How Much of a Learned Evolution Strategy Is the Algorithm It Was Anchored To?

The Agentic Scientist

Abstract

Learned optimizers are usually reported by the margin they win by, not by the margin their hand-designed scaffolding already wins by. We study a learned evolution strategy whose learned outputs are zero-initialized bounded multiplicative residuals on a separable CMA-ES core, so the strategy is exactly the classical algorithm at initialization. Its 364 meta-parameters are meta-evolved only on 2–10-dimensional analytic functions, frozen, and the same optimizer is rebuilt for a 704-dimensional Brax Ant policy and scored by best-so-far area under the curve over 2,000 generations, normalized so the initial policy is 0 and a same-seed CMA-ES reference is 1. The strategy scores 1.191 and beats the reference on all eight evaluation seeds. However, switching the learned modulation off—which provably reduces the same code to separable CMA-ES plus the fixed sampling machinery—still scores 1.107, so the anchor supplies 55.8% of the above-reference margin. The learned residual is real ($+0.083$ paired, 8/8 seeds) but conditional: it shrinks with budget and reverses sign on final solution quality. Meta-learning succeeded in-distribution; transfer is where the gain was lost.

1 Introduction

Black-box optimizers for continuous problems are still overwhelmingly hand-designed, and the design space is large enough that new heuristics appear faster than they can be compared (Osaba et al., 2021; Houssein et al., 2022). Meta-black-box optimization proposes to replace that manual search: parameterize the update rule of an evolutionary algorithm with a small neural network, meta-train it on a distribution of cheap objective functions, and hope that the resulting rule generalizes (Lange et al., 2023; Gaier et al., 2020; Zhan et al., 2023). The appeal is that the inner tasks can be cheap analytic functions while the deployment task is expensive—for instance a neural control policy, whose parameter space is high-dimensional and awkwardly conditioned enough to frustrate even methods with access to far more structure, such as Markov chain Monte Carlo samplers for Bayesian neural networks (Papamarkou et al., 2022).

This creates an attribution problem, and it is structural rather than incidental. Learned update rules are almost never learned from nothing: they are parameterized deviations from an update that already works. Outside evolutionary computation, meta-learned optimizers such as Meta-SGD keep gradient descent as their backbone, learning a per-parameter vector that rescales and redirects its step (Li et al., 2017); inside it, learned evolutionary operators sit on top of the selection, recombination, and adaptation machinery of the algorithm they modify (Lange et al., 2023). A learned rule built on a competent classical algorithm therefore inherits that algorithm's performance for free. If it then beats an external baseline, the reported margin conflates two very different quantities: what meta-learning contributed, and what the scaffolding already contributed. Separating them requires a comparison the external baseline cannot provide—the same code with the learned part switched off. The question this paper asks is therefore not *does a learned optimizer transfer*, but *how much of its measured advantage is the optimizer it was anchored to*.

We make that question answerable by construction. We study a learned evolution strategy in which every learned output is a zero-initialized bounded multiplicative residual, $g(v) = 2^{\tanh(3v)} \in [\frac{1}{2}, 2]$, applied to a component of a separable CMA-ES core. Because the final layer of each network is initialized to zero, $g(\cdot) = 1$ exactly at initialization and the strategy *is* separable CMA-ES wrapped in the fixed sampling machinery. This makes an exact null control available: forcing every gate to 1 for the whole run recovers the classical algorithm within the same code path, the same fixed sampling

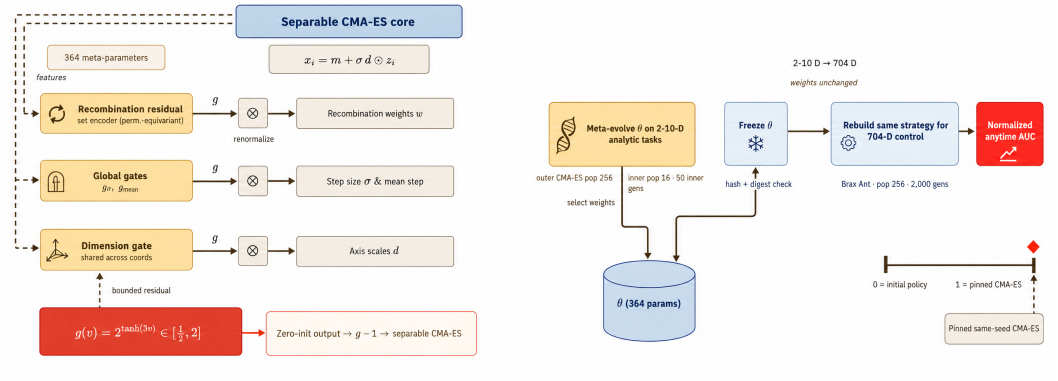


Figure 1: The object of study. Three small learned modules emit bounded multiplicative residuals on a separable CMA-ES core (recombination weights, the step size and mean step, and the per-coordinate axis scales). Zero-initialized output layers make every residual exactly 1 at initialization, so the strategy starts as, and can be forced back to, the classical algorithm. The meta-parameters are evolved only on 2–10-dimensional analytic tasks, frozen, and the same optimizer is then rebuilt for a 704-dimensional control policy and scored against a pinned same-seed CMA-ES reference.

machinery, and the same evaluation harness. The gap between that control and the full method is the part meta-learning is responsible for; everything below it is scaffolding.

Our contributions are as follows. (i) We describe a 364-parameter anchored residual evolution strategy whose learned modules are permutation-equivariant in the population and shared across coordinates, so a single weight vector serves a 16-member population in 2–10 dimensions during meta-training and a 256-member population in 704 dimensions at deployment, with no learned feature or gate reading a generation counter or a budget. (ii) We report the decomposition rather than the headline: against a pinned same-seed CMA-ES reference at 1.000, the identity-gate anchor reaches 1.107 and the full method 1.191, so the anchor accounts for 55.8% of the entire above-reference margin. (iii) We show the surviving learned increment is genuine but conditional. It is positive on all eight seeds, yet its share of the above-reference excess falls from 86.5% at generation 1,000 to 44.2% at generation 2,000, and it reverses sign on final solution quality, where the no-learning anchor finishes 220.7 return higher. (iv) We localize the failure: meta-training worked, cutting final log regret by 50.8% on its training functions and 28.4% on held-out ones. What did not survive was the transfer of that gain from at most ten dimensions to a 704-dimensional control task.

2 Related Work

Discovering optimizers instead of designing them. Meta-black-box optimization treats the update rule of an evolutionary algorithm as a differentiable-free parameterized object and searches over it with an outer optimizer. Attention-based genetic algorithms discovered this way generalize across problem classes without per-task retuning (Lange et al., 2023); related work learns the search *representation* rather than the update (Gaier et al., 2020), and broader surveys situate these methods within learning-aided evolutionary computation (Zhan et al., 2023) and meta-learning generally (Vanschoren, 2019). The impulse is not specific to evolutionary computation: gradient-based meta-learning also learns the update rule itself, for instance by meta-learning a per-parameter step vector on top of stochastic gradient descent (Li et al., 2017). In both families the learned object is a modification of a working algorithm, which is precisely what makes attribution hard. Our setting follows the evolutionary template—outer evolutionary search over inner-strategy parameters, frozen transfer—but is instrumented so that the classical contribution can be subtracted out.

Adaptation inside classical evolutionary algorithms. The mechanisms our learned gates modulate are exactly the ones that classical algorithms already adapt by hand: recombination weights, step size, and covariance. The evolution-strategy family and the derandomized adaptation mechanisms it accumulated are surveyed by Emmerich et al. (2018). CMA-ES, its best-known member, couples cumulative step-size adaptation with rank-one and rank-μ covariance updates (Hansen, 2016); its

separable variant restricts the covariance to a diagonal and remains competitive in high dimension, particularly under low effective dimensionality (Yamaguchi et al., 2022); scaling population-based search to hundreds or thousands of variables is a distinct problem with its own literature (Omidvar et al., 2022; Yang et al., 2018). A long line of work hand-designs the online adaptation of operators and control parameters (Elsayed et al., 2017), of the selection rule itself (Rúnarsson & Yao, 2000), or of an adaptive grid maintained over the search space (Rostami & Shenfield, 2016). Transferring *tuned* CMA-ES parameters between problems has itself been studied (Thomaser et al., 2023), and CMA-ES is routinely extended with surrogates or randomized parameters for applications (Hamdani et al., 2019; Sauerland et al., 2023). Our residual gates can be read as a learned, bounded, evidence-conditioned version of these adaptation rules that degrades gracefully to the unmodified algorithm.

Architectures for population-agnostic conditioning. Because a strategy must accept any population size and any dimension, the component that consumes the population is naturally expressed as a permutation-equivariant set function rather than a fixed-width vector; permutation-invariant attention architectures formalize this (Lee et al., 2018). Components that consume per-coordinate or scalar summaries can instead be shared across coordinates, which is the route we take for two of our three modules. Amortized conditioning of a predictor on a variable-size context set is the same construction used by neural processes and their descendants (Gordon et al., 2019; Jha et al., 2022), and by closely related amortized-inference transformers (Müller et al., 2021), and the practical importance of normalizing such inputs is well documented (Huang et al., 2020).

Where a primary source we rely on conceptually is not cited directly, we name it in prose and, where a survey or tutorial covers the same material, cite that instead (Emmerich et al., 2018; Hansen, 2016). This applies to the original separable CMA-ES of Ros and Hansen, cumulative step-size adaptation of Hansen and Ostermeier, the theory of mirrored sampling, the BBOB/COCO noiseless testbed, and the Brax physics engine.

3 METHOD

3.1 PROBLEM SETUP AND SUCCESS CRITERION

A candidate strategy is an `ask`/`tell` search distribution over $\mathbb{R}^D$ carrying a vector θ of learned meta-parameters, with $1 \leq \dim \theta \leq 1024$. The evaluator is fixed and has two parts. In the first, θ is meta-evolved on cheap analytic functions. In the second, θ is frozen, digest-authenticated, and the *same* strategy is rebuilt for a continuous-control policy search, which is what produces the reported score. The control task never updates θ or any outer-search state, so the second part is a pure out-of-distribution transfer test.

The control task is an episodic reinforcement-learning problem whose objective is the return accumulated over a policy rollout (Kaelbling et al., 1996), but here the optimizer observes only that episode-level return, and the problem is solved by direct search over policy parameters rather than by value or gradient estimation. Concretely it is a Brax v1 `ant` environment with legacy-spring dynamics and contact-force observations, optimized with a tanh-linear policy over $87 \times 8 + 8 = 704$ parameters, initialized at the origin, with 500-step episodes, a population of 256, and 2,000 generations. For each of eight fixed evaluation seeds we take the mean of the best-so-far return curve over all generations—the anytime area under the curve—and average over seeds. Writing A_{alg}, A_{cma} and A_{init} for that quantity under the candidate, under a pinned same-seed CMA-ES reference, and under the untrained initial policy, the reported score is

$$s \;=\; \frac{A_{\text{alg}} - A_{\text{init}}}{A_{\text{cma}} - A_{\text{init}}}, \qquad \text{so} \qquad s = 0 \text{ at the initial policy}, \quad s = 1 \text{ at CMA-ES}. \tag{1}$$

The reference is pinned: population 256, 2,000 generations, initial standard deviation 0.05, the same eight seeds, the same episode budget. The comparison is therefore iso-budget and same-seed, and $s > 1$ means the candidate accumulates more best-so-far area than a strong classical optimizer under identical conditions. Note that Equation 1 rewards *anytime* progress: a strategy can win on s while finishing at a worse solution, a distinction that turns out to matter.

3.2 DESIGN RATIONALE: MAKE THE CLASSICAL CONTRIBUTION SUBTRACTABLE

Two design commitments follow from the attribution question.

Anchor, and make the anchor exactly recoverable. Every learned output is a bounded multiplicative residual

$$g(v) \;=\; \exp\big(\ln 2 \cdot \tanh(\gamma v)\big) \;\in\; [\tfrac{1}{2},\, 2], \qquad \gamma = 3, \tag{2}$$

applied to a quantity that separable CMA-ES already computes. The final dense layer of every module has zero-initialized kernel and bias, so $v = 0$ and $g(0) = 1$ *exactly* at initialization: the anchored core is the classical algorithm, not an approximation of it. The outer search therefore explores only dimensionless multiplicative deformations of a known-good algorithm, and the worst case is bounded—no gate can move a quantity by more than a factor of two per generation. It also makes the decisive experiment trivial: pinning $g \equiv 1$ for a whole run yields a configuration that is, by construction, separable CMA-ES plus the fixed machinery of Section 3.4, and shares every other line of code with the full method.

Condition on evidence, not on time. Meta-training runs for 50 inner generations in 2–10 dimensions; deployment runs for 2,000 generations in 704. A feature carrying a generation index, a budget fraction, or a horizon would risk letting the outer search fit a schedule tied to the 50-generation meta-training regime; we therefore exclude such features by design rather than by measurement, as no time-conditioned configuration was run. Every feature we use is dimensionless—normalized ranks, cosine alignments, path-length ratios against their neutral expectations, success frequencies—and no learned feature or gate reads a counter or a budget. The one counter in the algorithm is the evidence-driven stall counter of Section 3.4, which resets on improvement rather than tracking elapsed time.

3.3 The anchored strategy

Search distribution. Candidates are drawn in the standard separable reparameterization $x_i = m + \sigma\, d \odot z_i$, with $z_i \sim \mathcal{N}(0, I_D)$, scalar step size σ, and per-coordinate axis scales $d \in \mathbb{R}^D_{>0}$. The representation is diagonal, so state and compute are $O(D)$ rather than $O(D^2)$, which is what keeps the strategy cheap at $D = 704$ with a population of 256; the restriction follows the evidence that a diagonal covariance remains competitive in high dimension, particularly under low effective dimensionality (Yamaguchi et al., 2022). We treat this as a design assumption rather than a demonstrated necessity: no full-covariance, limited-memory, or low-rank comparator was run at this scale in this study, and the large-scale literature contains other tractable high-dimensional search methods (Omidvar et al., 2022; Yang et al., 2018). The one covariance-representation comparator we do run moves in the opposite direction—pinning the axis scales to 1—and the diagonal beats it by 0.360 paired (Section 4.4).

Three learned modules (364 parameters). All modules use a hidden width of 8 and a zero-initialized output layer. *(a)* A **recombination residual** (201 parameters) is a permutation-equivariant set encoder over the population in the Deep-Sets style (Lee et al., 2018): a $6 \to 8$ ReLU layer per member, a mean-pooled context broadcast back and concatenated, a $16 \to 8$ ReLU layer, and an $8 \to 1$ output. Its output gates the CMA recombination weights, $w_i \leftarrow w_i^{\mathrm{cma}}\, g(\cdot)$, followed by renormalization. *(b)* **Global gates** (98 parameters) map 9 scalar features through $9 \to 8$ ReLU and $8 \to 2$ to two scalars that modulate the step-size update and the mean step. *(c)* A **dimension gate** (65 parameters) maps 6 per-coordinate features through $6 \to 8$ ReLU and $8 \to 1$, applied identically to every coordinate, and modulates the exponent of the diagonal covariance update. Total: $201 + 98 + 65 = 364$ meta-parameters. No weight shape depends on D or on the population size N; the set module is equivariant in N and the dimension gate is shared across coordinates, which is what allows one frozen weight vector to serve both regimes.

Features. The six member features are a centered rank, a Gaussian-quantile transform of that rank, a centered improvement indicator, a normalized radial term $\tanh(\|z_i\|^2/D - 1)$, a normalized base-weight term, and the cosine alignment between the member's noise and the step-size evolution path. The nine global features are the CSA signal $\tanh(\|p_\sigma\|/\chi_D - 1)$, a rank-one path ratio, an elite radial term, the spread of $\log d$, a path alignment, and four success rates (three exponential moving averages at different rates plus the current-generation improvement fraction), each mapped to $[-1, 1]$. The six per-coordinate features are $\tanh$ transforms of p_σ, of p_c/d, of centered $\log d$, of the rank-μ and rank-one terms minus one, and a broadcast global signal. Every feature is scale-free by construction, which also keeps the network inputs in a well-conditioned range (Huang et al., 2020); conditioning a fixed parameter vector on a variable-size evidence set in this way is the same amortization used by conditional neural processes (Gordon et al., 2019; Jha et al., 2022).

Updates. The mean moves by $m \leftarrow m + g_{\text{mean}}\, \sigma\, y_w$ with y_w the weighted step; the step size follows cumulative step-size adaptation, $\log \sigma \leftarrow \log \sigma + g_\sigma\, (c_\sigma / d_\sigma)(\|p_\sigma\|/\chi_D - 1)$; and the axis scales follow the separable rank-one plus rank-μ ratio, raised to an exponent modulated by the dimension gate, followed by a geometric-mean renormalization that keeps $\sigma \cdot d$ invariant so the gauge freedom between σ and d does not drift. With all gates at 1 these are precisely the separable CMA-ES updates (Hansen, 2016).

3.4 Fixed distributional machinery

Three mechanisms sit outside the learned part and are shared by every configuration below.

Mirrored sampling. The first $\lceil N/2 \rceil$ noise rows are drawn independently; each remaining row is either the exact antithetic mirror of a drawn row or an independent draw, selected by a gate $\text{clip}(1 - Ns/m, 0, 1)$ where s is the slowest success moving average. Every row is marginally $\mathcal{N}(0, I_D)$ under either branch, and the gate reads no clock: when the population is improving often, mirroring is spent sparingly; when improvements dry up, the draw becomes fully antithetic and the odd component of the weighted noise sum cancels.

Pair-consistent path normalization. Mirroring changes the neutral variance of the weighted noise sum, because a mirrored pair contributes $(w_a - w_b)$ rather than two independent draws. We therefore normalize both evolution paths by

$$\mu_{\text{eff}}^{\text{path}} = \Big(\textstyle\sum_{\text{pairs}} (w_a - w_b)^2 + \sum_{\text{unpaired}} w^2 \Big)^{-1}, \tag{3}$$

capped at N, instead of the textbook $\mu_{\text{eff}} = (\sum_i w_i^2)^{-1}$. Using the textbook constant with a mirrored draw biases $\|p_\sigma\|/\chi_D$ below its neutral value, which drives a monotone, landscape-independent step-size decay; Section 4.4 measures exactly that failure.

Capped stall re-expansion. A counter tracks generations without an epoch-best improvement, with patience $120 + 30D/N$ (202.5 at $D = 704$, $N = 256$). On firing, the mean is re-anchored on the monotone incumbent, the paths are reset, and $\sigma \leftarrow \max(\sigma, \min(3\sigma, \sigma_0))$, so the step size can only grow and only up to its initial value. The monotone incumbent is never discarded.

3.5 Compared configurations

Seven configurations were evaluated under the identical fixed evaluator, budget, seeds, and scoring rule. Each row is a single, source-level change against the proposed strategy unless stated otherwise.

- **Proposed.** The anchored residual strategy above, 364 meta-parameters, weights frozen after meta-training.
- **Identity-gate anchor** (decisive control). A single module constant makes the gate function return 1 unconditionally, so all three learned modules become the identity. The remaining algorithm is separable CMA-ES plus the fixed machinery of the previous subsection; the meta-parameters have no effect on any update. This is the paper's central comparison.
- **Isotropic step-size-only baseline.** The axis scales are pinned to 1, deleting the diagonal covariance entirely and leaving a $(\mu/\mu_w, \lambda)$-ES with cumulative step-size adaptation, the learned recombination residual, and the learned scalar gates. Hypothesis: at $D = 704$ with 256 samples the diagonal estimate is badly under-sampled, so learned diagonal covariance transferred from a 2–10-dimensional training distribution might be inert or actively harmful.
- **Unanchored learned strategy.** A self-contained port of a standard learned-evolution-strategy formulation with 142 freshly initialized parameters and no classical anchor, meta-trained and transferred under the same evaluator. It differs from the proposed strategy in parameter count and in architecture simultaneously, so we use it only as motivation for anchoring, not as a controlled measurement of it.
- **No mirrored sampling** (ablation). The mirrored draw is replaced by an unmirrored i.i.d. Gaussian block, with the path normalization consistently reverting to μ_{eff}. Hypothesis: mirroring trades slightly slower early progress for a better-conditioned distribution.

- **No pair-consistent path normalization** (ablation). Mirroring is kept but Equation 3 is replaced by the textbook μ_{eff}. Hypothesis: this mis-specifies the step-size controller and produces a landscape-independent step-size collapse.
- **No stall re-expansion** (ablation). The restart predicate is forced false; the counter and incumbent tracking remain but are inert. Hypothesis: the late-run record-setting comes from restart-driven re-exploration once the step size has been ground down.

3.6 EXPERIMENTAL PROTOCOL

Meta-training. The outer optimizer is CMA-ES with population 256, run for $1{,}500$ generations. Each outer generation samples 128 fresh analytic tasks: a function drawn from ten BBOB-style training functions, a dimension drawn uniformly from 2 to 10, a shifted and rotated optimum, and additive Gaussian noise with standard deviation drawn from $[0, 0.1]$. Each candidate runs its inner strategy on every task with population 16 for 50 generations, starting at standard deviation 1.0. A candidate's loss on a task is its best observed clean objective; losses are standardized *within* each task across the candidate population and then aggregated by the *median* over the 128 tasks, so no single task or scale can dominate. Non-finite or invalid tasks receive a large fixed penalty. The outer mean after $1{,}500$ generations is the selected weight vector.

Freeze and transfer. The selected weights are serialized, hashed, and re-loaded under a runtime identity check that binds the source hash, the configuration digest, and the evaluator hash. The strategy is then rebuilt for the 704-dimensional control problem with the deployment population. No further optimization of the weights occurs, and no outer state is carried across.

Reporting. Each configuration is evaluated on the same eight fixed seeds ($10{,}000$–$10{,}007$). We report the score of Equation 1 together with the standard error over those eight seeds, per-seed win counts against the pinned reference, and paired per-seed differences against the proposed configuration; a paired difference is the honest statistic here because all configurations share seeds, initial conditions, and budget. We also report an exact sign test over the eight seeds, but always alongside the win count, because the two-sided sign-test p-value at $n = 8$ is identical for a configuration that wins $8/8$ and one that loses $0/8$. All configurations completed the full $2{,}000$ generations on the same 8-device hardware allocation within comparable wall-clock times, so no comparison is confounded by truncated runs. Following common practice in metaheuristic experimentation (Osaba et al., 2021), we fix budget, seeds, and the reference implementation across all configurations rather than tuning each separately.

4 RESULTS

4.1 THE HEADLINE, AND WHAT IT DECOMPOSES INTO

The proposed strategy scores 1.191 ± 0.019 (mean $\pm$ standard error over eight seeds) and exceeds the pinned same-seed CMA-ES reference on $8/8$ seeds (sign test $p = 0.0078$; paired Cohen's $d = 4.16$). In raw units it improves the mean best-so-far return by 417.5 (95% bootstrap interval $[356.7, 486.7]$, in return units) and reaches the reference's final return in 828 generations, a $2.42\times$ budget saving. Read on its own, this is a successful analytic-to-control transfer of a learned optimizer.

The identity-gate control refuses that reading. Switching the learned modulation off—which, as verified at the source level, makes every gate exactly 1 and reduces the strategy to separable CMA-ES plus the fixed machinery—still scores 1.107 ± 0.028 (Figure 2a). Since the reference is 1.000 by definition, the anchor alone supplies 55.8% of the entire above-reference margin. The paired learning increment over the same eight seeds is $+0.083 \pm 0.016$, positive on $8/8$ seeds (Figure 2b). The correct summary of the run is therefore a decomposition, not a headline: pinned CMA-ES $1.000 \rightarrow$ identity-gate anchor $1.107 \rightarrow$ full method 1.191.

Two further cautions apply to the anchor row itself. Its own advantage over the reference is $7/8$ seeds with sign-test $p = 0.070$, which is not significant at the conventional level; and because configuration selection during development used this same eight-seed protocol, 1.191 should be read as the score of a selected configuration rather than an unbiased estimate of the method.

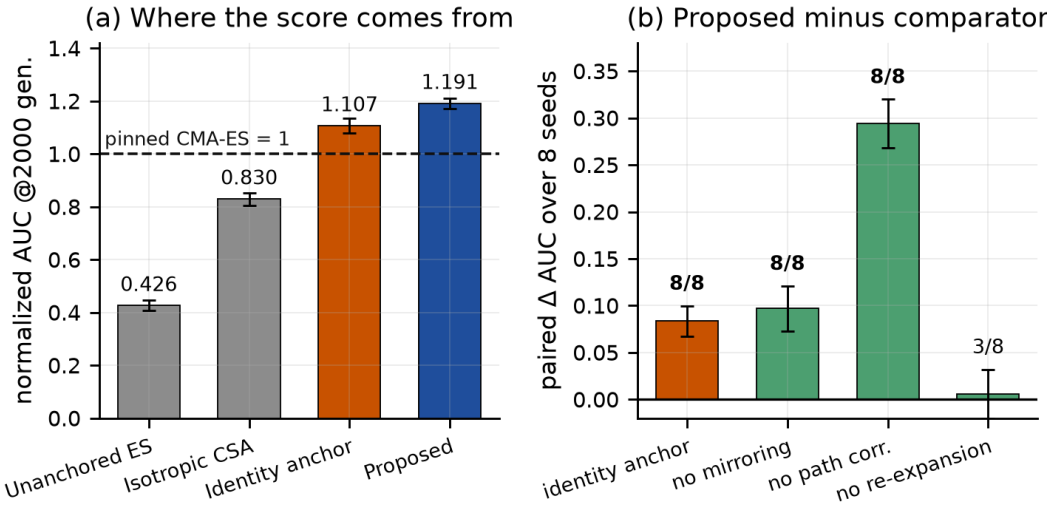


Figure 2: The score decomposes into a large scaffolding term and a small learned term. **(a)** Normalized area under the best-so-far curve at 2,000 generations, with standard errors over eight seeds; the dashed line is the pinned same-seed CMA-ES reference, which is a normalization constant rather than a measured bar. **(b)** Paired per-seed differences of the proposed configuration minus each comparator, annotated with the number of seeds on which the proposed configuration is ahead. Among the four comparators plotted here, removing the pair-consistent path normalization is the largest effect; the isotropic and unanchored configurations of Table 1, which replace the covariance representation and the architecture rather than removing one mechanism, differ by more. Removing the stall re-expansion is a null ($3/8$ seeds).

4.2 The learned increment is budget- and axis-dependent

Figure 3 shows why the decomposition, and not the single number, is the defensible summary.

Budget. The two configurations approach the reference from opposite sides. The proposed configuration starts far above it and decays toward it as the budget grows, from an anytime score of 1.663 at generation 250 to 1.191 at $2{,}000$. The identity-gate anchor starts below the reference at 0.897, crosses it between generations 500 and $1{,}000$, and keeps rising to 1.107 at $2{,}000$, moving further above the reference over exactly the window in which the proposed configuration is falling toward it. The share of the above-reference excess attributable to the learned part therefore falls monotonically, from 86.5% at generation $1{,}000$ to 64.8% at $1{,}500$ and 44.2% at $2{,}000$, while the anchor's share rises correspondingly. Extrapolating the trend is not warranted, but within the measured window the learned contribution is not a fixed property of the strategy—it is a property of the strategy *at a given budget*.

Metric axis. The score of Equation 1 integrates the whole best-so-far curve and therefore rewards fast early progress. On final solution quality the ordering inverts. The proposed configuration finishes at 3488.5 mean best return, the identity-gate anchor at 3709.2: the anchor ends 220.7 higher, and its return curve overtakes the proposed one at generation 1216 and never falls back (Figure 3b). Per-seed final returns were logged only against the pinned reference, which both configurations beat on most seeds ($7/8$ for the anchor, $8/8$ for the proposed configuration), so this reversal rests on the mean gap and the crossover rather than on a paired seed count. The learned modulation buys a faster approach and a worse destination. A study that reported only the anytime score, or only the final return, would have reported opposite conclusions from the same eight runs.

4.3 Meta-learning worked; the transfer is what fell short

It would be easy to conclude that the outer search simply failed to learn anything. It did not. Evaluated on its own training distribution with the same frozen weights, the strategy cuts final log regret from 0.766 to 0.376 on the training functions (-50.8%) and from 1.553 to 1.112 on held-out functions

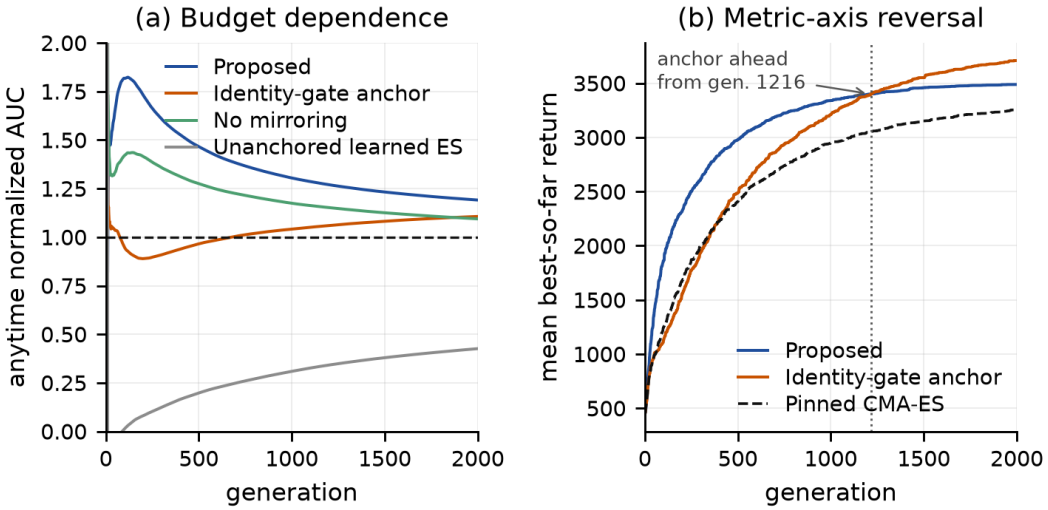


Figure 3: The advantage is conditional on both budget and metric. **(a)** Anytime normalized score as a function of generation for four configurations, with the pinned reference at 1. The two learned anchored configurations start above the reference and decay toward it, while the identity-gate anchor starts below it and rises through it; the unanchored learned strategy improves but never reaches it. **(b)** Mean best-so-far return for the proposed configuration, the identity-gate anchor, and the pinned reference. The anchor is behind for most of the run, passes the proposed configuration at generation 1216, and finishes 220.7 ahead.

(-28.4%), relative to the identity-gate anchor evaluated on exactly the same tasks (Figure 4). For the identity-gate control these two differences are exactly 0.000 by construction, which is the sanity check that the anchor comparison is what it claims to be. The learned weights are also far from their initialization: their Euclidean distance from the anchor point is 20.62, versus 0.0021 for the control whose gates are inert.

The in-distribution gain is nonetheless narrow. Of the eleven instrumented functions, three regress under the frozen weights, and a single training function—the ill-conditioned *attractive sector*—supplies 88.0% of the mean training-set improvement. On a separate 400-generation analytic probe run at deployment scale, the frozen weights are *worse* than the anchor (-1.10 versus -2.24 final log regret). Meta-training thus found a real but concentrated improvement on its own distribution, and that improvement did not broaden into the transfer regime.

4.4 Which components carry the score

Table 1 gives the full picture. Three findings stand out.

The diagonal covariance is essential, not inert. Pinning the axis scales to 1 costs 0.360 paired ($8/8$ seeds) and drops the configuration below the reference to 0.830. The concern that a diagonal estimate from 256 samples in 704 dimensions would be pure noise is not borne out: even a badly under-sampled diagonal, carrying learned and anchored structure, is worth far more than the learned recombination residual and scalar gates that remain without it.

Path normalization is a correctness requirement, not a tuning choice. Removing the pair-consistent normalization of Equation 3 while keeping the mirrored draw costs 0.294 paired ($8/8$ seeds), the largest of the fixed-machinery ablations. Two configurations in Table 1 differ by more—the isotropic baseline at 0.360 and the unanchored strategy at 0.765—but both replace a representation or an architecture wholesale rather than removing a single mechanism, so they are not comparable effect sizes. The mechanism is visible in the trajectory: that configuration's return freezes at 2527.8 from roughly generation 500 onward, exactly the signature of a step-size controller receiving a persistently biased neutral signal. This is best read as a measurement of self-inflicted mis-specification rather than as the contribution of a component—which is precisely why we report the honest mirroring ablation

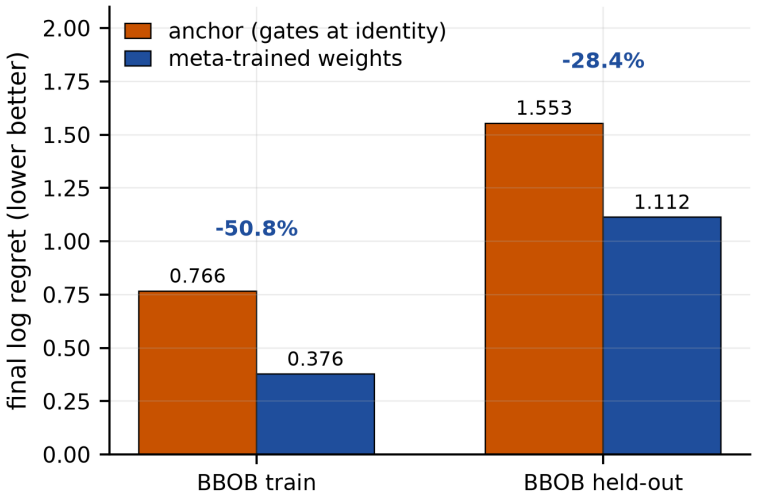


Figure 4: Meta-training succeeded on its own distribution. Final log regret of the frozen weights against the identity-gate anchor on the analytic meta-training functions and on held-out analytic functions (lower is better). The same frozen weights buy only a $+0.083$ paired increment on the 704-dimensional control transfer, and lose on final solution quality, which places the shortfall in transfer rather than in meta-learning.

Table 1: All seven configurations under the identical evaluator, budget, seeds, and scoring rule. Score is the normalized anytime area under the best-so-far curve at $2{,}000$ generations (pinned same-seed CMA-ES $= 1$), with standard error over eight seeds. Δ is the paired per-seed difference of the proposed configuration minus the row, with the number of seeds on which the proposed configuration is ahead. Wins is seeds above the pinned reference. G_{cma} is generations to match the reference's final return; a dash marks configurations that never reached it within $2{,}000$ generations.

Configuration	Score	Δ (paired)	Wins	Final return	G_{cma}
Proposed (anchored residual)	1.191 ± 0.019	—	8/8	3488.5	828
No stall re-expansion	1.185 ± 0.023	0.006 ± 0.026 (3/8)	8/8	3461.1	839
Identity-gate anchor	1.107 ± 0.028	0.083 ± 0.016 (8/8)	7/8	3709.2	1042
No mirrored sampling	1.094 ± 0.022	0.097 ± 0.024 (8/8)	8/8	3286.9	1509
No pair-consistent normalization	0.897 ± 0.021	0.294 ± 0.026 (8/8)	0/8	2527.8	—
Isotropic step-size only	0.830 ± 0.024	0.360 ± 0.019 (8/8)	0/8	2362.5	—
Unanchored learned strategy	0.426 ± 0.020	0.765 ± 0.021 (8/8)	0/8	2007.5	—
Pinned CMA-ES reference	1.000	—	—	3256.9	2000
Initial policy	0.000	—	—	497.4	—

separately. Removing mirroring itself, with the normalization consistently reverted, costs only 0.097 paired—the same order as the $+0.083$ learning increment.

The stall re-expansion is a null. Deleting it changes the score by 0.006 paired, with the proposed configuration ahead on only $3/8$ seeds and the paired standard error (0.026) four times the effect; the per-seed differences straddle zero, from -0.136 to $+0.084$. The mechanism's patience is 202.5 generations at $D = 704$, $N = 256$, and on the separate 400-generation analytic probe—run at the same $D = 704$, $N = 256$ scale—the stall counter peaks at 7 and the re-expansion never fires there. We did not instrument firings over the $2{,}000$-generation control runs, and we cannot infer that the mechanism was dormant there: the longest recorded stretch without a strict best-so-far improvement on those runs averages 275 generations and reaches 578, above the patience threshold. The negative result therefore rests on the paired delta alone—whether or not the re-expansion fired on Ant, deleting it does not move the score.

4.5 Anchoring as motivation, not as a controlled measurement

The unanchored learned strategy scores 0.426 and never reaches the reference on any seed, ending at 2007.5 return against the reference's 3256.9. The contrast with 1.191 is the intuition that motivated anchoring in the first place, and Figure 3a shows it is not a slow-start artifact: that configuration's anytime score rises monotonically and still ends far below 1. We deliberately decline to read the 0.765 gap as the causal effect of anchoring. That configuration differs from the proposed one in parameter count (142 versus 364), in module architecture, and in feature design at once; the parameter-matched unanchored comparison that would identify the effect was not run. What the run does establish cleanly is the opposite direction of the same question—that the anchor by itself already accounts for most of what the anchored system achieves.

5 Limitations and Threats to Validity

One meta-training run per configuration. Every configuration was meta-evolved once. The eight seeds are *evaluation* seeds inside that single run, so every standard error we report excludes meta-training-seed variance, which is plausibly the larger source. Reported dispersion is therefore a lower bound on the true uncertainty.

Comparisons confound mechanism with meta-training outcome. Every configuration, including the identity-gate control, was meta-evolved in its own separate run, and three of them—the isotropic step-size-only baseline and the no-mirroring and no-path-normalization ablations—landed at weights measurably different from the proposed configuration's: the recorded distance from the anchor point is 20.62 for the proposed configuration and 19.12, 13.50 and 9.90 for those three. For them, the reported delta mixes "this mechanism was removed" with "a different single meta-training run happened." The no-stall-re-expansion ablation is the exception among the configurations whose gates are active: its recorded distance is 20.62, matching the proposed configuration's to full precision, so for that row the confound does not apply and the delta is closest to a clean mechanism removal. We note this as an observation about the recorded weights and do not use it to argue anything about the other rows. The identity-gate control is immune to the confound by construction, because its meta-parameters provably cannot affect any update—which is another reason it, rather than the ablation table, carries the paper's claim.

One environment, one policy class, one budget regime. All transfer evidence comes from a single Brax Ant task with a tanh-linear policy at $D = 704$. We have shown the conclusion is sensitive to the budget and to the metric; we have not shown it is stable across environments, policy classes, or population sizes. The $2{,}000$-generation window also sits close to where the two curves cross, so the sign of the "learning helps" conclusion is not robust to a modest change in horizon.

Selection and reporting. The eight-seed protocol used to report the headline was also used to select the configuration during development, so the reported score carries a selection bias of unquantified size. The sign-test p-value we quote is direction-agnostic at $n = 8$ and is only interpretable next to the win count. Finally, the analytic probes and the control task are different measurement families, and we have kept them on separate axes rather than merging them into a single figure.

Interpretability of the learned part. Bounded multiplicative residuals around an identity make the learned component unusually legible for a learned optimizer—its worst-case effect is a factor of two on a named quantity—but we have not extracted a human-readable account of *what* the 364 parameters encode, and the gate diagnostics available in this study were recorded only for the configuration in which the gates are inert. Learned components embedded in otherwise transparent algorithms remain black boxes in the sense studied by the explainability literature (Guidotti et al., 2018; Arrieta et al., 2020), and the anchored design mitigates the risk rather than removing it.

6 Conclusion

We asked how much of a frozen-transferred learned evolution strategy's advantage is produced by meta-learning and how much by the classical algorithm it is anchored to, and we built the strategy so that the question has an exact answer. Against a pinned same-seed CMA-ES reference at 1.000, the full method scores 1.191 and the identity-gate anchor—the same code with every learned gate

provably pinned to one—scores 1.107. The anchor supplies 55.8% of the above-reference margin. The remaining increment is real and consistent across seeds, but it shrinks as the budget grows and reverses sign on final solution quality.

The negative result is specific and, we think, more useful than the positive one. Meta-training on cheap analytic functions clearly worked: it halved final log regret in-distribution and improved held-out analytic functions by nearly thirty percent. What did not survive was the transport of that gain to a 704-dimensional control policy, where it degraded into a budget-dependent head start. Anchoring is the design choice that bounds how far that failure can propagate—no gate can move a quantity by more than a factor of two in a generation, and the classical algorithm remains exactly recoverable as a control. It does not guarantee that the anchored system matches that algorithm, and here it did not: the full method finished 220.7 return below the identity-gate anchor. The one unanchored configuration we ran under the same protocol collapsed to 0.426, but it also differs in parameter count and architecture, so we report it as motivation rather than as a measurement of what anchoring contributes—an identification this study does not provide.

The methodological point generalizes beyond evolution strategies, and beyond this benchmark. Whenever a learned component is layered onto a competent classical procedure—and gradient-free optimizers of this family are deployed that way in settings from engineering design to variational quantum algorithms (Cerezo et al., 2021; Hamdani et al., 2019)—the honest experiment is the one that can switch the learned part off exactly, on the same code path, and report the two numbers side by side. When such a control is cheap to build, as it is for zero-initialized bounded residuals, there is little excuse for reporting only the headline.

REFERENCES

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. ISSN 1566-2535. doi: 10.1016/j.inffus.2019.12.012. URL http://dx.doi.org/10.1016/j.inffus.2019.12.012.

M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, August 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00348-9. URL http://dx.doi.org/10.1038/s42254-021-00348-9.

Saber Elsayed, Ruhul Sarker, Carlos A. Coello Coello, and Tapabrata Ray. Adaptation of operators and continuous control parameters in differential evolution for constrained optimization. *Soft Computing*, 22(19):6595–6616, 2017. ISSN 1433-7479. doi: 10.1007/s00500-017-2712-6. URL http://dx.doi.org/10.1007/s00500-017-2712-6.

Michael Emmerich, Ofer M. Shir, and Hao Wang. *Evolution Strategies*, pp. 89–119. Springer International Publishing, 2018. ISBN 9783319071244. doi: 10.1007/978-3-319-07124-4_13. URL http://dx.doi.org/10.1007/978-3-319-07124-4_13.

Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. Discovering representations for black-box optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, pp. 103–111. ACM, 2020. doi: 10.1145/3377930.3390221. URL http://dx.doi.org/10.1145/3377930.3390221.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes, 2019. URL https://www.repository.cam.ac.uk/handle/1810/300991.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):1–42, August 2018. ISSN 1557-7341. doi: 10.1145/3236009. URL http://dx.doi.org/10.1145/3236009.

Hamid Hamdani, Bouchaïb Radi, and Abdelkhalak El Hami. Optimization of solder joints in embedded mechatronic systems via kriging-assisted cma-es algorithm. *International Journal for Simulation and Multidisciplinary Design Optimization*, 10:A3, 2019. ISSN 1779-6288. doi: 10.1051/smdo/2019002. URL http://dx.doi.org/10.1051/smdo/2019002.

Nikolaus Hansen. The cma evolution strategy: A tutorial, 2016. URL https://arxiv.org/abs/1604.00772.

Essam H. Houssein, Mohamed H. Hassan, Mohamed A. Mahdy, and Salah Kamel. Development and application of equilibrium optimizer for optimal power flow calculation of power system. *Applied Intelligence*, 53(6):7232–7253, 2022. ISSN 1573-7497. doi: 10.1007/s10489-022-03796-7. URL http://dx.doi.org/10.1007/s10489-022-03796-7.

Lei Huang, Jie Qin, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Normalization techniques in training dnns: Methodology, analysis and application, 2020. URL https://arxiv.org/abs/2009.12836.

Saurav Jha, Dong Gong, Xuesong Wang, Richard E. Turner, and Lina Yao. The neural process family: Survey, applications and perspectives, 2022. URL https://arxiv.org/abs/2209.00517.

Leslie Pack Kaelbling, Michael L. Littman, and Andrew Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, May 1996. ISSN 1076-9757. doi: 10.1613/jair.301. URL http://dx.doi.org/10.1613/jair.301.

Robert Tjarko Lange, Tom Schaul, Yutian Chen, Chris Lu, Tom Zahavy, Valentin Dalibard, and Sebastian Flennerhag. Discovering attention-based genetic algorithms via meta-black-box optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, pp. 929–937. ACM, 2023. doi: 10.1145/3583131.3590496. URL http://dx.doi.org/10.1145/3583131.3590496.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks, 2018. URL https://arxiv.org/abs/1810.00825.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning, 2017. URL https://arxiv.org/abs/1707.09835.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference, 2021. URL https://arxiv.org/abs/2112.10510.

Mohammad Nabi Omidvar, Xiaodong Li, and Xin Yao. A review of population-based metaheuristics for large-scale black-box global optimization—part i. *IEEE Transactions on Evolutionary Computation*, 26(5):802–822, October 2022. ISSN 1941-0026. doi: 10.1109/tevc.2021.3130838. URL http://dx.doi.org/10.1109/tevc.2021.3130838.

Eneko Osaba, Esther Villar-Rodríguez, Javier Del Ser, Antonio J. Nebro, Daniel Molina, Antonio LaTorre, Ponnuthurai Nagaratnam Suganthan, Carlos A. Coello Coello, and Francisco Herrera. A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm and Evolutionary Computation*, 64:100888, 2021. ISSN 2210-6502. doi: 10.1016/j.swevo.2021.100888. URL http://dx.doi.org/10.1016/j.swevo.2021.100888.

Theodore Papamarkou, Jacob Hinkle, M. Todd Young, and David E. Womble. Challenges in markov chain monte carlo for bayesian neural networks. *Statistical Science*, 37(3), August 2022. ISSN 0883-4237. doi: 10.1214/21-sts840. URL http://dx.doi.org/10.1214/21-sts840.

Shahin Rostami and Alex Shenfield. A multi-tier adaptive grid algorithm for the evolutionary multi-objective optimisation of complex problems. *Soft Computing*, 21(17):4963–4979, 2016. ISSN 1433-7479. doi: 10.1007/s00500-016-2227-6. URL http://dx.doi.org/10.1007/s00500-016-2227-6.

Thomas Philip Rúnarsson and Xin Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, 2000. ISSN 1089-778X. doi: 10.1109/4235.873238. URL http://dx.doi.org/10.1109/4235.873238.

Volkmar Sauerland, Claudine von Hallern, Iris Kriest, and Julia Getzlaff. A cma-es algorithm allowing for random parameters in model calibration. *Journal of Advances in Modeling Earth Systems*, 15(8), August 2023. ISSN 1942-2466. doi: 10.1029/2022ms003390. URL http://dx.doi.org/10.1029/2022ms003390.

André Thomaser, Marc-Eric Vogt, Anna V. Kononova, and Thomas Bäck. *Transfer of Multi-objectively Tuned CMA-ES Parameters to a Vehicle Dynamics Problem*, pp. 546–560. Springer Nature Switzerland, 2023. ISBN 9783031272509. doi: 10.1007/978-3-031-27250-9_39. URL http://dx.doi.org/10.1007/978-3-031-27250-9_39.

Joaquin Vanschoren. *Meta-Learning*, pp. 35–61. Springer International Publishing, 2019. ISBN 9783030053185. doi: 10.1007/978-3-030-05318-5_2. URL http://dx.doi.org/10.1007/978-3-030-05318-5_2.

Teppei Yamaguchi, Kento Uchida, and Shinichi Shirakawa. Improvement of sep-cma-es for optimization of high-dimensional functions with low effective dimensionality. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1659–1668. IEEE, December 2022. doi: 10.1109/ssci51031.2022.10022244. URL http://dx.doi.org/10.1109/ssci51031.2022.10022244.

Qiang Yang, Wei–Neng Chen, Jeremiah D. Deng, Yun Li, Tianlong Gu, and Jun Zhang. A level-based learning swarm optimizer for large-scale optimization. *IEEE Transactions on Evolutionary Computation*, 22(4):578–594, August 2018. ISSN 1941-0026. doi: 10.1109/tevc.2017.2743016. URL http://dx.doi.org/10.1109/tevc.2017.2743016.

Zhi-Hui Zhan, Jian-Yu Li, Sam Kwong, and Jun Zhang. Learning-aided evolution for optimization. *IEEE Transactions on Evolutionary Computation*, 27(6):1794–1808, December 2023. ISSN 1941-0026. doi: 10.1109/tevc.2022.3232776. URL http://dx.doi.org/10.1109/tevc.2022.3232776.