

Cognitive evolution enables de novo design of untethered millimeter-scale robots

Haiwang Li

lihaiwang@buaa.edu.cn

Beihang University

Weizhi Zhao

Beihang University

Zhi Tao

Beihang University

Peihan Li

Beihang University

Xizhi Wang

Beihang University

Qinghong Wang

Beihang University

Boxian Wang

Beihang University

Yanxin Zhai

Beihang University

Wei Wei

Beijing Institute of Technology

Shitong Wang

Beihang University <https://orcid.org/0000-0003-4341-3510>

Shuangqi Li

The Hong Kong Polytechnic University

Tiantong Xu

Beihang University

Article

Keywords:

Posted Date: October 8th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10770716/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

Abstract

Untethered millimeter-scale robots, which can operate in spaces inaccessible to larger systems, require the integration of actuation, mechanics, and onboard power within an extremely limited volume. For these robots, design optimization is hindered by severe integration constraints, strong cross-system coupling, and physical feasibility that is difficult to formalize. Here, we present EvoAgent, a large-language-model agent for cognitive evolution that uses physical priors, constraint feedback, and design history to propose plausible mutations and then converts evaluated outcomes into reusable lineage-memory rules. Using EvoAgent, we designed and fabricated a 10-mm robot with electromagnetic actuation and mechanical locomotion systems. With fewer successful simulator evaluations, the evolved actuator achieved a 2.3-fold higher thrust coefficient than the genetic-algorithm baseline. The robot reached speed of 6.5 body lengths per second and remained mobile across 30 frequency–current conditions compared to 10 for the baseline. These results demonstrate cognitive evolution as a design paradigm for high-performance untethered robots with broad operating margins.

Introduction

Microrobots can inspect confined spaces^{1,2}, monitor environments^{3,4}, and operate within engineered microsystems^{5,6}. However, achieving high-performance untethered locomotion at the millimeter scale remains a major design challenge^{7,8}. At this scale, onboard power, actuation, and mechanical transmission systems must share severely limited size, mass, and energy budgets^{9–11}. This sharing and the interdependent operating conditions strongly couple these subsystems, such that modifying one subsystem may require resource redistribution and alter the mechanical loading across the robot^{12–14}. Consequently, locally beneficial modifications can compromise geometric, manufacturing, or functional feasibility elsewhere in the integrated design^{15–17}. Moreover, the system-level physical behaviors arising from these couplings cannot always be completely captured using predefined mathematical constraints. Thus, identifying manufacturable, high-performance robot-level configurations within a strongly coupled design space becomes a major challenge when system-level feasibility is only partially captured by explicit constraints.

Evolutionary robotics provides a framework for addressing this challenge through the iterative generation, evaluation, and performance-guided updating of candidate robot bodies and behaviors^{18–21}. Within this iterative robot-design loop, candidate proposal and search-state updating can be implemented using different search strategies. Population-based methods, such as the genetic algorithm (GA), explore candidate populations without requiring gradients^{22,23}, but become costly when each candidate requires high-fidelity evaluation²⁴. Surrogate-based methods, including Bayesian optimization (BO), reduce the evaluation cost using predictive models²⁵, with constraint-aware and pretrained variants further incorporating feasibility information or prior optimization experience^{26–28}. However, the reliability of these methods can decline near constraint boundaries and at discrete structural transitions, where predictive models struggle to accurately represent the underlying physical

relationships. Large-language-model (LLM) agents provide another proposal mechanism by incorporating physical priors, evaluation feedback, and prior experience expressed in natural language into subsequent design decisions^{29–34}. Notably, the influence of conversational history on LLM proposals depends on the context length and information position; therefore, the reliable use of relevant earlier outcomes degrades as the design record grows³⁵. However, in these strategies, the retrieval of the physical diagnoses behind the evaluated outcomes and their reuse in subsequent subsystem-level decisions remain difficult as the design history grows.

In this study, we defined cognitive evolution as an evolutionary robot-design process in which the physical reasons for successful and failed outcomes guide subsequent design decisions. We implemented this process in EvoAgent, an LLM agent. Before evaluation, EvoAgent combines physical priors, active constraints, and the accumulated design history to generate a structured proposal containing parameter edits, physical rationale, expected performance trends, and risk notes. The proposal is parsed into a candidate genotype, revised when necessary, and screened by deterministic validators for geometric, manufacturing, and assembly constraints before task-specific physical evaluation is performed. After each evaluation, EvoAgent records the performance, constraint margins, failure mode, and physical diagnosis. Recurring observations from successful and failed outcomes are then consolidated into lineage-memory rules that guide subsequent proposals. In our experiments, we applied EvoAgent to design actuation and locomotion systems, which we fabricated as a 10-mm-long, 1-g untethered robot. Our observations demonstrate cognitive evolution as a design paradigm for realizing high-performance untethered robots with broad operating margins.

Results

Cognitive evolution for untethered-robot design

To achieve untethered locomotion at the millimeter scale, all subsystems must be confined to the same restricted volume. In the designed robot, the electromagnetic actuator, onboard battery, and locomotion structure collectively occupied a volume of approximately 10 mm in each dimension (Fig. 1a). The magnetic-core and winding dimensions set the available thrust while simultaneously determining the actuator mass and volume available to the remaining components. The battery placement and front- and rear-leg morphologies governed the center of mass and foot–ground contact, through which thrust is converted into forward motion (Supplementary Fig. S1). Importantly, these variables cannot be selected independently; a change that increases the actuator force can also enlarge the package, shift the center of mass, or eliminate the effective ground contact. Therefore, a feasible design must simultaneously satisfy electromagnetic, geometric, manufacturing, and locomotion requirements, and each design candidate must be subjected to high-fidelity physical evaluation.

The EvoAgent LLM agent addresses this design problem by altering both the manner in which candidate descendants are generated and the process through which the evaluated outcomes are inherited within a common physical evolution loop (Fig. 1b). Specifically, each generation begins from a design context

assembled from the current and incumbent best designs, recent lineage, retrieved physical priors, accumulated experience, and constraint and critic feedback. EvoAgent uses this context to perform a cognitive mutation, that is, the language model proposes parameter edits and supplies their physical rationale, expected performance trends, and anticipated risks. This proposal is encoded as a candidate genotype and screened against deterministic geometric, manufacturing, and assembly constraints before being submitted to an expensive evaluator. For each evaluated descendant, whether successful or failed, a performance score is returned together with a physical diagnosis of the mechanism that limited its performance. These outcomes are retained as individual design records and compressed into recurring mutation rules and higher-level design principles, which together form the lineage memory that is returned to the design context to guide later mutations. Thus, candidate generation, physical evaluation, and knowledge inheritance form a closed evolutionary loop. A representative record of this context-to-gene proposal, evaluated descendants, and the resulting lineage-memory hierarchy is provided in Supplementary Fig. S2.

We applied this framework to two coupled subsystem-design tasks for the untethered robot (Fig. 1c). The first task involved the evolution of the linear electromagnetic actuator. The magnetic-core dimensions, air-gap geometry, slot dimensions, and winding-related variables formed the genotype and developed into actuator phenotypes, which were evaluated using finite-element electromagnetic simulations. Hence, an actuator design was selected. The second task adopted this selected actuator as the starting point and involved the evolution of the locomotion system. The front- and rear-leg dimensions, foot angles, and battery position formed the genotype, and the resulting forward displacement of the robot was evaluated using a multibody-dynamics simulation. The two tasks differed in terms of their variables, constraints, and physical evaluators; however, both were implemented through the same cognitive evolution process in which the design context guided the mutation before evaluation, and the evaluated outcomes were inherited as design experience. The selected designs for the actuation and locomotion systems were then fabricated and integrated into the untethered robot shown in Fig. 1a. The genotype definitions, search ranges, deterministic constraints, and physical-evaluator settings for both subsystems are specified in Supplementary Notes 1 and 2 and Supplementary Tables S1–S6.

Lineage-guided constrained evolution

The closed evolutionary loop described previously links candidate generation to the inheritance of evaluated outcomes. To determine the effects of this linkage on constrained evolution, we compared EvoAgent with conventional baselines under matched task objectives, constraints, and physical evaluators. The baselines comprised population-based methods (GA and particle swarm optimization (PSO)), surrogate-based methods (BO, sequential model-based algorithm configuration (SMAC), and BO by density-ratio estimation (BORE)), pretrained optimizers (pretrained-transformer constrained expected improvement (PFN-CEI) and the pretrained optimization model (POM)), and an LLM-only baseline without external lineage memory. We assessed the endpoint performance and number of successful simulator evaluations required for convergence. The configuration of each method, together with the

shared evaluation-accounting and convergence rules, is reported in Supplementary Note 3 and Supplementary Table S7.

For the linear electromagnetic-actuator task, the EvoAgent path was compared with that of a Darwinian baseline, that is, a GA operating under the same encoding, constraints, and evaluators. The projected trajectories revealed that failed proposals and feasible simulations from the Darwinian baseline were dispersed across broad, weakly productive regions of the design space (Fig. 2a). In contrast, the EvoAgent lineage advanced toward lower composite scores, which indicate more favorable actuator designs, through a narrower sequence of physically evaluable descendants. Note that the actuator landscape was locally rugged because the force-related output conflicted with the mass and volume penalties. Increasing the magnetic path area, slot capacity, or winding turns would improve force generation; however, these modifications would also increase the device size or weight, with discrete winding-layer transitions introducing abrupt local changes. The lineage memory employed by EvoAgent captured this structure as reusable mutation guidance. For example, the stored rule linking the slot geometry to the winding-layer count increased the willingness of the search to accept a small mass–volume penalty when it enabled a considerably larger force-related gain. Across repeated runs, this guided route through the rugged actuator landscape was reflected in a substantially better mean actuator score at a lower convergence cost than those achieved by the Darwinian baseline (Fig. 2b).

The same pattern applied to the robot-leg morphology evolution. In the projected trajectory, the EvoAgent lineage moved toward higher robot-leg scores, whereas the representative Darwinian-baseline run contained failed descendants and evaluated descendants with lower scores (Fig. 2c). The projected landscape formed a horseshoe-like threshold structure, with high displacement concentrated in a limited region that was reached only when the actuator output and leg morphology crossed a matching threshold. Through stochastic variation, this region could be approached near sharp morphology-dependent transitions; however, this method also generated many descendants outside the effective locomotion regime. In contrast, EvoAgent used lineage-guided decisions to concentrate the evaluated descendants in the high-score region. Across repeated runs, EvoAgent reached this regime with fewer successful simulator evaluations than the baseline (Fig. 2d). The relevant numerical scores, convergence costs, and failure counts are listed in Supplementary Tables S8 and S9.

Lineage-level evolutionary accounting

Lineage-tree visualizations were prepared to distinguish the two optimization paradigms within the same constrained physical design problem (Fig. 3a). The Darwinian baseline followed a random heuristic optimization paradigm in which variations generated many branches, with selection occurring after physical evaluation. In this lineage, many branches terminated without improving the inherited design state; therefore, successful simulator evaluations often remained isolated search events. In contrast, EvoAgent implemented a sequential cognitive decision paradigm in which the proposal, evaluation, and inheritance were linked by lineage memory. The improved descendant and founder states were connected through a narrower inherited path. Therefore, this lineage-guided mutation changed both the

candidates that were evaluated and the evaluated outcomes that comprised the useful ancestry employed for subsequent mutations.

Evolutionary fate maps were also prepared, in which the same process was decomposed into generated descendants, constraint-passing candidates, successful simulator evaluations, and lineage-improving outcomes (Fig. 3b). The Darwinian baseline, an LLM-only mutation approach, and EvoAgent were considered. Without a cognitive proposal, the Darwinian baseline relied on random variation to find candidates that satisfied the coupled constraints. This approach produced many dead descendants before a useful physical evaluation was performed. In the LLM-only mutation case, this stage was adjusted through language-model reasoning without lineage memory; hence, both the fraction of descendants that passed the constraints and the yield of improved outcomes among the evaluated candidates were increased. EvoAgent also changed the update step: By storing evaluated outcomes, critic diagnoses, and compressed design rules in the lineage memory, it stabilized long-horizon decisions and increased the fraction of descendants that advanced the lineage. Note that the constraint-passing fraction of EvoAgent remained below that of the LLM-only mutation case because the directions that are easiest to validate and those that improve the objective can conflict under coupled constraints. Therefore, the fate map quantifies the manner in which evaluations were allocated to descendants with higher lineage-improvement values. The corresponding proposal-budget accounting is listed in Supplementary Table S10.

To show the performance consequences of these differences, convergence traces were plotted for the Darwinian baseline, LLM-only mutation, EvoAgent with expert feedback, and full EvoAgent (Fig. 3c). During the early search stage, methods employing language-model reasoning outperformed the Darwinian baseline, indicating that proposal reasoning improved the quality of the evaluated descendants. The distinction between the LLM-only mutation and EvoAgent emerged later. Specifically, the LLM-only trajectories became more dispersed during later evaluations, indicating less stable proposal decisions. In contrast, a broader early exploration phase was observed for EvoAgent, followed by a narrower late trajectory. This pattern matches the rule-consolidation mechanism in lineage memory: Repeated diagnoses of saturation, clearance, and winding capacity are compressed into rules that stabilize later proposals and help the lineage move beyond local improvement plateaus. The separate contributions of retrieval and lineage memory to this behavior are shown in Supplementary Fig. S3.

Lineage-memory transfer and expert steering were used to further test the inheritance mechanism. When the wire diameter or actuator topology was changed, the transferred lineage memory provided parameter priors that aided the selection of promising initial designs, along with reusable saturation, clearance, and winding rules to guide subsequent mutations (Fig. 3d and Supplementary Figs. S4 and S5). By combining memory-informed initialization with rule-guided mutation, lineage-memory transfer reduced the number of rounds required to reach the target score thresholds, with the largest acceleration approaching one order of magnitude. Expert feedback provided an external prior over the promising design settings (Fig. 3e and Supplementary Fig. S6). Compared with autonomous searching alone, the expert-assisted lineage began from a more favorable initial design state and followed a steeper

subsequent improvement trajectory, indicating faster convergence after feedback was introduced. These results show that lineage memory acts as reusable design knowledge across nearby design regimes and can be initialized externally through expert guidance.

Hardware validation of evolved untethered-robot designs

The selected actuator and locomotion-system designs were fabricated and tested to connect simulation-selected designs with robot-level measurements (Fig. 4). A component test was conducted to evaluate whether the actuator selected through cognitive evolution retained its simulated force advantage after fabrication. The actuator assembly sequence is shown in Supplementary Fig. S7. Under the same thrust–current protocol, three EvoAgent actuators achieved a mean fitted thrust coefficient of 0.663 mN mA^{-1} , compared with 0.284 mN mA^{-1} obtained across three Darwinian-baseline actuators (Fig. 4a). At the highest tested current of 50 mA, the corresponding mean static thrusts were 32.9 and 14.3 mN, respectively. Agreement was achieved between the fitted coefficients and high-current thrusts. Moreover, compared to the Darwinian baseline, the fabricated EvoAgent actuator exhibited an increased current-to-force gain over the tested range.

The actuator, leg morphology, and battery placement were then integrated into complete untethered robots (Fig. 4b). The corresponding robot assembly sequence is shown in Supplementary Fig. S8. The 10-mm-long, 1-g EvoAgent robot reached a maximum forward speed of 6.5 body lengths (BL) per second. High-speed imaging resolved the robot-leg motion and body displacement across the tested driving frequencies (Supplementary Movie 1). Thirty conditions were tested across a 6×5 frequency–total-current operating map, which spanned 5–50 Hz and 20–100 mA combined current supplied to the two actuator channels; the EvoAgent robot produced a measurable forward displacement for all 30 test conditions. In contrast, the robot derived from the Darwinian baseline moved for only 10 of the 30 conditions (Fig. 4c). Averaging the three repeats within each condition and then averaging across the 30 conditions yielded 2.76 and 0.43 BL s^{-1} for the EvoAgent robot and baseline, respectively. This difference was most evident at lower total currents: Over the 20–60 mA total-current rows of the map, the baseline produced no measurable forward speed under any tested frequency condition, whereas the EvoAgent robot remained responsive throughout. Note that the ability to locomote at lower total currents relaxes the current requirement for untethered operation.

Overall, the integrated platform demonstrated fast untethered locomotion (Fig. 4d), turning at up to 356° s^{-1} (Fig. 4e), along with transport of a payload equal to 5.2 times its own weight. The forward locomotion, turning, and payload transport of the assembled robot are recorded in Supplementary Movies 2–4. To contextualize the size-normalized locomotion performance and maneuverability of the prototype, these characteristics were compared with those of previously reported untethered, onboard-powered millimeter-scale platforms, as shown in Fig. 4f^{7,8,15,16,36–43}. Collectively, the component- and robot-level tests showed that the selected subsystem designs retained their performance after fabrication and integration.

Discussion

For untethered millimeter-scale robots, successful onboard integration requires an actuator output, mechanical transmission, leg–ground interaction, and battery placement designed to support locomotion within tight size, mass, and energy budgets. Therefore, many candidate designs fail to satisfy geometric, manufacturing, or physical constraints before they become functional robotic systems. EvoAgent addresses this robot-design difficulty by introducing physical reasoning before evaluation. Lineage diagnosis, constraint retrieval, and feasibility screening guide this LLM agent toward physically plausible actuator and leg descendants. After evaluation, successful and failed outcomes are converted into reusable lineage-memory rules. Hence, design experience is inherited as performance evidence and interpretable physical knowledge for subsequent robot-design proposals.

The EvoAgent concept is related to language-model optimizers, surrogate-based optimization, and evolutionary methods. LLM-based optimizers can use score histories or textual feedback to propose candidates^{32–34}. BO, SMAC, BORE, PFN-CEI, POM, and related methods fit or reuse search models or optimization experience from evaluated samples and previous tasks^{25,27,28,44,45}. These method families retain evaluated history in different forms: Surrogate methods fit response models, pretrained optimizers reuse learned parameters, and LLM-based optimizers typically impose conditions based on score or text histories. EvoAgent stores physically grounded constraint diagnoses and mutation rules as lineage memory. This distinction is reflected in both the actuator and leg-morphology optimizations. The actuator landscape features a rugged tradeoff among the force-related output, mass, and volume, whereas the leg-morphology landscape exhibits a threshold-like locomotion regime governed by actuator–structure matching. For methods that depend primarily on smooth response models, local irregularities and threshold transitions can impede the exploitation of global trends. This behavior parallels Baldwin-like acceleration, in which learned behavior imposes biases on evolutionary trajectories⁴⁶. In EvoAgent, within-lineage learning acts at the mutation stage before evaluation; therefore, it changes the candidates exposed to selection while maintaining continuity across successive descendants.

The experiments conducted in this study identified three capabilities of this framework. First, lineage memory renders the evolutionary process interpretable. Under strongly coupled constraints, many random descendants are removed through geometry, magnetic-circuit, or fabrication checks. The useful design experience pertains to failure modes such as saturation, insufficient clearance, or winding-capacity violations, which are not recorded by scalar scores. Second, lineage memory is transferable across neighboring design regimes that share physical constraints but differ in parameter scale, topology, or local optima, as demonstrated by the lineage-memory-assisted transfer protocol. Third, the search remains controllable because expert feedback can initialize the lineage with more favorable design settings and accelerate the subsequent convergence. Conventional population and surrogate methods remain attractive when evaluations are inexpensive, massively parallelizable, or sufficiently dense to support reliable model fitting.

The fabricated EvoAgent-derived actuator achieved a thrust coefficient that was 2.3 times that of the Darwinian baseline, and the integrated robot combining the EvoAgent-optimized actuator and leg structure produced measurable motion for all 30 operating-map conditions compared to 10 for the baseline. Collectively, these hardware measurements demonstrate that the selected subsystem designs retained their performance advantages after fabrication and system integration.

The mechanisms that render EvoAgent effective for constrained robot design also define its limitations. Proposal decisions are mediated by a probabilistic language model and constrained at the prompt and critic levels. Repeated runs continue to exhibit substantial variance and occasional divergence, even when reflection and critic review stabilize the lineage. The memory transfer behavior also depends on the match between the retrieved experience and physical couplings active in the current design regime; poorly matched records can redirect mutations toward target-irrelevant regions. In future implementations, lineage records selected on the basis of physical and textual semantic similarities should be retrieved. The hardware results reported herein validate one untethered millimeter-scale robot platform; sim-to-real calibration and application to other coupled multiphysics systems will be tested in future work. A further step is to extend the framework toward embodied evolutionary loops, in which physical experiments contribute directly to lineage updates. In its current form, EvoAgent establishes a closed simulation-based evolutionary loop, followed by the fabrication and hardware validation of the selected designs.

Methods

Evolutionary representation and evaluation accounting

In this study, “cognitive evolution” denotes an evolutionary robotics process in which language-model reasoning shapes candidate generation prior to physical evaluation. A genotype $x \in X$ encodes the actuator- or leg-design variables. A deterministic development map $\phi: X \rightarrow G$ converts x into a physical phenotype $g = \phi(x)$, and a physical evaluator $E: G \rightarrow R \cup \{fail\}$ returns either an objective value or an execution failure, denoted *fail*.

At generation t , a proposal mechanism M acts on the current lineage or search state s_t and generates the proposal set

$$P_{gen}^{(t)} = M(s_t) \subseteq X. \quad (1)$$

The generated set is then filtered through viability checks and physical evaluation:

$$P_{val}^{(t)} = \{x \in P_{gen}^{(t)} : v(x) = 1\}, \quad (2)$$

$$P_{succ}^{(t)} = \{x \in P_{val}^{(t)} : E(\phi(x)) \in R\}, \quad (3)$$

where $v(x)$ is the deterministic viability check. Successful simulator evaluations receive objective values:

$$f(x) = E(\phi(x)), x \in P_{succ}^{(t)}. \quad (4)$$

An update operator U advances the search state:

$$s_{t+1} = U(s_t, P_{gen}^{(t)}, P_{val}^{(t)}, P_{succ}^{(t)}, f). \quad (5)$$

The evaluation count is defined as the cumulative number of successful simulator evaluations, and the convergence cost is the evaluation count at which the cumulative-best trace satisfies the plateau criterion defined in the “Statistical analysis” section. In a successful simulator evaluation, the external simulator terminates normally and returns an interpretable result. Predefined physical-failure outcomes are included because each is mapped to a task-defined objective value: Actuator failures receive the penalty value, and locomotion outcomes, such as a robot-flight event in which the simulated robot loses ground contact, receive zero forward displacement. Therefore, although such outcomes never improve the cumulative-best performance, they belong to the successful-evaluation set defined above and enter the evaluation budget and convergence index. Generated proposals, viability failures, and revised proposal attempts rejected before simulation are reported separately and excluded from this budget. Simulator crashes, time-outs, non-converged runs, and uninterpretable outputs are recorded separately as execution failures.

Operators M and U define the points at which the algorithmic differences enter this common optimization loop. Conventional baselines realize them through native sampling, variation, replacement, or model-update rules. In EvoAgent, cognitive evolution occurs through these two operators. M is realized as the pre-evaluation proposal mechanism, and U is realized as the lineage memory update. The language-model reasoning acts before an expensive evaluation is performed, and the evaluated outcomes are retained as reusable design experience. In the application considered herein, all methods employed the same variable bounds, constraints, and evaluators for each subproblem. Task-specific objectives were defined for the design of actuation and locomotion systems.

In the proposed approach, over one optimization run, the generated candidates form the accounting funnel

$$P_{imp} \subseteq P_{succ} \subseteq P_{val} \subseteq P_{gen} \subseteq X, \quad (6)$$

$$X^* = opt_x \in P_{succ} f(x), \quad (7)$$

where opt denotes the task-specific optimization direction. For the actuator design problem, $f(x)$ corresponds to the reference-normalized composite actuator score, with lower values denoting a more favorable force-related output after considering power, volume, and mass. For the locomotion-system problem, $f(x)$ corresponds to the robot-leg score derived from the signed forward locomotion objective, with higher values denoting more favorable locomotion. Actuator phenotypes were evaluated by magnetostatic finite-element analysis in ANSYS Electronics Desktop 2023 R1, and locomotion phenotypes were reconstructed parametrically in SolidWorks and evaluated by multibody-dynamics

simulation in MSC Adams 2024. The full objective definitions, variable bounds, derived relations, solver settings, and constraint expressions are provided in the Supplementary Information and code.

Pre-evaluation proposal mechanism in EvoAgent

EvoAgent implements M as a structured pre-evaluation harness. Each generation is represented by the following lineage state:

$$s_t = (x_t, x_t^*, F_t^*, H_t, K_t, E_t, C_t), (8)$$

where x_t is the current design; x_t^* and F_t^* are the incumbent best designs and scores, respectively; H_t is the recent lineage history; K_t is the retrieved domain knowledge; E_t is the accumulated experience; and C_t is the constraint and critic feedback. A context builder compresses this state into the design context that is supplied to the language model:

$$z_t = \Gamma(s_t), (9)$$

where Γ denotes the context-building step. The language model acts as a conditional proposal policy within this context:

$$a_t \sim \pi_{pre}(\cdot | z_t), (10)$$

and the harness implements the action by yielding generated candidates:

$$P_{gen}^{(t)} = M_{EA}(s_t) = \psi(s_t, a_t). (11)$$

Here, ψ converts the state–action pair into structured genotype proposals and applies format constraints, revision, and pre-evaluation checks.

The workflow proceeds as follows: The current lineage state is diagnosed, relevant design knowledge is retrieved, the model is asked to output a structured descendant, and the proposal is screened prior to simulation. Deterministic viability checks remove candidates that violate geometric or manufacturability constraints, and a critic ensemble reviews the remaining proposals based on magnetic plausibility, expected performance trends, constraint boundary risks, and mutation magnitude. The per-round information flow of this decision layer and the critic ensemble applied before evaluation are described in Supplementary Note 4 and listed in Supplementary Tables S11 and S12. The lineage-memory record schema, reflection triggers, and knowledge hierarchy used for inheritance are described in Supplementary Note 5 and listed in Supplementary Tables S13–S15.

The back-end language model employed in this study was GPT-5.2, which was accessed through an OpenAI-compatible OpenRouter interface. The model identifier, decoding settings, retry rules, prompts, tool schemas, and output formats are archived with the run logs and summarized in Supplementary Note

6 and Supplementary Tables S16 and S17. The notation treats proposal generation as a conditional decision rule. The language-model parameters remain fixed, and adaptation occurs through external lineage memory, retrieved rules, and strategy records supplied in later contexts.

Lineage update for interpretable, transferable, and steerable mutation

Following screening and evaluation, EvoAgent updates the lineage through an explicit lineage-memory record. Let d_t denote the diagnostic record for generation t , which includes the proposal rationale, constraint outcome, critic comments, and reflection summary. EvoAgent implements the update operator as

$$s_{t+1} = U_{EA}(s_t, P_{gen}^{(t)}, P_{val}^{(t)}, P_{succ}^{(t)}, f, d_t). \quad (12)$$

This update changes the current and incumbent designs in accordance with the same performance accounting implemented by all methods and stores the proposal, viability outcome, evaluator result, replacement decision, and diagnosis in the lineage memory.

Repeated observations are compressed into mutation rules and broader design principles. These rules capture physical couplings, such as winding-layer transitions and saturation-related clearance changes, and are retrieved as lineage contexts in subsequent generations.

For the transfer experiments, EvoAgent retained design experience through lineage-memory records. During raw transfer, archived records were reused directly; for distilled transfer, natural-language summaries of design principles and parameter priors were used; and for hybrid transfer, both were combined. Expert feedback was incorporated once through the lineage-memory interface and before the first evaluated generation of the EvoAgent condition. The expert reviewed the initial design space, variable bounds, and relevant physical constraints and provided high-level parameter guidance. This guidance was used to establish the priors from which the initial genotype was generated.

Baselines and budgets

The baselines included GA, PSO, BO, SMAC, BORE, PFN-CEI, and POM; the GA condition is referred to as the Darwinian baseline throughout. These baselines follow standard population, surrogate, and pretrained-optimizer formulations^{22,23,25,27,28,44,45}. An LLM-only baseline that used a language-model mutation without external lineage memory^{29,31} was also employed. All methods were implemented using the same genotype ranges, phenotype-generation scripts, viability checks, physical evaluators, objectives, and evaluation budgets. The baselines were implemented with their native sampling, replacement, and model-update rules. EvoAgent used lineage memory, retrieval, and critic reviews before evaluation. The hyperparameters, random seeds, and implementation details are fully reported in Supplementary Table S7 and the archived run logs. The ablation conditions that vary retrieval and lineage

memory are defined in Supplementary Note 7 and summarized in Supplementary Table S18, and the transfer and expert-feedback protocols are described in Supplementary Note 8.

EvoAgent was first applied to the actuation system and then to the locomotion system. The actuator evolution first established a viable flux-path geometry and then refined the winding, miniaturization, and local magnetic-circuit parameters. The leg evolution then optimized the morphology and foot-angle parameters using the evolved actuator output. For each subproblem, the same variable bounds, constraints, evaluators, and evaluation budget were used by the algorithms to facilitate comparison.

Proposal revision was applied before high-fidelity evaluation using a deterministic constraint validator. Revised proposals were counted as generated-proposal attempts but did not enter the simulation-evaluation budget. The generated proposals, validated descendants, and successful simulator evaluations are reported separately, and the fully instrumented proposal-audit runs are summarized in Supplementary Table S10.

Fabrication and hardware characterization

The evolved electromagnetic actuators were fabricated using custom-machined components. The coils were machine-wound to the target turn counts. The E-core and I-plate were wire-cut from fixed-thickness silicon-steel sheets, and the permanent magnets were custom-cut from NdFeB N52 and magnetized. To form the intended magnetic circuit, the magnets and I-plate were held on the E-core by magnetic attraction during assembly. Additionally, a wound coil was inserted through the center beam of the E-core. The static thrust was measured on a platform comprising an analytical balance, three-axis positioning stage, vertical lift stage, custom fixture, and external power supply. The coil and core were aligned at the contact position and tested under direct currents ranging from 10 to 50 mA. Three independently fabricated EvoAgent actuators and three independently fabricated Darwinian-baseline actuators were used for thrust comparison. Component fabrication and assembly details are provided in Supplementary Note 9, and the static-thrust measurement setup is shown in Supplementary Fig. S9.

The untethered robot was assembled by integrating the actuator with the evolved leg structure, an onboard battery, and wireless control electronics. The front-leg module contained the front leg, actuator stator, battery, and controller, with the E-core joined to the front leg via a light interference fit. The rear-leg module contained the rear leg and mover coil, which were fixed in a reserved mounting position using an ultraviolet-curable adhesive. Locomotion was tested on a polyvinyl chloride substrate, and the wireless controller was used to set the frequency and duty cycle of the two actuator channels. During symmetric forward driving, the reported operating-map current was the combined current supplied to the two actuator channels, yielding a total of 20–100 mA (10–50 mA per actuator). For each specified frequency–total-current condition, the displacement was measured over a 1-s drive window, with three repeats per condition used for the operating-map summary. Turning and payload tests were conducted for the same 1-s window with three repeats per condition. The turning behavior was recorded from a top–down view with one actuator channel energized and the other deactivated, and the turning rate was determined from the recorded video. Payload transport was tested by adding mass to the robot body;

stable locomotion was retained for up to 5.2 times the robot weight. The locomotion measurement setup is shown in Supplementary Fig. S10, and further assembly, electronics, and measurement details, together with the fabricated robot dimensions and hardware summary statistics, are provided in Supplementary Notes 9 and 10 and Supplementary Tables S19 and S20.

Statistical analysis

The convergence costs were computed from the cumulative-best traces. The convergence was assessed using a trailing 20-evaluation window, which was applied to the successful simulator-evaluation trace of every method, starting from the first successful simulator evaluation. A run reached convergence when the relative improvement of the cumulative-best objective across that window fell to 1% or lower; thus, the agent and baseline runs were assessed using the same criterion. No statistical method was used to predetermine the sample size; the number of repeats was set based on the cost of high-fidelity electromagnetic and multibody-dynamics evaluations. For the hardware tests, the number of repeats was based on the fabrication cost and yield.

All successful simulator evaluations were included in the convergence index, including evaluations that returned predefined physical-failure outcomes, such as robot-flight events. Such failure outcomes were not used to update the incumbent best performance; therefore, the previous cumulative-best value was retained. Constraint-rejected proposals and revised proposal attempts rejected before simulation were excluded from the evaluation count and reported separately. Additionally, simulator execution failures, including crashes, time-outs, non-convergence, and uninterpretable outputs, were reported separately.

The generated-proposal statistics included both viable and failed candidates. Representative trajectories are shown in Figs. 2 and 3. Repeated-run summaries, sample sizes, mean $\pm$ 95% confidence intervals, baseline hyperparameters, random seeds, and convergence-cost distributions are reported in Supplementary Tables S7–S10 and the archived run logs. The points in Fig. 2b represent the method-level means computed from the archived repeated-run trajectories.

The final actuator performance scores were summarized at the independent-run level, with three runs per method, and their mean $\pm$ 95% confidence intervals are listed in Supplementary Table S8. The statistical unit for algorithm comparisons was an independent optimization run. The hardware thrust slopes were estimated for each fabricated actuator and summarized across the devices ($n = 3$ EvoAgent actuators and $n = 3$ Darwinian-baseline actuators). The operating-map displacement was summarized by first averaging the three repeats at each frequency–total-current setting and then averaging the 30 condition means. The reported values for the untethered platforms compared in Fig. 4f were acquired from the corresponding source papers, with the inclusion criteria and per-platform values listed in Supplementary Note 11 and Supplementary Table S21.

Declarations

Competing interests

The authors declare no competing interests.

Funding

Support for this study was provided by the National Science Fund for Distinguished Young Scholars (grant number 52225602).

Author contributions

W.Z., Z.T., and H.L. conceived and designed the study. W.Z. developed the EvoAgent framework, implemented the optimization pipeline, and performed the simulation experiments. P.L. and T.X. contributed to the electromagnetic actuator design and finite-element evaluation. X.W., Q.W., and B.W. fabricated the actuators and assembled the robots. Y.Z. and W.W. constructed the locomotion test platform and performed the hardware characterization. S.W. and S.L. contributed to the control electronics and data analysis. W.Z., Z.T., and H.L. wrote the manuscript with input from all authors. H.L. supervised the project. All authors have discussed the results and have approved the final version of the manuscript.

Data availability

Source data supporting Figs. 2–4, Supplementary Figs. S3–S6, and Supplementary Table S19 are provided with this paper. All other data supporting the findings of this study are available within the article and its Supplementary Information or from the corresponding author upon request.

Code availability

The code, setup instructions, and configuration files necessary to reproduce the results reported in this paper are available at GitHub (<https://github.com/Zhaoweizhi/Evoagent-microrobot-code.git>).

References

1. Wang M et al (2026) Magnetically actuated momentum-driven millirobots. *Nat Commun* 17:1169
2. Wu Y, Dong X, Kim J-K, Wang C, Sitti M (2022) Wireless soft millirobots for climbing three-dimensional surfaces in confined spaces. *Sci Adv* 8:eabn3431
3. Gao Q et al (2025) Soft magnetic microrobots with remote sensing and communication capabilities. *Nat Commun* 16:10489

4. Urso M, Ussia M, Novotný F, Pumera M (2022) Trapping and detecting nanoplastics by MXene-derived oxide microrobots. *Nat Commun* 13:3573
5. Gardi G, Ceron S, Wang W, Petersen K, Sitti M (2022) Microrobot collectives with reconfigurable morphologies, behaviors, and functions. *Nat Commun* 13:2239
6. McCaskill JS, Bandari VK, Schmidt S, Schmidt OG (2026) Modular microrobotics transitioning from remote to on-board electronic control. *Nat Rev Mater* 11:393–412
7. Liu Z et al (2024) A wireless controlled robotic insect with ultrafast untethered running speeds. *Nat Commun* 15:3815
8. Goldberg B et al (2018) Power and control autonomy for high-speed locomotion with an insect-scale legged robot. *IEEE Robot Autom Lett* 3:987–993
9. Lee HC et al (2024) A fabrication strategy for millimeter-scale, self-sensing soft-rigid hybrid robots. *Nat Commun* 15:8456
10. Jayaram K, Shum J, Castellanos S, Farrell Helbling E, Wood RJ (2020) Scaling down an insect-size microrobot, HAMR-VI into HAMR-Jr. In *Proc. 2020 IEEE Int. Conf. Robot. Autom.* 10305–10311
11. Tawfick S, Pikul J (2024) Materials challenges for powering miniature bioinspired robots. *MRS Bull* 49:100–106
12. Do TN, Phan H, Nguyen T-Q, Visell Y (2018) Miniature soft electromagnetic actuators for robotic applications. *Adv Funct Mater* 28:1800244
13. Mao G et al (2022) Ultrafast small-scale soft electromagnetic robots. *Nat Commun* 13:4456
14. Zhao W et al (2026) High-speed, maneuverable, and terrain-adaptive micro-robot with tree frog-inspired bionic feet. *Adv Sci* 13:e14807
15. Wu Y et al (2025) Untethered soft microrobot driven by a single actuator for agile navigations. *Nat Commun* 16:6842
16. Yu X et al (2024) Forward and backward control of an ultrafast millimeter-scale microrobot via vibration mode transition. *Sci Adv* 10:eadr1607
17. Liang J et al (2021) Electrostatic footpads enable agile insect-scale soft robots with trajectory control. *Sci Robot* 6:eabe7906
18. Nolfi S, Floreano D (2001) *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Bradford Books
19. Doncieux S, Bredeche N, Mouret J-B, Eiben AE (2015) Evolutionary robotics: what, why, and where to. *Front Robot AI* 2:4
20. Lipson H, Pollack JB (2000) Automatic design and manufacture of robotic lifeforms. *Nature* 406:974–978
21. Bongard J, Zykov V, Lipson H (2006) Resilient machines through continuous self-modeling. *Science* 314:1118–1121
22. Goldberg DE (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley

23. Kennedy J, Eberhart R (1995) Particle swarm optimization. In *Proc. ICNN'95-Int. Conf. Neural Networks* 4, 1942–1948
24. Garud SS, Karimi IA, Kraft M (2017) Design of computer experiments: A review. *Comput Chem Eng* 106:71–95
25. Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. *Adv Neural Inf Process Syst* 25:2960–2968
26. Khatamsaz D et al (2023) Bayesian optimization with active learning of design constraints using an entropy-based approach. *npj Comput Mater* 9:49
27. Li X et al (2024) Pretrained optimization model for zero-shot black box optimization. *Adv Neural Inf Process Syst* 37:14283–14324
28. Yu RT-Y, Picard C, Ahmed F (2025) Fast and accurate Bayesian optimization with pre-trained transformers for constrained engineering problems. *Struct Multidiscip Optim* 68:66
29. Yao S et al (2023) ReAct: Synergizing reasoning and acting in language models. In *Int Conf Learn Represent*
30. Shinn N, Cassano F, Gopinath A, Narasimhan K, Yao S, Reflexion (2023) Language agents with verbal reinforcement learning. *Adv Neural Inf Process Syst* 36:8634–8652
31. Zhao A et al (2024) ExpeL: LLM agents are experiential learners. In *Proc. AAAI Conf. Artif. Intell.* 38, 19632–19642
32. Yang C et al (2024) Large language models as optimizers. In *Int. Conf. Learn. Represent.* 12028–12068 (2024)
33. Romera-Paredes B et al (2024) Mathematical discoveries from program search with large language models. *Nature* 625:468–475
34. Liu F et al (2024) Evolution of heuristics: toward efficient automatic algorithm design using large language model. In *Proc. 41st Int. Conf. Mach. Learn.* 235, 32201–32223
35. Liu NF et al (2024) Lost in the middle: How language models use long contexts. *Trans Assoc Comput Linguist* 12:157–173
36. Miao Z et al (2023) Power autonomy and agility control of an untethered insect-scale soft robot. *Soft Robot* 10:749–759
37. Liu Y et al (2022) S²worm: A fast-moving untethered insect-scale robot with 2-DoF transmission mechanism. *IEEE Robot Autom Lett* 7:6758–6765
38. Liu Y et al (2023) Singularity analysis and solutions for the origami transmission mechanism of fast-moving untethered insect-scale robot. *IEEE Trans Robot* 40:777–796
39. Hoover AM, Steltz E, Fearing RS (2008) RoACH: An autonomous 2.4-g crawling hexapod robot. 2008 IEEE/RSJ Int Conf Intell Robot Syst, 26–33
40. Yang X, Chang L, Pérez-Arancibia NO (2020) An 88-milligram insect-scale autonomous crawling robot driven by a catalytic artificial muscle. *Sci Robot* 5:eaba0015

41. Zhu Y et al (2022) A 5-mm untethered crawling robot via self-excited electrostatic vibration. *IEEE Trans Robot* 38:719–730
42. Ji X et al (2019) An autonomous untethered fast soft robotic insect driven by low-voltage dielectric elastomer actuators. *Sci Robot* 4:eaaz6451
43. Kim B et al (2025) Insect-scale terrestrial robot capable of untethered environmental monitoring with multisensors. *IEEE/ASME Trans Mechatron* 30:7395–7405
44. Hutter F, Hoos HH, Leyton-Brown K (2011) Springer,. Sequential model-based optimization for general algorithm configuration. In *Int. Conf. Learn. Intell. Optim.* 507–523
45. Tiao LC et al (2021) BORE: Bayesian optimization by density-ratio estimation. In *Proc. 38th Int. Conf. Mach. Learn.*, 139, 10289–10300
46. Baldwin JM (1896) A new factor in evolution. *Am Nat* 30:441–451

Figures

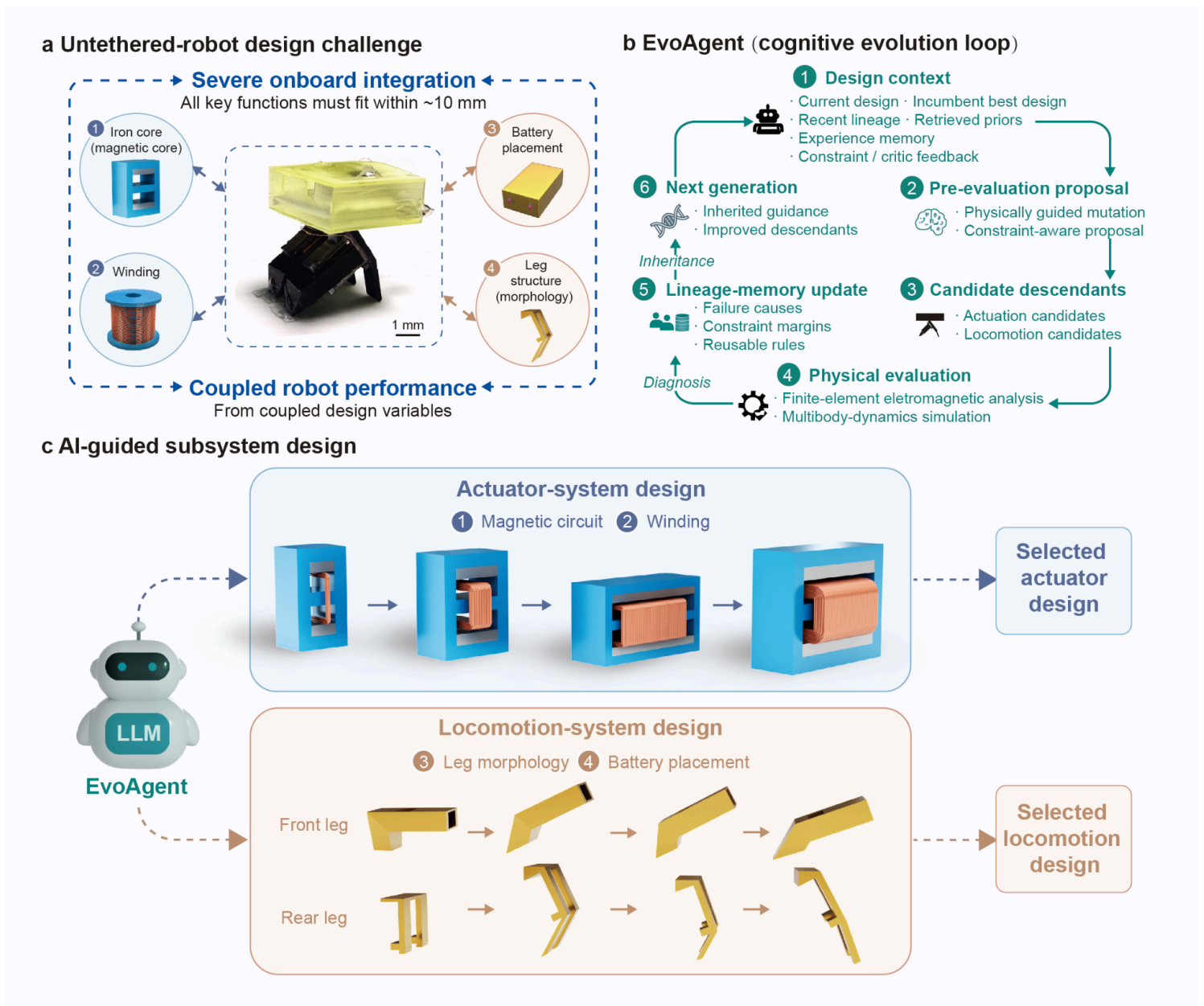


Figure 1

Untethered-robot design challenge, cognitive evolution framework, and subsystem-design application.

a Untethered millimeter-scale robot-design challenge. The fabricated 10-mm-long, 1-g platform is surrounded by representative coupled design elements, comprising the magnetic core, winding, battery placement, and leg structure, which must be reconciled within the onboard package. **b** Cognitive evolution loop implemented in EvoAgent. The design context guides cognitive mutation before evaluation; gene-encoded descendants are screened and passed to physical evaluation; and diagnosed outcomes are retained as design records and compressed into reusable mutation rules and design principles, which are returned to the design context and inherited by the next generation. **c** Application of a cognitive evolution loop to the design of actuation and locomotion systems. To select the actuator design, the magnetic-circuit and winding variables are evolved using finite-element electromagnetic

analysis. Subsequently, to select the locomotion design, the leg morphology and battery placement are evolved using multibody-dynamics analysis.

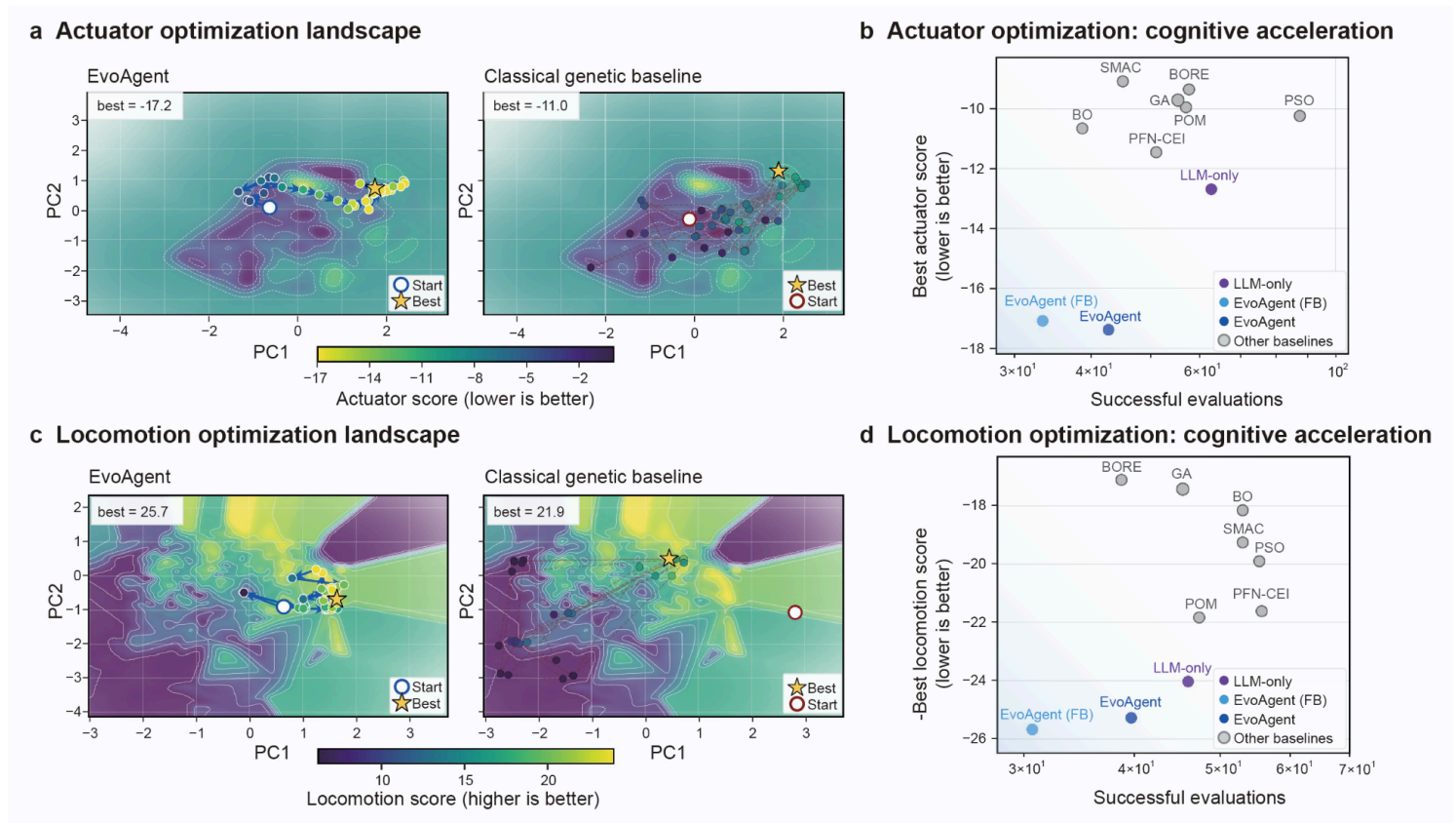


Figure 2

Optimization performance under physical evaluation. **a** Actuator optimization trajectories projected onto the two leading principal components, which captured 18.7% and 13.8% of the variance. The EvoAgent path was compared with a representative Darwinian-baseline run; failed proposals and feasible simulations from the Darwinian baseline are shown separately. The colors encode the composite score along the feasible points, with lower values denoting more favorable actuator objective values. **b** Actuator-task point plot of the mean best reference-normalized actuator score against the mean convergence cost. The points denote the mean across three independent runs per method; the numerical values are listed in Supplementary Table S8. **c** Robot-leg optimization trajectories projected onto the two leading components, which captured 28.5% and 25.5% of the variance. The EvoAgent path was compared with a representative Darwinian-baseline run; failed proposals and successful simulator evaluations from the Darwinian baseline are shown separately. The colors encode the robot-leg score, with higher values denoting greater simulated forward locomotion. **d** Robot-leg point plot of the mean final best robot-leg score against the mean convergence cost. The points denote the mean across three independent runs per method; the numerical values are listed in Supplementary Table S9.

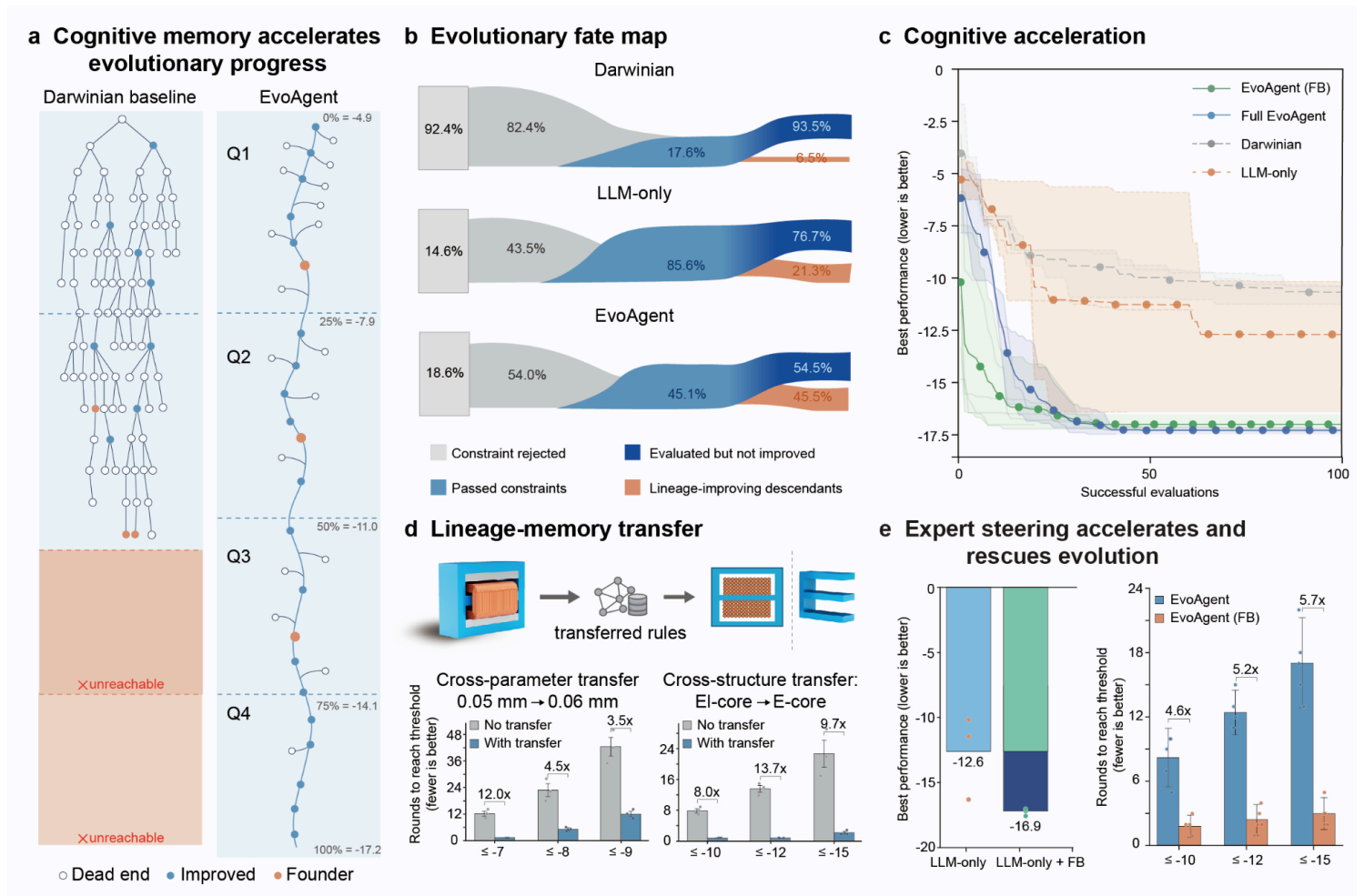


Figure 3

Lineage accounting, transfer, and steering. **a** Lineage-tree visualizations for the Darwinian baseline, LLM-only mutation, and EvoAgent, with node colors indicating dead-end, improved, and founder states, and background shading indicating performance thresholds. **b** Evolutionary fate map comparing the Darwinian baseline, LLM-only mutation, and EvoAgent across generated candidates, constraint filtering, simulated outcomes, and improved descendants. **c** Best reference-normalized actuator score plotted against successful simulator evaluations for EvoAgent with expert feedback, full EvoAgent, Darwinian baseline, and LLM-only mutation. **d** Lineage-memory-assisted transfer experiments across winding-wire diameter and actuator topology. Under the transfer condition, the source-task memory provides parameter priors for initialization and reusable physical rules for subsequent mutation. In contrast, the non-transfer condition uses default target-task initialization without source-task memory. The bars show the rounds required to reach each target threshold and quantify the end-to-end benefit of the complete transfer protocol. **e** Expert-feedback experiment comparing autonomous and expert-assisted EvoAgent searches, highlighting improved initial design quality and faster convergence.

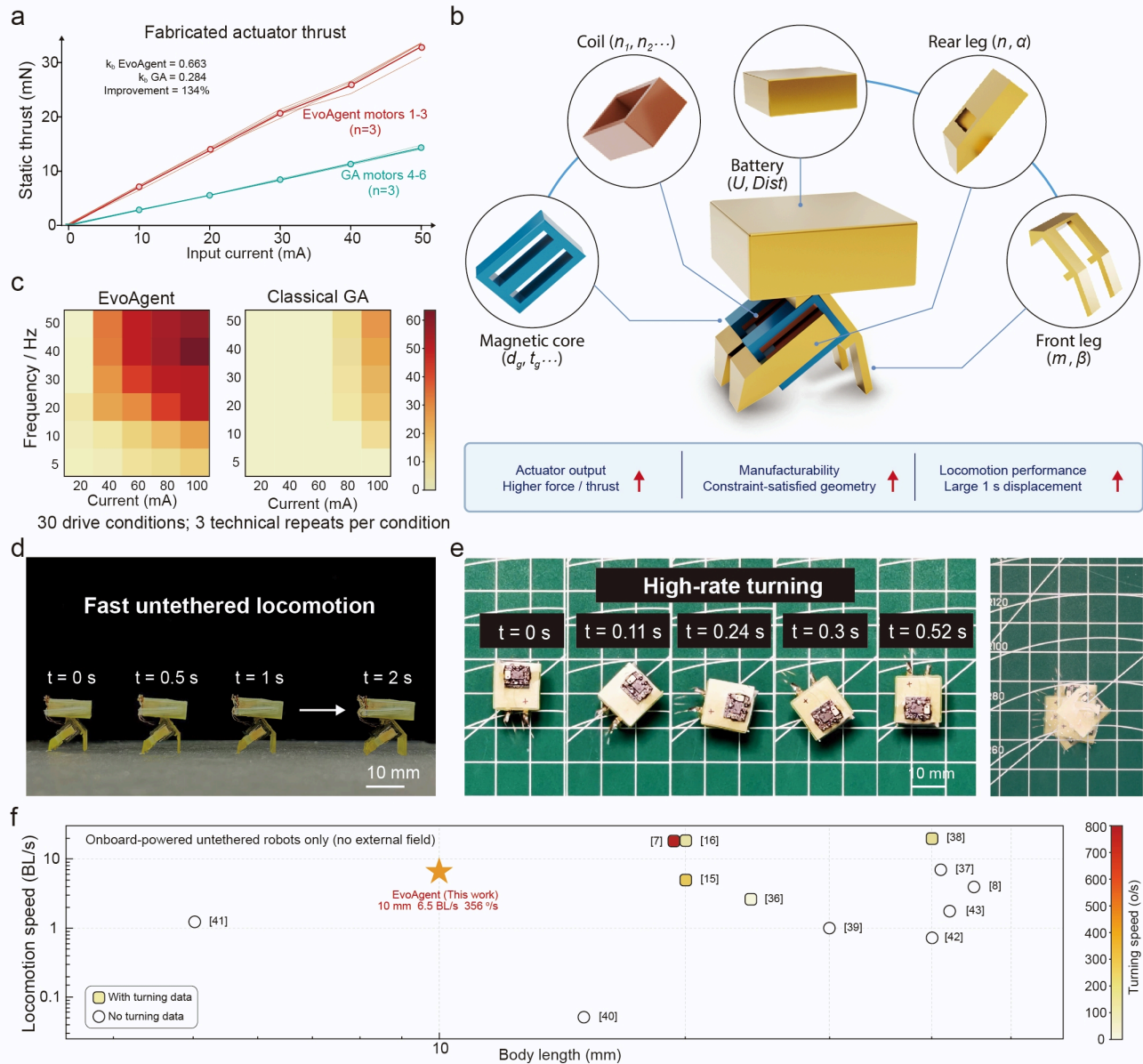


Figure 4

Performance of the fabricated untethered robot. **a** Static thrust as a function of input current for EvoAgent-derived actuators and actuators from the Darwinian baseline. The fitted thrust coefficients were $k_b = 0.663$ and 0.284 mN mA^{-1} , respectively. **b** Integrated robot and variables considered for actuation- and locomotion-system tasks. The actuation variables include the magnetic-core and winding parameters, and the locomotion variables include the battery placement and front- and rear-leg morphologies. The lower boxes summarize the actuator output, manufacturability, and locomotion performance. **c** Frequency–total-current operating maps for the EvoAgent and Darwinian-baseline robots. The current axis represents the combined current supplied to the two actuator channels during symmetric forward driving. The color indicates the mean forward speed over three technical repeats for each of the 30 drive conditions. **d** Time-lapse photographs of fast untethered locomotion under 40 Hz

and 80 mA drive (scale bar: 10 mm). **e** Time-lapse photographs of high-rate turning with the left and right legs driven at 40 and 0 Hz, respectively, under an 80 mA drive, yielding a $356^{\circ}\text{-s}^{-1}$ turning rate (scale bar: 10 mm). **f** Comparison with previously reported onboard-powered untethered robots. The body length and locomotion speed are plotted on logarithmic axes. The filled square markers denote platforms with reported turning data and are colored by turning speed; the open circles denote platforms without reported turning-speed data. The star denotes this work.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Sl.pdf](#)
- [SupplementaryMovie.zip](#)