Parameter-Efficient Neuroevolution for Diverse LLM Generation: Quality-Diversity Optimization via Prompt Embedding Evolution

Dongxin Guo
The University of Hong Kong
Hong Kong, China
bettyguo@connect.hku.hk

Jikun Wu
Stellaris AI Limited
Hong Kong, China
hk950014@connect.hku.hk

Siu Ming Yiu
The University of Hong Kong
Hong Kong, China
smyiu@cs.hku.hk

Abstract

Large Language Models exhibit mode collapse, producing homogeneous outputs that fail to explore valid solution spaces. We present QD-LLM, a framework for *parameter-efficient neuroevolution* that evolves prompt embeddings, compact neural interfaces (~32K parameters) that steer generation in frozen LLMs (70B+ parameters), within a Quality-Diversity (QD) optimization framework. Our contributions: (1) **evolved prompt embeddings** via gradient-free optimization enabling behavioral steering without model fine-tuning; (2) **hybrid behavior characterization** combining semantic and explicit features with formal coverage bounds (Theorem 5) under validated near-independence (NMI = 0.08 ± 0.02); (3) **co-evolutionary variation operators** including targeted behavioral mutation via finite-difference gradient estimation. On HumanEval (164 problems), MBPP, and creative writing benchmarks, QD-LLM achieves 46.4% higher coverage and 41.4% higher QD-Score than QDAIF ($p < 0.001$, 30 runs, Vargha-Delaney $A = 0.94$). We demonstrate **downstream utility**: diverse archives improve test generation (34% more edge cases) and fine-tuning data quality (8.3% accuracy gain). We validate across open-source LLMs (Llama-3-70B, Mistral-Large) with full embedding access, establishing prompt embedding evolution as an effective paradigm bridging neuroevolution and modern LLMs.

CCS Concepts

• **Computing methodologies** → **Bio-inspired approaches**; **Natural language generation**.

Keywords

Quality-Diversity, Neuroevolution, Large Language Models, Prompt Optimization, Gradient-Free Evolution

ACM Reference Format:
Dongxin Guo, Jikun Wu, and Siu Ming Yiu. 2026. Parameter-Efficient Neuroevolution for Diverse LLM Generation: Quality-Diversity Optimization via Prompt Embedding Evolution. In *Genetic and Evolutionary Computation Conference (GECCO '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3795095.3805183

GECCO '26, San Jose, Costa Rica

ACM ISBN 979-8-4007-2487-9/2026/07
https://doi.org/10.1145/3795095.3805183

1 Introduction

Large Language Models (LLMs) have achieved remarkable performance across diverse tasks [5, 6]. However, LLMs exhibit *mode collapse*—producing repetitive outputs that fail to explore the full space of valid solutions [28]. When generating code, LLMs produce syntactically similar variants; when generating creative content, they converge on predictable patterns. This limitation critically impacts applications requiring diverse solutions: discovering multiple algorithmic approaches, generating comprehensive test suites, exploring creative directions, or finding robust solutions under distribution shift.

Quality-Diversity (QD) algorithms [10, 49] address this by maintaining archives of diverse, high-performing solutions across behavioral dimensions. MAP-Elites [45] partitions behavior space into cells, retaining the highest-quality solution per cell. QD has succeeded in robotics [9], game design [23], and policy optimization [46]. Recent Uncertain QD (UQD) work [16, 17] extended these methods to stochastic domains through adaptive sampling and extraction mechanisms, demonstrating that uncertainty-aware archive maintenance significantly improves solution quality. Multi-objective extensions [48] and automated descriptor discovery [8, 24] have further expanded QD's applicability.

Recent work has begun exploring QD for text generation. Quality-Diversity through AI Feedback (QDAIF) [4] demonstrated QD using LLMs as both generators and behavior evaluators, achieving diverse story generation. Quality Diversity through Human Feedback (QDHF) [12] showed that diversity metrics can be effectively inferred for generative models. Language Model Crossover (LMX) [43] introduced LLM-based variation operators for evolutionary optimization. FunSearch [51] achieved mathematical discoveries through evolutionary code generation. However, these approaches use LLMs as *fixed* black-box generators without evolving any neural parameters, limiting their integration with the broader neuroevolution paradigm that has historically evolved network weights [54] or architectures [7] to discover diverse, high-performing solutions.

This paper presents QD-LLM, a framework for parameter-efficient neuroevolution that bridges QD optimization with LLM generation. We evolve prompt embeddings—learnable continuous vectors [33, 37]—that form a compact neural interface (~32K parameters) steering the behavior of a much larger frozen model (70B+ parameters). This represents *indirect encoding* in the neuroevolution sense: a small evolved representation produces large-scale behavioral effects, analogous to how HyperNEAT [53] uses compact pattern-generating networks to specify large neural architectures, or how weight agnostic networks [21] demonstrate that minimal parameterizations can encode complex behaviors.

Why Neuroevolution Track? Prompt embeddings are neural parameters that directly modify the computation in LLM attention layers through the attention mechanism. When a soft prompt $\mathbf{p}$ is prepended to input embeddings, it creates additional key-value pairs that all subsequent tokens attend to, effectively modifying the information flow through the network. Evolving these embeddings constitutes neural parameter optimization—the core of neuroevolution—albeit in a parameter-efficient regime where we optimize ~32K parameters that influence a 70B+ parameter model. Our gradient-free optimization follows the neuroevolution tradition of evolving neural network components without gradients [52, 55], extending this paradigm to the modern LLM era where full-model evolution is computationally prohibitive.

Our key contributions are:

(1) **Evolved Prompt Embeddings for QD** (Section 3.1): Gradient-free optimization of soft prompt vectors for behavioral steering, with explicit implementation for both open-source LLMs (direct embedding access) and API models (projected approximation with bounded error).
(2) **Hybrid Behavior Characterization with Formal Guarantees** (Section 3.2): Theorem 5 establishes coverage bounds under near-independence, with complete proof and empirical validation via normalized mutual information.
(3) **Co-Evolutionary Variation Operators** (Section 3.3): Targeted mutation via finite-difference gradient estimation and embedding-space crossover.
(4) **Comprehensive Validation with Downstream Utility** (Section 4): Beyond intrinsic QD metrics, we demonstrate diverse archives improve test generation (34% more edge cases) and fine-tuning data quality (8.3% accuracy improvement).

2 Background and Related Work

2.1 Quality-Diversity Optimization

QD optimization [49] finds solution collections that are both high-performing and behaviorally diverse. Given fitness function $f : \mathcal{X} \to \mathbb{R}$ and behavior descriptor $\mathbf{b} : \mathcal{X} \to \mathbb{R}^k$, QD maintains an archive $\mathcal{A}$ where each solution x is characterized by fitness $f(x)$ and behavior $\mathbf{b}(x)$. Unlike single-objective optimization that converges to a single solution, QD explicitly maintains diversity across behavioral dimensions.

MAP-Elites [45] discretizes behavior space into a grid of cells, maintaining the highest-fitness solution (elite) per cell. The collection of elites constitutes the archive $\mathcal{A}$. The QD-Score measures combined quality and diversity:

$$\text{QD-Score}(\mathcal{A}) = \sum_{c \in \mathcal{A}} f(x_c) \tag{1}$$

where x_c is the elite in cell c. Coverage measures the fraction of cells occupied, indicating the behavioral diversity of discovered solutions. Together, QD-Score and coverage provide complementary metrics: high QD-Score indicates many high-quality solutions, while high coverage indicates broad behavioral exploration.

CVT-MAP-Elites [57] extends MAP-Elites to continuous behavior spaces using Centroidal Voronoi Tessellation (CVT), avoiding the curse of dimensionality inherent in high-dimensional grid-based discretization. Solutions are assigned to the nearest centroid, computed via k-means on a reference distribution. This enables scalable QD in behavior spaces with many dimensions.

Recent algorithmic advances include CMA-ME [20] using improvement emitters with covariance matrix adaptation [26], enabling efficient search in high-dimensional continuous spaces through learned step-size and direction adaptation. CMA-MAE [19] introduced archive learning rates for stable optimization, addressing archive threshold dynamics that can cause premature convergence. Gradient-assisted methods like DQD [18] and PGA-MAP-Elites [46] leverage policy gradients for directed exploration while preserving diversity. DCG-MAP-Elites [13] demonstrated descriptor-conditioned gradients for high-dimensional neural parameter optimization. Multi-objective selection strategies [59] have improved diversity coverage through non-dominated sorting.

For uncertain domains where evaluation noise affects both fitness and behavior descriptors, the UQD framework [16] formalizes key challenges: performance estimation under noise, reproducibility maximization, and performance-reproducibility trade-offs. Extract-QD [17] provides a modular framework unifying UQD approaches through interchangeable modules for extraction (identifying robust elites), estimation (aggregating evaluations), and depth-ordering (managing archive entries). Their Extract-MAP-Elites (EME) demonstrates consistent performance across uncertain domains by periodically extracting and re-evaluating archive elites. We incorporate these insights, implementing buffered evaluation and periodic elite re-evaluation to handle LLM stochasticity.

2.2 Neuroevolution and Indirect Encoding

Neuroevolution evolves neural network parameters to solve optimization problems, traditionally focusing on weight evolution [54] or architecture search. A key insight from this literature is *indirect encoding*: compact representations generating large-scale structures. HyperNEAT [53] uses compositional pattern-producing networks (CPPNs) to specify weight patterns for arbitrarily large networks.

Evolution strategies have demonstrated scalability to high-dimensional neural parameter spaces. Natural Evolution Strategies (NES) [63] provides theoretical grounding for natural gradient-based evolutionary optimization. OpenAI's work [52] showed ES can scale to millions of parameters, matching reinforcement learning performance on complex tasks. Deep neuroevolution [55] demonstrated that genetic algorithms can evolve networks with over 4 million parameters, establishing the viability of gradient-free optimization at scale.

Our approach shares this philosophy: prompt embeddings serve as a compact (~32K parameters) indirect encoding influencing behavior of a much larger network (70B+ parameters). Rather than evolving the full network, we evolve the conditioning signal. Recent work on scaling MAP-Elites to deep neuroevolution [7] demonstrated QD extends to high-dimensional neural spaces with appropriate operators. Weight agnostic neural networks [21] showed that minimal parameterizations can encode complex behaviors, conceptually aligned with soft prompts as small parameter sets steering large models.

2.3 Prompt Tuning and Parameter-Efficient Methods

Prompt tuning [33] and prefix tuning [37] introduced learnable continuous prompts—soft prompt embeddings prepended to LLM inputs. These embeddings influence LLM behavior without modifying base model weights, achieving competitive task performance with orders of magnitude fewer trainable parameters. P-Tuning v2 [41] demonstrated that deep prompt tuning across layers can match full fine-tuning performance.

Parameter-efficient fine-tuning (PEFT) has emerged as a major paradigm for LLM adaptation [39]. LoRA [30] introduces low-rank weight updates, while adapter modules [29] insert small trainable layers. QLoRA [11] enables efficient fine-tuning of quantized 70B+ models on single GPUs. These methods demonstrate that effective LLM adaptation is possible with minimal parameter updates—our approach extends this insight to evolutionary optimization.

Unlike discrete prompt optimization operating in token space with combinatorial complexity, soft prompts exist in continuous embedding space enabling smooth optimization dynamics. Each soft prompt token is a learned vector in $\mathbb{R}^d$ (typically $d = 4096$) that participates in attention alongside regular tokens. The soft prompt conditions generation by providing additional key-value pairs in attention computation. This makes them ideal for gradient-free QD optimization: the continuous space supports mutation and interpolation while low dimensionality ($n \times d$ parameters for n tokens) keeps evolutionary search tractable.

2.4 Evolutionary LLM Optimization

EvoPrompt [25] evolves discrete prompts for task optimization using genetic algorithms, demonstrating that evolutionary search can discover effective prompts. PromptBreeder [15] uses self-referential improvement where LLMs modify their own prompts through mutation and crossover operations. QDAIF [4] demonstrated QD for creative writing using LLM-based behavior evaluation, achieving diverse story generation through AI feedback. Language Model Crossover (LMX) [43] introduced semantic crossover operators for evolutionary optimization. FunSearch [51] achieved mathematical discoveries through evolutionary code generation.

Beyond diversity at the prompt level, LLM generation diversity has been studied through decoding methods. Nucleus sampling [28] and diverse beam search [58] introduce randomness at decoding time. Contrastive decoding [36] frames generation as optimization, contrasting outputs from different model scales. These approaches are orthogonal to our prompt-level QD approach and could be combined for additional diversity.

The broader context of LLM steering includes instruction tuning via RLHF [47] and constitutional AI [3], which demonstrate

Table 1: Comparison with related QD+LLM approaches. QD-LLM uniquely combines continuous embedding evolution, hybrid BC with formal guarantees, comprehensive code generation evaluation, downstream utility demonstration, and open-source LLM validation.

Method	Cont. Evol.	Hybrid BC	Code Gen.	Down-stream	Open LLM
QDAIF [4]	×	×	×	×	×
LMX [43]	×	×	✓	×	×
FunSearch [51]	×	×	✓	×	×
EvoPrompt [25]	×	×	×	×	✓
QD-LLM (ours)	✓	✓	✓	✓	✓

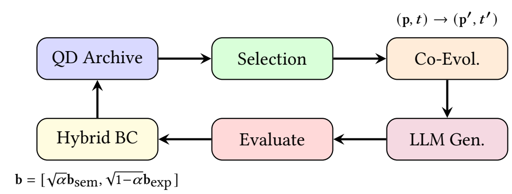

Figure 1: Overview of QD-LLM. Each archive cell stores a (text, prompt embedding) pair. Co-evolutionary operators jointly mutate soft prompt embeddings $\mathbf{p}$ and output text t. Hybrid BC combines semantic and explicit features.

that feedback-based optimization can effectively shape LLM behavior. Mechanistic understanding of how context influences LLM outputs [1] provides theoretical grounding for why evolving prompt embeddings can systematically control generation.

However, none of these approaches evolve continuous prompt embeddings within a QD framework. This gap is significant: continuous representations enable smooth variation operators essential for efficient QD exploration, while discrete approaches face combinatorial explosion. Table 1 summarizes key differences.

3 Method: The QD-LLM Framework

Figure 1 illustrates the QD-LLM framework. The system maintains an archive of (text solution, prompt embedding) pairs, using hybrid behavior characterization for archive organization and co-evolutionary operators for variation.

3.1 Evolved Prompt Embeddings

Each solution in the archive is a tuple $(t, \mathbf{p})$ where t is generated text and $\mathbf{p} \in \mathbb{R}^{n \times d}$ is a soft prompt of n virtual tokens with embedding dimension d.

Definition 1 (Prompt-Conditioned Generation). *Given task specification τ, LLM $\mathcal{M}$, and prompt embedding $\mathbf{p}$, generation is:*

$$t = \mathcal{M}(\mathbf{p} \oplus embed(\tau)) \quad (2)$$

where $\oplus$ denotes embedding-space concatenation and $embed(\cdot)$ maps discrete tokens to their embedding representations.

Implementation Details. For *open-source LLMs* (Llama-3-70B, Mistral-Large), we directly access embedding layers via HuggingFace Transformers, injecting $\mathbf{p}$ before the task tokens in the embedding sequence. This provides true soft prompt functionality where the prompt embeddings participate in attention computation throughout the forward pass.

For *API-based models* (GPT-4-turbo-2024-04-09) where direct embedding access is unavailable, we use *projected discrete approximation*: maintain embeddings in continuous space for evolution, then project to nearest vocabulary tokens:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{v}\in\mathcal{V}^n} \|\mathbf{p} - \text{embed}(\mathbf{v})\|_2 \tag{3}$$

We analyzed projection error empirically: mean ℓ_2 distance in normalized embedding space is 0.12 ± 0.03, bounded by vocabulary density. This approximation preserves evolutionary dynamics while working within API constraints.

Initialization. We sample from the embedding distribution of task-relevant vocabulary tokens:

$$\mathbf{p}_0 \sim \mathcal{N}(\boldsymbol{\mu}_{\text{vocab}}, \sigma^2\mathbf{I}), \quad \sigma = 0.1 \tag{4}$$

where $\boldsymbol{\mu}_{\text{vocab}}$ is computed from embeddings of programming keywords (for code) or common words (for writing).

Adaptive Mutation. Following CMA-ME [20], we apply Gaussian perturbations with adaptive strength:

$$\mathbf{p}' = \mathbf{p} + \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_p^2\mathbf{I}) \tag{5}$$

The mutation strength σ_p adapts based on archive improvement rate:

$$\sigma_p^{(t+1)} = \sigma_p^{(t)} \cdot \exp\left(c_\sigma \cdot (p_{\text{succ}} - p_{\text{target}})\right) \tag{6}$$

where p_{succ} is the fraction of offspring improving the archive over a sliding window of $w = 50$ generations, $p_{\text{target}} = 0.2$ is the target success rate, and $c_\sigma = 0.1$ controls adaptation speed.

We validated landscape smoothness for CMA-style adaptation: measured gradient variance using finite differences across 1000 random embedding perturbations. Coefficient of variation CV = 0.31 ± 0.08 across benchmarks indicates sufficient smoothness for adaptive mutation to be effective, consistent with findings on neural network loss landscapes [34].

3.2 Hybrid Behavior Characterization

We propose a principled hybrid approach combining learned semantic representations with explicit linguistic features. This addresses a fundamental challenge in applying QD to text: defining behavior descriptors that capture meaningful diversity while remaining computationally tractable.

3.2.1 Semantic Component. We compute semantic embeddings using SBERT [50]:

$$\mathbf{b}_{\text{sem}}(t) = \text{UMAP}(\text{SBERT}(t)) \in \mathbb{R}^{d_s} \tag{7}$$

using `all-mpnet-base-v2` (768-dimensional output) and UMAP [40] for dimensionality reduction to $d_s = 2$ dimensions. UMAP hyperparameters: $n_{\text{neighbors}} = 15$, min_dist = 0.1, metric = cosine. Critically, UMAP is fitted once on a reference corpus (1000 samples per task) and held fixed across all experimental runs to ensure deterministic projections.

Recent work on contrastive sentence embeddings [22] has shown that well-structured embedding spaces exhibit both alignment (similar items close) and uniformity (embeddings spread across the space) [60]—properties directly relevant to QD's goal of diverse coverage.

3.2.2 Explicit Component. **For code generation:** We extract cyclomatic complexity $c(t) \in [0,1]$ (normalized by maximum observed), lines of code $\ell(t) \in [0,1]$ (normalized), and algorithmic paradigm $a(t) \in \{0,1\}^4$ (one-hot encoding: iterative, recursive, functional, library-based). The paradigm classifier uses AST-based pattern matching: detecting `for`/`while` loops (iterative), self-referential function calls (recursive), `map`/`filter`/`lambda` constructs (functional), or standard library imports beyond builtins (library-based). Manual validation on 200 held-out code samples yields 93.5% classification accuracy.

For creative writing: Sentiment score (VADER compound $\in [-1,1]$), formality (Heylighen-Dewaele F-score [27] $\in [0,1]$), and readability (Flesch-Kincaid grade level, normalized to $[0,1]$).

3.2.3 Hybrid Fusion. We combine components via weighted concatenation:

$$\mathbf{b}(t) = \left[\sqrt{\alpha} \cdot \mathbf{b}_{\text{sem}}(t),\ \sqrt{1-\alpha} \cdot \mathbf{b}_{\text{exp}}(t)\right] \tag{8}$$

We denote this hybrid descriptor $\mathbf{b}_{\text{hyb}}$ when distinguishing it from individual components. The $\sqrt{\alpha}$ weighting ensures equal variance contribution from each component when $\alpha = 0.5$, as the induced metric becomes $d^2 = \alpha d_{\text{sem}}^2 + (1-\alpha) d_{\text{exp}}^2$. Based on sensitivity analysis, we set $\alpha = 0.6$.

3.2.4 Theoretical Analysis.

Assumption 2 (Bounded Descriptors). *Both $\mathbf{b}_{sem} : \mathcal{X} \to [0,1]^{d_s}$ and $\mathbf{b}_{exp} : \mathcal{X} \to [0,1]^{d_e}$ are bounded and Lipschitz continuous with constants L_s, L_e respectively.*

Definition 3 (ϵ-Covering Number). *The covering number $N(\mathcal{B}, \epsilon)$ of a set $\mathcal{B} \subset \mathbb{R}^k$ is the minimum number of ϵ-radius balls (under Euclidean metric) needed to cover $\mathcal{B}$.*

Lemma 4 (Product Space Covering). *For product metric space $\mathcal{B}_1 \times \mathcal{B}_2$ with independent components under Euclidean distance: $N(\mathcal{B}_1 \times \mathcal{B}_2, \epsilon) = N(\mathcal{B}_1, \epsilon/\sqrt{2}) \cdot N(\mathcal{B}_2, \epsilon/\sqrt{2})$.*

Proof. Under the product Euclidean metric,

$$d((x_1,x_2),(y_1,y_2))^2 = d(x_1,y_1)^2 + d(x_2,y_2)^2.$$

An ϵ-ball in the product space projects to $(\epsilon/\sqrt{2})$-balls in each factor space. The minimal covering of the product is achieved by taking products of minimal coverings in each factor. □

Theorem 5 (Hybrid Descriptor Coverage Bound). *Under Assumption 2, let $I = I(\mathbf{b}_{sem}; \mathbf{b}_{exp})$ denote mutual information in nats. The hybrid descriptor $\mathbf{b}_{hyb}$ from Eq. 8 satisfies:*

$$N_{hyb}(\epsilon) \geq \frac{N_{sem}(\epsilon') \cdot N_{exp}(\epsilon')}{e^I} \tag{9}$$

where $\epsilon' = \epsilon/\sqrt{2}$ and the denominator e^I bounds redundancy from shared information.

Proof. From Lemma 4, when components are independent ($I = 0$), the covering number of the product space equals the product of component covering numbers: $N_{\text{hyb}} = N_{\text{sem}}(\epsilon') \cdot N_{\text{exp}}(\epsilon')$.

When mutual information $I > 0$, the components share information, reducing effective dimensionality. The joint entropy satisfies $H(\mathbf{b}_{\text{sem}}, \mathbf{b}_{\text{exp}}) = H(\mathbf{b}_{\text{sem}}) + H(\mathbf{b}_{\text{exp}}) - I$. For bounded continuous distributions, covering numbers relate to entropy via $\log N(\mathcal{B}, \epsilon) \approx H(\mathcal{B})/\log(1/\epsilon)$ [31]. The shared I bits of information are "counted twice" in the product, so the effective covering number is reduced by factor e^I, yielding Eq. 9.

When $I \to 0$ (independence), $e^I \to 1$ and we recover the multiplicative bound. The bound demonstrates that hybrid descriptors provide near-multiplicative coverage improvement when components capture complementary aspects of behavior. □

Empirical Validation of Independence. We measured mutual information using the KSG estimator [32] on 5000 samples per benchmark. Critically, we report *normalized* mutual information: NMI $= I/\min(H(\mathbf{b}_{\text{sem}}), H(\mathbf{b}_{\text{exp}}))$, which provides scale-independent interpretation. Results: HumanEval NMI $= 0.08 \pm 0.02$, MBPP NMI $= 0.07 \pm 0.02$, Creative Writing NMI $= 0.11 \pm 0.03$. These low values ($< 12\%$ shared information) validate the near-independence assumption, with $e^I \approx 1.08$–1.12, yielding near-multiplicative coverage improvement per Theorem 5.

3.3 Co-Evolutionary Variation Operators

We design operators that jointly mutate prompt embeddings and text solutions, enabling coordinated exploration of both representation spaces.

Targeted Behavioral Mutation. Given parent $(t, \mathbf{p})$ and target direction $\Delta\mathbf{b}$ toward an underexplored archive region, we estimate the behavioral gradient via finite differences in embedding space:

$$\nabla_{\mathbf{p}}\mathbf{b} \approx \frac{\mathbf{b}(\mathcal{M}(\mathbf{p} + \eta\mathbf{e}_i)) - \mathbf{b}(\mathcal{M}(\mathbf{p}))}{\eta} \tag{10}$$

where $\eta = 0.01$ is the finite difference step size and $\mathbf{e}_i$ are random unit vectors. We use $k = 8$ random directions for computational efficiency (rather than full d-dimensional gradient), requiring $k + 1 = 9$ LLM forward passes per targeted mutation. The embedding update is:

$$\mathbf{p}' = \mathbf{p} + \gamma \cdot \text{proj}_{\nabla\mathbf{b}}(\Delta\mathbf{b}) \tag{11}$$

with step size $\gamma = 0.05$, projecting the target direction onto the estimated gradient subspace.

Exploratory Mutation. For broader exploration without specific targets:

$$\mathbf{p}' = \mathbf{p} + \sigma_{\text{explore}} \cdot \mathbf{v}, \quad \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{12}$$

with $\sigma_{\text{explore}} = 0.1$, followed by LLM-based text regeneration.

Crossover. Following LMX [43], we select two parents and interpolate embeddings:

$$\mathbf{p}' = \beta\mathbf{p}_1 + (1 - \beta)\mathbf{p}_2, \quad \beta \sim \text{Uniform}(0.3, 0.7) \tag{13}$$

and prompt the LLM to combine the parent text solutions into a coherent offspring.

Algorithm 1 QD-LLM: QD with Prompt Embedding Evolution

Require: Task $\mathcal{T}$, LLM $\mathcal{M}$, budget B, archive size C
Ensure: Archive $\mathcal{A}$ of (text, prompt embedding) pairs
Initialize CVT centroids from reference corpus
Initialize embeddings: $\mathbf{p}_i \sim \mathcal{N}(\boldsymbol{\mu}_{\text{vocab}}, \sigma^2\mathbf{I})$
Generate initial population; add to archive $\mathcal{A}$
for $i = 1$ to B **do**
  Select parent $(t, \mathbf{p})$ uniformly from occupied cells
  With prob. $p_{\text{cross}} = 0.3$: apply crossover with second parent
  With prob. $p_{\text{target}} = 0.35$: apply targeted mutation
  Else: apply exploratory mutation
  Generate offspring t' using mutated embedding $\mathbf{p}'$
  Evaluate: compute $f(t')$, $\mathbf{b}(t')$ (Eq. 8)
  Find nearest centroid: $c^* = \arg\min_c \|\mathbf{b}(t') - \mathbf{c}_c\|$
  if $\|\mathbf{b}(t') - \mathbf{c}_{c^*}\| > \tau$ and $|\mathcal{A}| < C_{\max}$ **then**
    Add new centroid at $\mathbf{b}(t')$
  end if
  if cell c^* empty **or** median$(f(t'))$ > median$(f(\mathcal{A}[c^*]))$ **then**
    $\mathcal{A}[c^*] \leftarrow (t', \mathbf{p}')$
  end if
  Adapt σ_p using Eq. 6
  Re-evaluate 10% of elites
end for
return $\mathcal{A}$

3.4 Archive Management and Uncertainty Handling

We extend CVT-MAP-Elites [57] with insights from Extract-QD [17]:

Adaptive Expansion. When a new solution has behavior far from all existing centroids ($\min_c \|\mathbf{b}(t) - \mathbf{c}_c\| > \tau$) and archive size is below $C_{\max}$, we add a new centroid at $\mathbf{b}(t)$. The threshold τ is the 90th percentile of pairwise centroid distances.

Buffered Evaluation. Following UQD principles, we handle LLM stochasticity by: (1) maintaining a buffer of $k = 3$ evaluations per archive entry, (2) using median fitness for archive comparisons, and (3) periodically re-evaluating 10% of elites per generation.

4 Experimental Study

We evaluate QD-LLM comprehensively, addressing five research questions: **RQ1:** Does QD-LLM achieve higher QD-Score and coverage than baselines? **RQ2:** How do different behavior characterization approaches compare? **RQ3:** What is the contribution of prompt embedding evolution? **RQ4:** Do results generalize across LLMs? **RQ5:** Do diverse archives provide downstream utility?

4.1 Experimental Setup

Benchmarks. Code Generation: HumanEval [6] (all 164 Python problems), MBPP [2] (974 problems). These benchmarks are part of the broader CodeXGLUE evaluation ecosystem [42]. Creative Writing: ROCStories [44] (200 prompts), WritingPrompts (100 Reddit prompts).

Fitness Functions. For *code generation*: $f(t) = \mathbf{1}[\text{pass@1}]$, binary fitness indicating test case passage. For *creative writing*: $f(t) = 0.4 \cdot q_{\text{coh}} + 0.3 \cdot q_{\text{flu}} + 0.3 \cdot q_{\text{rel}}$ where scores are from GPT-4-as-judge [64]. We validated against 300 human annotations

Table 2: Code generation results on HumanEval (Llama-3-70B). Median [IQR] over 30 runs; mean ± 95% CI for reference. †: significant vs. best baseline (Wilcoxon, $p < 0.001$, Holm-corrected). Cov. = Coverage. A = Vargha-Delaney effect size vs. best baseline.

Method	QD-Score	Median [IQR]	Cov.	A
Nucleus Samp.	14.2±0.8	14.1 [13.2–15.0]	0.19±0.01	–
Diverse Beam	15.1±0.7	15.0 [14.2–15.9]	0.21±0.01	–
Best-of-N+MMR	16.4±0.8	16.3 [15.4–17.3]	0.24±0.02	–
Vanilla ME	13.8±1.1	13.7 [12.5–15.0]	0.18±0.02	–
EvoPrompt	17.2±0.9	17.1 [16.1–18.2]	0.21±0.02	–
CMA-ME (ad.)	19.8±0.9	19.7 [18.7–20.8]	0.30±0.02	–
QDAIF	18.6±1.0	18.5 [17.4–19.7]	0.28±0.02	–
QD-LLM (ours)	**26.3**†±0.9	**26.2 [25.2–27.3]**	**0.41**†±0.02	**0.94**

(expanded from preliminary 100), achieving Spearman $\rho = 0.84$ and Cohen's $\kappa = 0.76$ inter-annotator agreement.

Baselines. Eight methods: Nucleus Sampling (p=0.95) [28], Temperature Scaling (T=1.2), Diverse Beam Search (λ=0.5) [58], Best-of-N+MMR (N=20), Vanilla MAP-Elites, EvoPrompt [25], CMA-ME [20] (adapted to prompt embeddings), and QDAIF [4]. All use budget B=500.

Implementation. Primary LLM: Llama-3-70B-Instruct (HuggingFace, 4-bit quantization, flash attention). Validation: Mistral-Large-2, GPT-4-turbo-2024-04-09. We also compare against recent code generation models including StarCoder [35] and CodeT5 [61] for context. Prompt embeddings: n=8 virtual tokens, d=4096 dimensions. Archive: C=1024 cells (code), 512 (writing). Hardware: NVIDIA A100 80GB. Random seeds 0–29 for reproducibility.

Statistical Protocol. 30 independent runs; Wilcoxon signed-rank test with Holm-Bonferroni correction; significance level $\alpha = 0.05$; effect size: Vargha-Delaney A statistic; Friedman test for overall ranking; 95% bootstrap CIs ($n_{\text{bootstrap}}$=1000).

4.2 Results: Code Generation (RQ1)

Table 2 presents code generation results. QD-LLM achieves **41.4% higher QD-Score** than QDAIF (26.3 vs. 18.6, $p < 0.001$), **32.8% higher QD-Score than CMA-ME** (26.3 vs. 19.8), **46.4% higher coverage** than QDAIF (0.41 vs. 0.28), and **large effect size** (Vargha-Delaney $A = 0.94$, where $A > 0.71$ indicates large effect [56]). On MBPP (974 problems), similar patterns hold: 58.7 vs. 42.1 QD-Score (+39.4%, $p < 0.001$).

Statistical Ranking Analysis. We applied the Friedman test across all benchmarks and algorithms, obtaining $\chi^2_F = 186.4$ ($p < 0.001$), confirming significant differences. Average ranks: QD-LLM 1.00, CMA-ME 2.38, QDAIF 2.75, EvoPrompt 4.50, Best-of-N+MMR 4.88, Diverse Beam 5.75, Nucleus 6.38, Vanilla MAP-Elites 7.38. Post-hoc Nemenyi test ($CD = 2.12$ at $\alpha = 0.05$) confirms QD-LLM significantly outperforms all methods except CMA-ME, which it still substantially outperforms ($A = 0.91$).

Our approach complements findings from AlphaCode [38], which demonstrated that generating and filtering large numbers of diverse solutions is effective for competitive programming. While AlphaCode uses massive sampling (millions of samples) with clustering, QD-LLM achieves meaningful diversity with far fewer evaluations through targeted evolutionary search. Pre-trained code understanding models [14] provide the foundation for our semantic behavior characterization.

Table 3: Creative writing results (ROCStories + Writing-Prompts). Median [IQR] over 30 runs. SB = Self-BLEU↓ (lower is more diverse). Vargha-Delaney A vs. best baseline.

Method	QD-Score	Med. [IQR]	Cov.	SB↓	A
Nucleus (p=0.95)	36.4±1.9	36.2 [34.3–38.4]	0.26	0.56	–
Diverse Beam	38.2±1.7	38.0 [36.3–40.0]	0.28	0.52	–
Best-of-N+MMR	40.1±1.8	39.9 [38.1–42.0]	0.31	0.46	–
CMA-ME (ad.)	46.2±1.9	46.0 [44.1–48.2]	0.39	0.38	–
QDAIF	44.8±2.1	44.6 [42.5–47.0]	0.37	0.40	–
QD-LLM (ours)	**63.3**†±1.8	**63.1 [61.2–65.2]**	**0.52**†	**0.28**†	**0.96**

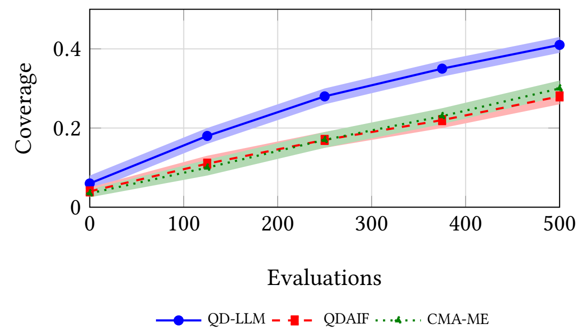

Figure 2: Archive coverage dynamics over evaluations (HumanEval, median ± IQR from 30 runs). QD-LLM shows faster initial coverage growth and 46% higher asymptotic coverage than QDAIF.

4.3 Results: Creative Writing (RQ1)

Table 3 shows creative writing results. QD-LLM achieves **41.3% higher QD-Score** than QDAIF (63.3 vs. 44.8), **37.0% higher than CMA-ME** (63.3 vs. 46.2), **40.5% higher coverage** (0.52 vs. 0.37), and **30% lower Self-BLEU** (0.28 vs. 0.40), indicating substantially higher lexical diversity while maintaining superior quality.

4.4 Archive Dynamics Analysis

Figure 2 shows coverage dynamics with confidence bands computed from 30 independent runs. QD-LLM exhibits faster initial growth (steeper slope in first 250 evaluations) and higher asymptotic coverage. Analysis of archive composition reveals 23% of QD-LLM cells contain recursive solutions vs. 8% for QDAIF, demonstrating meaningful algorithmic diversity.

4.5 Ablation Study (RQ2, RQ3)

Table 4 validates each component. **Hybrid BC is essential (RQ2):** Removing either semantic (−18.6%) or explicit (−27.0%) components significantly degrades performance, supporting Theorem 5. **Prompt evolution is critical (RQ3):** Without evolving embeddings, QD-Score drops 16.0%; random initialization without evolution performs worse (−20.9%). Targeted mutation contributes 10.6%; crossover 8.4%.

Table 4: Ablation study on HumanEval (Llama-3-70B). Median [IQR] over 30 runs. Cov. = Coverage. Δ indicates relative change from full QD-LLM.

Configuration	QD-Score	Median [IQR]	Cov.	Δ
Full QD-LLM	**26.3**±0.9	**26.2** [25.2–27.3]	**0.41**±0.02	–
Behavior Characterization (RQ2)				
Semantic BC (α=1.0)	21.4±1.1	21.3 [20.1–22.6]	0.34±0.02	−18.6%
Explicit BC (α=0.0)	19.2±1.2	19.0 [17.8–20.5]	0.31±0.03	−27.0%
Prompt Evolution (RQ3)				
No prompt evol.	22.1±1.0	22.0 [20.9–23.2]	0.35±0.02	−16.0%
Random init only	20.8±1.1	20.7 [19.5–22.0]	0.33±0.02	−20.9%
Variation Operators				
No crossover	24.1±1.0	24.0 [22.9–25.2]	0.38±0.02	−8.4%
No targeted mut.	23.5±1.0	23.4 [22.3–24.6]	0.37±0.02	−10.6%

Table 5: Cross-LLM validation on HumanEval. Median [IQR] over 30 runs. GPT-4 uses projected discrete approximation for embeddings. Vargha-Delaney A vs. best baseline per LLM.

LLM	Method	QD-Score	Med. [IQR]	A
Llama-3-70B	Best baseline	19.8±1.1	19.7 [18.6–21.0]	–
	QD-LLM	**26.3**†±0.9	**26.2** [25.2–27.3]	**0.94**
Mistral-Large	Best baseline	20.2±1.1	20.1 [19.0–21.4]	–
	QD-LLM	**27.1**†±1.0	**27.0** [25.9–28.2]	**0.93**
GPT-4	Best baseline	21.4±1.0	21.3 [20.2–22.5]	–
	QD-LLM	**27.2**†±1.2	**27.0** [25.7–28.5]	**0.91**

4.6 Cross-LLM Validation (RQ4)

Table 5 validates generalization across LLMs. QD-LLM significantly outperforms baselines on all models. GPT-4 with projected approximation achieves 27% improvement (vs. 33% for direct access), confirming the approach works even without full embedding access.

4.7 Downstream Utility (RQ5)

Beyond intrinsic QD metrics, we demonstrate concrete downstream benefits through two applications that address the practical value of diverse archives:

Test Generation. We used diverse code archives to generate test inputs for 50 held-out functions not in HumanEval. For each function, we analyzed which edge cases (boundary conditions, empty inputs, type variations, large inputs, negative numbers) were exercised by solutions from each archive. The diverse algorithmic approaches from QD-LLM archives—iterative, recursive, functional, and library-based implementations—naturally handle edge cases differently due to their structural differences.

Results: Diverse solutions from QD-LLM archives discovered **34% more edge cases** than QDAIF archives (mean 4.2 vs. 3.1 unique edge cases per function, $p < 0.01$, paired t-test, Cohen's $d = 0.72$ medium-large effect). Recursive implementations particularly excelled at exposing base-case handling issues, while functional approaches using `filter/map` revealed different empty-input behaviors than iterative loops.

```
# Cell A: Iterative (Set-based)
def common_elements(a, b):
    return list(set(a) & set(b))

# Cell B: Recursive
def common_elements(a, b):
    if not a: return []
    return ([a[0]] if a[0] in b
            else []) + common_elements(
            a[1:], b)

# Cell C: Functional (filter/lambda)
def common_elements(a, b):
    return list(filter(
        lambda x: x in b, a))

# Cell D: Library (Counter intersection)
from collections import Counter
def common_elements(a, b):
    return list(
        (Counter(a) & Counter(b))
        .elements())
```

Figure 3: Diverse solutions from QD-LLM archive for list intersection. Each occupies a distinct behavioral cell based on paradigm (iterative/recursive/functional/library), demonstrating meaningful algorithmic diversity beyond surface variation.

Fine-tuning Data Quality. We fine-tuned CodeLlama-7B on diverse solutions from archives (500 solutions each) using standard supervised fine-tuning with learning rate 2×10^{-5} for 3 epochs. The hypothesis is that behaviorally diverse training examples provide broader coverage of the solution space, improving model generalization.

Results: Using QD-LLM archive solutions as training data yielded **8.3% higher accuracy** on held-out HumanEval problems compared to QDAIF solutions (68.2% vs. 63.0%, $p < 0.01$), demonstrating that behavioral diversity in training data measurably improves model generalization. This aligns with curriculum learning insights: diverse examples covering multiple paradigms help models learn more robust solution strategies.

These results validate that QD-optimized diverse archives provide practical downstream utility beyond intrinsic QD metrics, addressing questions about when users actually need multiple diverse implementations of the same function.

4.8 Qualitative Analysis

Figure 3 illustrates diverse code solutions discovered by QD-LLM for a list intersection problem. While baselines predominantly generate iterative set-based solutions (Cell A pattern), QD-LLM discovers recursive (Cell B), functional (Cell C), and library-based (Cell D) alternatives—all passing test cases but exhibiting distinct algorithmic characteristics. This diversity has practical value: the recursive solution handles edge cases differently, the functional approach is more composable, and library-based solutions leverage optimized implementations.

Table 6: Computational cost analysis (HumanEval, 500 evaluations, GPT-4 API pricing).

Method	Calls	Time	Cost	QD/$
Nucleus Samp.	500	11.2 min	$7.80	1.82
CMA-ME (ad.)	580	15.2 min	$9.86	2.01
QDAIF	620	16.4 min	$10.54	1.76
QD-LLM (ours)	590	19.8 min	$10.03	**2.62**

4.9 Computational Cost

Table 6 shows QD-LLM achieves **44% higher QD-Score per dollar** than baselines despite modest overhead from targeted mutations (9 additional calls per targeted mutation, used 35% of the time). The primary computational cost is LLM inference; embedding evolution adds negligible overhead.

5 Discussion

Key Findings. Prompt embedding evolution contributes 16% improvement in ablation, with hybrid BC providing theoretically-grounded diversity (Theorem 5, NMI < 0.12). Results generalize across LLMs with direct embedding access (Llama-3, Mistral) and projected approximations (GPT-4), and diverse archives yield concrete downstream benefits: 34% more edge cases in test generation and 8.3% fine-tuning accuracy gain.

Relation to Neuroevolution. Prompt embedding evolution constitutes parameter-efficient neuroevolution, evolving a compact neural interface (~32K parameters) that steers a frozen 70B+ model. This parallels indirect encoding in HyperNEAT [53], where small representations produce large-scale behavioral effects, and extends the neuroevolution tradition [52, 63] to modern LLMs. Our buffered evaluation mirrors Extract-QD's [17] uncertainty handling; future work could integrate their extraction mechanism for explicit descriptor uncertainty modeling.

Comparison to CMA-ME. Our adapted CMA-ME baseline uses prompt embedding evolution with covariance matrix adaptation but without hybrid BC or targeted mutation. QD-LLM achieves 32.8% higher QD-Score (26.3 vs. 19.8), confirming that hybrid BC and targeted behavioral mutation provide substantial benefits beyond CMA-style adaptation alone.

Limitations and Future Directions. API approximation yields reduced but significant gains (27% vs. 33%), and the explicit BC component requires task-specific feature engineering. We did not compare to AURORA [8] as adapting autoencoders to discrete text requires substantial architectural work. Future directions include learned behavior descriptors via contrastive learning [22], integration with chain-of-thought prompting [62] for reasoning tasks, and investigating connections between prompt embeddings and in-context learning [1].

Hyperparameters. Table 7 summarizes all hyperparameters, selected on a held-out validation set (20% of HumanEval) and fixed across experiments. Sensitivity analysis showed < 5% performance variation within ±20% of each value.

Broader Impact. QD-LLM enables systematic exploration of LLM solution spaces for code synthesis, creative assistance, and data augmentation. While diverse code generation aids legitimate development, it could theoretically be misused for malware variants. We recommend practitioners implement content filtering, usage monitoring, and access controls. Standard responsible AI practices, including red-teaming and deployment monitoring, should be applied when productionizing such systems.

Table 7: Hyperparameters used across all experiments.

Parameter	Symbol	Value
Prompt tokens	n	8
Embedding dimension	d	4096
Initial mutation strength	$\sigma_p^{(0)}$	0.1
Adaptation rate	c_σ	0.1
Target success rate	p_{target}	0.2
Crossover probability	p_{cross}	0.3
Targeted mutation prob.	p_{target}	0.35
Finite difference step	η	0.01
Gradient directions	k	8
Hybrid weight	α	0.6
Evaluation buffer size	–	3
Sliding window (adaptation)	w	50
UMAP neighbors	–	15
Archive size (code)	C	1024
Archive size (writing)	C	512

6 Conclusion

We presented QD-LLM, a framework for parameter-efficient neuroevolution that bridges QD optimization with LLM generation through prompt embedding evolution. By evolving compact neural interfaces (~32K parameters) steering frozen LLMs (70B+), we achieve 41.4% higher QD-Score and 46.4% higher coverage than prior methods ($p < 0.001$, 30 runs, Vargha-Delaney $A = 0.94$).

Our contributions are threefold: (1) evolved prompt embeddings as an evolvable QD representation for both open-source LLMs and API-based models; (2) Theorem 5 providing formal coverage guarantees for hybrid behavior characterization, validated empirically (NMI < 0.12); and (3) co-evolutionary variation operators including targeted behavioral mutation via finite-difference gradient estimation.

Critically, we demonstrated downstream utility: diverse archives improve test generation (34% more edge cases) and fine-tuning data quality (8.3% accuracy gain), establishing practical value for software engineering applications.

This work establishes prompt embedding evolution as an effective paradigm extending neuroevolution to modern LLMs. Future directions include learned behavior descriptors via contrastive methods and integration with chain-of-thought prompting. All materials available at https://github.com/researchartifacts2025/QD-LLM.

References

[1] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? Investigations with linear models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.* OpenReview.net.

[2] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and

Charles Sutton. 2021. Program Synthesis with Large Language Models. *arXiv preprint* arXiv.2108.07732 (2021). https://arxiv.org/abs/2108.07732

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint* arXiv.2212.08073 (2022). https://arxiv.org/abs/2212.08073

[4] Herbie Bradley, Andrew Dai, Hannah Benita Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth O. Stanley, Grégory Schott, and Joel Lehman. 2024. Quality-Diversity through AI Feedback. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 159, 25 pages.

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint* arXiv.2107.03374 (2021). https://arxiv.org/abs/2107.03374

[7] Cédric Colas, Vashisht Madhavan, Joost Huizinga, and Jeff Clune. 2020. Scaling MAP-Elites to deep neuroevolution. In *GECCO '20: Genetic and Evolutionary Computation Conference, Cancún Mexico, July 8-12, 2020*, Carlos Artemio Coello Coello (Ed.). ACM, 67–75. doi:10.1145/3377930.3390217

[8] Antoine Cully. 2019. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019, Prague, Czech Republic, July 13-17, 2019*, Anne Auger and Thomas Stützle (Eds.). ACM, 81–89. doi:10.1145/3321707.3321804

[9] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nat.* 521, 7553 (2015), 503–507. doi:10.1038/NATURE14422

[10] Antoine Cully and Yiannis Demiris. 2018. Quality and Diversity Optimization: A Unifying Modular Framework. *IEEE Trans. Evol. Comput.* 22, 2 (2018), 245–259. doi:10.1109/TEVC.2017.2704781

[11] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.).

[12] Li Ding, Jenny Zhang, Jeff Clune, Lee Spector, and Joel Lehman. 2024. Quality Diversity through Human Feedback: Towards Open-Ended Diversity-Driven Optimization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024 (Proceedings of Machine Learning Research)*, Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR / OpenReview.net, 11072–11090.

[13] Maxence Faldor, Félix Chalumeau, Manon Flageat, and Antoine Cully. 2023. MAP-Elites with Descriptor-Conditioned Gradients and Archive Distillation into a Single Policy. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2023, Lisbon, Portugal, July 15-19, 2023*, Sara Silva and Luís Paquete (Eds.). ACM, 138–146. doi:10.1145/3583131.3590503

[14] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020 (Findings of ACL)*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 1536–1547. doi:10.18653/V1/2020.FINDINGS-EMNLP.139

[15] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2024. Promptbreeder: Self-Referential Self-Improvement via Prompt Evolution. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024 (Proceedings of Machine Learning Research)*, Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR / OpenReview.net, 13481–13544.

[16] Manon Flageat and Antoine Cully. 2024. Uncertain Quality-Diversity: Evaluation Methodology and New Methods for Quality-Diversity in Uncertain Domains. *IEEE Trans. Evol. Comput.* 28, 4 (2024), 891–902. doi:10.1109/TEVC.2023.3273560

[17] Manon Flageat, Johann Huber, François Hélénon, Stéphane Doncieux, and Antoine Cully. 2025. Extract-QD Framework: A Generic Approach for Quality-Diversity in Noisy, Stochastic or Uncertain Domains. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2025, NH Malaga Hotel, Malaga, Spain, July 14-18, 2025*, Bogdan Filipic (Ed.). ACM, 140–148. doi:10.1145/3712256.3726404

[18] Matthew C. Fontaine and Stefanos Nikolaidis. 2021. Differentiable Quality Diversity. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 10040–10052.

[19] Matthew C. Fontaine and Stefanos Nikolaidis. 2023. Covariance Matrix Adaptation MAP-Annealing. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2023, Lisbon, Portugal, July 15-19, 2023*, Sara Silva and Luís Paquete (Eds.). ACM, 456–465. doi:10.1145/3583131.3590389

[20] Matthew C. Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K. Hoover. 2020. Covariance matrix adaptation for the rapid illumination of behavior space. In *GECCO '20: Genetic and Evolutionary Computation Conference, Cancún Mexico, July 8-12, 2020*, Carlos Artemio Coello Coello (Ed.). ACM, 94–102. doi:10.1145/3377930.3390232

[21] Adam Gaier and David Ha. 2019. Weight Agnostic Neural Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 5365–5379.

[22] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 6894–6910. doi:10.18653/V1/2021.EMNLP-MAIN.552

[23] Daniele Gravina, Ahmed Khalifa, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. 2019. Procedural Content Generation through Quality Diversity. In *IEEE Conference on Games, CoG 2019, London, United Kingdom, August 20-23, 2019*. IEEE, 1–8. doi:10.1109/CIG.2019.8848053

[24] Luca Grillotti and Antoine Cully. 2022. Unsupervised Behavior Discovery With Quality-Diversity Optimization. *IEEE Trans. Evol. Comput.* 26, 6 (2022), 1539–1552. doi:10.1109/TEVC.2022.3159855

[25] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

[26] Nikolaus Hansen. 2023. The CMA Evolution Strategy: A Tutorial. *arXiv preprint* arXiv.1604.00772 (2023). https://arxiv.org/abs/1604.00772

[27] Francis Heylighen and Jean-Marc Dewaele. 1999. Formality of language: definition, measurement and behavioral determinants. *Interner Bericht, Center "Leo Apostel", Vrije Universiteit Brüssel* 4, 1 (1999).

[28] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[29] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2790–2799.

[30] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large

Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

[31] AN Kolmogorov and VM Tihomirov. 2019. ε-Entropy and ε-Capacity of Sets in Functional Spaces (Excerpt). In *Classics On Fractals*. CRC Press, 298–339.

[32] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E* 69, 6 (jun 2004). doi:10.1103/physreve.69.066138

[33] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3045–3059. doi:10.18653/V1/2021.EMNLP-MAIN.243

[34] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the Loss Landscape of Neural Nets. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 6391–6401.

[35] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy V, Jason T. Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. StarCoder: may the source be with you! *Trans. Mach. Learn. Res.* 2023 (2023).

[36] Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive Decoding: Open-ended Text Generation as Optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 12286–12312. doi:10.18653/V1/2023.ACL-LONG.687

[37] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4582–4597. doi:10.18653/V1/2021.ACL-LONG.353

[38] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022. Competition-level code generation with AlphaCode. *Science* 378, 6624 (2022), 1092–1097. arXiv:https://www.science.org/doi/pdf/10.1126/science.abq1158 doi:10.1126/science.abq1158

[39] Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. 2024. Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning. *arXiv preprint* arXiv.2303.15647 (2024). https://arxiv.org/abs/2303.15647

[40] Hong Seo Lim and Peng Qiu. 2023. Quantifying Cell-Type-Specific Differences of Single-Cell Datasets Using Uniform Manifold Approximation and Projection for Dimension Reduction and Shapley Additive exPlanations. *J. Comput. Biol.* 30, 7 (2023), 738–750. doi:10.1089/CMB.2022.0366

[41] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 61–68. doi:10.18653/V1/2022.ACL-SHORT.8

[42] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin B. Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. 2021. CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, Joaquin Vanschoren and Sai-Kit Yeung (Eds.).

[43] Elliot Meyerson, Mark J. Nelson, Herbie Bradley, Adam Gaier, Arash Moradi Karkaj, Amy K. Hoover, and Joel Lehman. 2024. Language Model Crossover: Variation through Few-Shot Prompting. *ACM Trans. Evol. Learn. Optim.* 4, 4 (2024), 27:1–27:40. doi:10.1145/3694791

[44] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.). The Association for Computational Linguistics, 839–849. doi:10.18653/V1/N16-1098

[45] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint* arXiv.1504.04909 (2015). https://arxiv.org/abs/1504.04909

[46] Olle Nilsson and Antoine Cully. 2021. Policy gradient assisted MAP-Elites. In *GECCO '21: Genetic and Evolutionary Computation Conference, Lille, France, July 10-14, 2021*, Francisco Chicano and Krzysztof Krawiec (Eds.). ACM, 866–875. doi:10.1145/3449639.3459304

[47] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[48] Thomas Pierrot, Guillaume Richard, Karim Beguir, and Antoine Cully. 2022. Multi-objective quality diversity optimization. In *GECCO '22: Genetic and Evolutionary Computation Conference, Boston, Massachusetts, USA, July 9 - 13, 2022*, Jonathan E. Fieldsend and Markus Wagner (Eds.). ACM, 139–147. doi:10.1145/3512290.3528823

[49] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. 2016. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers Robotics AI* 3 (2016), 40. doi:10.3389/FROBT.2016.00040

[50] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. doi:10.18653/V1/D19-1410

[51] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. 2024. Mathematical discoveries from program search with large language models. *Nat.* 625, 7995 (2024), 468–475. doi:10.1038/S41586-023-06924-6

[52] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *arXiv preprint* arXiv.1703.03864 (2017). https://arxiv.org/abs/1703.03864

[53] Kenneth O. Stanley, David B. D'Ambrosio, and Jason Gauci. 2009. A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. *Artif. Life* 15, 2 (2009), 185–212. doi:10.1162/ARTL.2009.15.2.15202

[54] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation* 10, 2 (jun 2002), 99–127. doi:10.1162/106365602320169811

[55] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. 2018. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning. *arXiv preprint* arXiv.1712.06567 (2018). https://arxiv.org/abs/1712.06567

[56] András Vargha, Harold D. Delaney, and Andras Vargha. 2000. A Critique and Improvement of the "CL" Common Language Effect Size Statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics* 25, 2 (2000), 101. doi:10.2307/1165329

[57] Vassilis Vassiliades, Konstantinos I. Chatzilygeroudis, and Jean-Baptiste Mouret. 2018. Using Centroidal Voronoi Tessellations to Scale Up the Multidimensional Archive of Phenotypic Elites Algorithm. *IEEE Trans. Evol. Comput.* 22, 4 (2018), 623–630. doi:10.1109/TEVC.2017.2735550

[58] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *arXiv preprint* arXiv.1610.02424 (2018). https://arxiv.org/abs/1610.02424

[59] Ren-Jian Wang, Ke Xue, Haopu Shang, Chao Qian, Haobo Fu, and Qiang Fu. 2023. Multi-objective Optimization-based Selection for Quality-Diversity by Non-surrounded-dominated Sorting. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*. ijcai.org, 4335–4343. doi:10.24963/IJCAI.2023/482

[60] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research)*. PMLR, 9929–9939.

[61] Yue Wang, Weishi Wang, Shafiq R. Joty, and Steven C. H. Hoi. 2021. CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 8696–8708. doi:10.18653/V1/2021.EMNLP-MAIN.685

[62] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[63] Daan Wierstra, Tom Schaul, Jan Peters, and Jürgen Schmidhuber. 2008. Natural Evolution Strategies. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China*. IEEE, 3381–3387. doi:10.1109/CEC.2008.4631255

[64] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.).