	LB	2	Yes	Yes	2	7.830 ± 1.112	10.027 ± 0.589	14.000 ± 2.214	1.800 ± 0.374	23.076	0.467
	LB	3	No	No	0	8.460 ± 0.586	9.789 ± 0.486	11.400 ± 3.156	1.200 ± 0.200	13.869	0.381
	LB	3	No	No	2	8.075 ± 0.656	12.422 ± 2.085	15.200 ± 2.888	1.200 ± 0.200	27.637	0.349
	LB	3	No	Yes	2	7.832 ± 0.183	9.717 ± 0.465	12.000 ± 2.588	1.600 ± 0.400	18.843	0.467
	LB	3	Yes	No	0	7.291 ± 0.632	10.550 ± 1.101	10.400 ± 1.122	1.000 ± 0.000	28.501	0.310
	LB	3	Yes	No	2	6.905 ± 1.095	9.901 ± 0.439	11.400 ± 2.657	1.000 ± 0.000	28.431	0.504
	LB	3	Yes	Yes	2	7.804 ± 0.396	10.686 ± 0.758	14.400 ± 2.337	1.000 ± 0.000	25.554	0.638
	Circle	2	No	No	0	11.664 ± 3.720	11.664 ± 3.720	9.667 ± 3.844	9.667 ± 3.844	0.000	0.019
	Circle	2	No	No	2	12.950 ± 2.919	12.950 ± 2.919	12.500 ± 3.379	12.500 ± 3.379	0.000	0.019
	Circle	2	No	Yes	2	11.226 ± 3.332	11.226 ± 3.332	14.333 ± 0.333	14.333 ± 0.333	0.000	0.020
	Circle	2	Yes	No	0	10.478 ± 4.392	10.478 ± 4.392	11.200 ± 1.855	11.200 ± 1.855	0.000	0.011
	Circle	2	Yes	No	2	3.139 ± 1.309	6.518 ± 2.842	7.600 ± 2.462	7.400 ± 2.337	19.826	0.087
	Circle	2	Yes	Yes	2	5.880 ± 5.585	5.880 ± 5.585	10.667 ± 1.667	10.667 ± 1.667	0.000	0.013
	Circle	3	No	No	0	7.045 ± 4.241	8.420 ± 3.664	14.750 ± 3.276	12.250 ± 3.146	24.987	0.029
	Circle	3	No	No	2	9.131 ± 4.288	9.948 ± 3.936	12.000 ± 2.191	11.000 ± 1.817	18.175	0.059
	Circle	3	No	Yes	2	5.770 ± 5.360	5.770 ± 5.360	12.500 ± 1.500	12.500 ± 1.500	0.000	0.018
	Circle	3	Yes	No	0	5.563 ± 3.546	6.643 ± 3.277	10.200 ± 3.569	7.600 ± 3.816	19.995	0.025
	Circle	3	Yes	No	2	4.173 ± 2.654	8.734 ± 3.423	10.750 ± 2.496	5.000 ± 2.121	49.018	0.123
	Circle	3	Yes	Yes	2	6.200 ± 3.143	6.316 ± 3.087	11.600 ± 2.804	9.000 ± 1.924	19.710	0.084
	Ellipse	2	No	No	0	6.010 ± 0.958	9.740 ± 3.056	8.200 ± 2.653	1.400 ± 0.400	25.591	0.414
	Ellipse	2	No	No	2	5.166 ± 0.860	7.507 ± 1.320	12.600 ± 3.544	1.400 ± 0.400	29.698	0.562
	Ellipse	2	No	Yes	2	5.746 ± 0.928	6.984 ± 0.969	11.000 ± 3.302	1.600 ± 0.400	17.681	0.391
	Ellipse	2	Yes	No	0	4.794 ± 0.816	6.770 ± 1.179	8.800 ± 2.939	1.200 ± 0.200	24.563	0.283
	Ellipse	2	Yes	No	2	5.279 ± 0.905	8.558 ± 1.023	10.600 ± 2.839	1.000 ± 0.000	36.391	0.420
	Ellipse	2	Yes	Yes	2	4.323 ± 0.885	7.550 ± 1.321	15.200 ± 2.922	1.600 ± 0.600	38.938	0.589
	Ellipse	3	No	No	0	6.026 ± 0.949	6.664 ± 1.092	11.600 ± 4.179	1.200 ± 0.200	8.910	0.363
	Ellipse	3	No	No	2	5.958 ± 1.075	6.648 ± 1.076	5.800 ± 2.746	1.400 ± 0.245	11.515	0.524
	Ellipse	3	No	Yes	2	5.869 ± 0.854	6.886 ± 0.913	9.800 ± 3.153	1.600 ± 0.400	15.468	0.530
	Ellipse	3	Yes	No	0	4.752 ± 0.736	6.780 ± 1.049	8.800 ± 2.800	1.000 ± 0.000	29.121	0.314
	Ellipse	3	Yes	No	2	5.350 ± 1.282	9.062 ± 2.360	10.800 ± 4.030	1.000 ± 0.000	35.413	0.577
	Ellipse	3	Yes	Yes	2	4.562 ± 0.867	7.454 ± 1.236	14.000 ± 2.098	1.000 ± 0.000	35.717	0.554

Model	Shape	#Ex	Fdbk	SFB	Mem	Best chamf.	Fcham	Steps	Fstep	% Imp.	% Semantic
Qwen	Line	2	No	No	0	7.592 ± 1.657	8.548 ± 2.231	11.000 ± 2.720	7.400 ± 3.140	7.364	0.034
	Line	2	No	No	2	7.967 ± 2.032	7.967 ± 2.032	11.000 ± 2.915	11.000 ± 2.915	0.000	0.214
	Line	2	No	Yes	2	5.757 ± 2.652	8.488 ± 2.728	8.200 ± 3.338	8.000 ± 3.162	19.165	0.027
	Line	2	Yes	No	0	7.653 ± 1.741	18.455 ± 8.155	9.800 ± 1.772	6.400 ± 1.208	34.878	0.039
	Line	2	Yes	No	2	4.236 ± 2.421	13.285 ± 3.206	11.000 ± 3.209	5.200 ± 1.356	59.558	0.466
	Line	2	Yes	Yes	2	9.153 ± 7.081	18.403 ± 8.191	9.800 ± 2.245	2.800 ± 0.800	45.798	0.639
	Line	3	No	No	0	7.081 ± 2.225	7.421 ± 2.433	9.400 ± 2.581	9.400 ± 2.581	2.468	0.046
	Line	3	No	No	2	5.088 ± 1.761	11.248 ± 2.739	11.200 ± 2.615	4.800 ± 1.281	37.601	0.590
	Line	3	No	Yes	2	15.003 ± 8.797	16.356 ± 8.538	12.200 ± 1.800	11.200 ± 2.518	10.892	0.137
	Line	3	Yes	No	0	6.913 ± 1.656	15.881 ± 8.458	11.200 ± 3.426	3.200 ± 1.020	35.913	0.058
	Line	3	Yes	No	2	8.924 ± 7.667	14.021 ± 8.520	14.800 ± 1.655	3.400 ± 0.927	41.450	0.551
	Line	3	Yes	Yes	2	2.268 ± 0.919	20.053 ± 9.575	16.200 ± 1.685	5.000 ± 2.121	83.994	0.583
	Naca	2	No	No	0	1.285 ± 0.487	1.888 ± 0.356	11.600 ± 2.421	1.000 ± 0.000	41.059	0.689
	Naca	2	No	No	2	1.297 ± 0.467	1.953 ± 0.408	9.200 ± 2.154	1.000 ± 0.000	40.370	0.833
	Naca	2	No	Yes	2	1.323 ± 0.508	1.818 ± 0.472	7.600 ± 3.187	1.000 ± 0.000	32.664	0.731
	Naca	2	Yes	No	0	1.129 ± 0.476	2.320 ± 0.518	11.800 ± 2.131	1.000 ± 0.000	57.917	0.477
	Naca	2	Yes	No	2	0.887 ± 0.223	1.953 ± 0.408	14.800 ± 2.478	1.000 ± 0.000	55.968	0.553
	Naca	2	Yes	Yes	2	0.889 ± 0.263	2.677 ± 0.586	15.200 ± 2.375	1.000 ± 0.000	63.523	0.602
	Naca	3	No	No	0	1.306 ± 0.473	1.812 ± 0.472	9.600 ± 1.364	1.200 ± 0.200	32.670	0.651
	Naca	3	No	No	2	1.111 ± 0.283	1.893 ± 0.356	9.800 ± 3.216	1.000 ± 0.000	43.237	0.840
	Naca	3	No	Yes	2	1.380 ± 0.393	1.899 ± 0.407	4.600 ± 2.015	1.000 ± 0.000	31.158	0.802
	Naca	3	Yes	No	0	1.007 ± 0.382	2.130 ± 0.404	7.400 ± 1.749	1.000 ± 0.000	55.782	0.530
	Naca	3	Yes	No	2	0.876 ± 0.195	1.985 ± 0.436	13.800 ± 2.800	1.000 ± 0.000	56.462	0.742
	Naca	3	Yes	Yes	2	1.080 ± 0.401	1.980 ± 0.437	13.000 ± 2.214	1.000 ± 0.000	52.116	0.812
	Parabola	2	No	No	0	890.798 ± 24.891	896.765 ± 23.919	7.200 ± 2.746	1.200 ± 0.200	0.672	0.594
	Parabola	2	No	No	2	807.431 ± 18.711	897.205 ± 24.825	13.600 ± 2.135	1.000 ± 0.000	9.804	0.810
	Parabola	2	No	Yes	2	837.704 ± 21.792	897.545 ± 24.517	10.600 ± 3.655	1.200 ± 0.200	6.526	0.723
	Parabola	2	Yes	No	0	847.932 ± 38.709	898.009 ± 24.439	10.000 ± 2.702	1.800 ± 0.583	5.732	0.342
	Parabola	2	Yes	No	2	814.122 ± 39.460	897.491 ± 24.524	9.200 ± 2.922	1.000 ± 0.000	8.900	0.557
	Parabola	2	Yes	Yes	2	754.452 ± 54.486	899.164 ± 26.358	18.200 ± 0.860	1.200 ± 0.200	16.416	0.819
	Parabola	3	No	No	0	888.611 ± 24.141	896.433 ± 23.917	10.800 ± 2.764	1.000 ± 0.000	0.874	0.600
	Parabola	3	No	No	2	848.301 ± 41.711	897.992 ± 24.505	13.400 ± 3.219	1.200 ± 0.200	5.692	0.613
	Parabola	3	No	Yes	2	895.240 ± 25.034	897.025 ± 24.547	4.200 ± 3.200	1.000 ± 0.000	0.205	0.598
	Parabola	3	Yes	No	0	884.636 ± 25.279	897.639 ± 25.247	11.200 ± 1.497	1.200 ± 0.200	1.452	0.430
	Parabola	3	Yes	No	2	818.250 ± 35.861	897.503 ± 24.680	12.600 ± 3.444	1.200 ± 0.200	8.706	0.569
	Parabola	3	Yes	Yes	2	709.656 ± 67.207	897.195 ± 23.691	15.000 ± 2.470	1.000 ± 0.000	21.299	0.840

Table 6: Performance results across shapes and settings, where $\pm$ denotes the standard error.