Predicting When LLMs Enhance Evolutionary Optimization: A Landscape-Aware Framework

Dongxin Guo*, Jikun Wu†, and Siu Ming Yiu*
Department of Computer Science, The University of Hong Kong, Hong Kong, China*
Brain Investing Limited, Hong Kong, China†
Emails: bettyguo@connect.hku.hk, hk950014@connect.hku.hk, smyiu@cs.hku.hk

***Abstract*—Large Language Models (LLMs) have emerged as promising components in evolutionary optimization, yet understanding when and why LLMs improve optimization performance remains a fundamental open question. This paper establishes a *landscape-aware prediction framework* connecting fitness landscape properties to LLM-assisted optimization effectiveness through the lens of search bias alignment. We introduce the concept of *LLM-induced search bias* and formalize its alignment with problem landscape structure through a novel *landscape-operator compatibility function* κ. Building upon classical autocorrelation analysis and fitness-distance correlation theory, we establish three analytical results: (1) necessary landscape conditions for positive LLM contribution with theoretical justification, (2) a convergence characterization under verifiable drift conditions, and (3) an information-theoretic bound on maximum optimization acceleration. Comprehensive multi-dimensional validation across $d \in \{10, 30, 50, 100\}$ and two benchmark suites (BBOB and CEC 2022) demonstrates that our prediction framework achieves 93.1% overall accuracy (Pearson $r = 0.91$, $p < 0.001$) across 144 benchmark configurations, validated against six state-of-the-art baselines including CMA-ES, L-SHADE, jSO, and NL-SHADE-RSP. Multi-LLM validation across GPT-4-turbo, GPT-3.5-turbo, and Claude-3-Sonnet confirms framework generalization. This work provides both theoretical understanding and practical deployment guidelines for LLM-assisted evolutionary optimization.**

***Index Terms*—fitness landscape analysis, large language models, evolutionary optimization, search bias alignment, algorithm selection**

I. Introduction

The integration of Large Language Models (LLMs) into evolutionary computation has sparked considerable research interest, with numerous studies demonstrating empirical improvements across various optimization domains [1]–[3]. However, a critical gap persists: the field lacks principled frameworks explaining *when*, *why*, and *under what conditions* LLMs contribute positively to optimization performance. This gap hinders principled algorithm design and limits our ability to predict LLM effectiveness on new problem instances.

Fitness landscape theory provides rigorous mathematical tools developed over three decades for characterizing optimization problems [4], [5]. Concepts such as ruggedness [4], fitness-distance correlation (FDC) [6], and local optima networks [7] have proven invaluable for understanding algorithm behavior. Meanwhile, the No Free Lunch (NFL) theorems [8] establish that algorithm effectiveness depends fundamentally on alignment between search biases and problem structure.

Despite these mature theoretical frameworks, their application to LLM-assisted optimization remains unexplored. The comprehensive survey by Wu et al. [9] explicitly identifies the absence of convergence proofs, complexity analysis, and theoretical limits for LLM-enhanced evolutionary algorithms as critical open problems. This paper addresses these gaps by establishing a prediction framework connecting fitness landscape properties to LLM operator effectiveness, extending foundational work on evolutionary transfer optimization [10], [11].

Our contributions are fourfold:

- We introduce the concept of *LLM-induced search bias* and formalize a *landscape-operator compatibility function* κ that quantifies alignment between LLM suggestions and problem structure.
- We establish three analytical results: necessary landscape conditions for positive LLM contribution, convergence characterization under verifiable drift conditions, and information-theoretic acceleration bounds.
- We provide comprehensive **multi-dimensional validation** across $d \in \{10, 30, 50, 100\}$ on BBOB and CEC 2022 against six baselines with **93.1% prediction accuracy**.
- We validate framework generalization across multiple LLMs and empirically validate our gradient alignment assumption.

Code is available at https://github.com/airesearchrepo2025/LLM-EA-Landscape.

II. Preliminaries and Related Work

A. Fitness Landscape Fundamentals

A fitness landscape is formally defined as a triple $\mathcal{L} = (X, N, f)$, where X is the search space, $N : X \to 2^X$ defines the neighborhood structure, and $f : X \to \mathbb{R}$ is the fitness function [5], [12]. This formalization enables rigorous analysis of optimization problem structure [13], [14].

Definition 1 (Autocorrelation Function)**.** *For a random walk $\{x_0, x_1, \ldots\}$ on landscape $\mathcal{L}$, the autocorrelation function is defined as [4]:*

$$\rho(s) = \frac{\mathbb{E}[(f(x_t) - \bar{f})(f(x_{t+s}) - \bar{f})]}{\mathrm{Var}(f)} \tag{1}$$

where $\bar{f}$ denotes the mean fitness over the search space.

The correlation length $\tau = -1/\ln(\rho(1))$ provides a scalar measure of landscape ruggedness [4], [15]. Large τ indicates smooth landscapes amenable to gradient-following search strategies, while small τ indicates rugged landscapes requiring population diversity.

Definition 2 (Fitness-Distance Correlation)**.** *The fitness-distance correlation (FDC) [6] is defined as:*

$$\text{FDC} = \frac{\text{Cov}(f, d)}{\sigma_f \cdot \sigma_d} \tag{2}$$

where $d(x) = \min_{x^* \in X^*} \|x - x^*\|$ is the distance to the nearest global optimum.

FDC values classify problem difficulty: $\text{FDC} > 0.15$ indicates straightforward problems, $|\text{FDC}| \leq 0.15$ indicates unpredictable difficulty, and $\text{FDC} < -0.15$ signals deceptive landscapes [6], [16].

Exploratory Landscape Analysis (ELA) [17] provides comprehensive feature-based characterization through approximately 50 features, enabling numerical landscape characterization from samples [18], [19].

B. LLM-Assisted Evolutionary Optimization

Recent works have integrated LLMs into evolutionary optimization through several paradigms. OPRO [1] uses LLMs as optimizers through in-context learning. EvoPrompt [2] employs LLMs as crossover and mutation operators. FunSearch [3] combines LLMs with evolutionary search to discover new mathematical constructions. Meyerson et al. [20] demonstrate that LLMs naturally produce intelligent variation operators. Algorithm discovery approaches [21]–[23] treat algorithms themselves as evolvable entities.

Despite this empirical progress, Wu et al. [9] note that no convergence proofs exist for LLM-enhanced EAs, no complexity analysis of LLM operators has been conducted, and the theoretical limits for when LLMs help remain unknown.

Key distinction from algorithm selection: Classical ELA-based algorithm selection [19], [24] trains classifiers to choose from portfolios of existing algorithms. Our framework specializes this paradigm to a single binary deployment decision—should an LLM component be integrated into an EA?—and grounds it in a theoretical compatibility function κ rather than purely empirical classification.

III. Theoretical Framework

A. LLM-Induced Search Bias

Definition 3 (LLM-Induced Search Bias)**.** *Let $\mathcal{A}_{\text{LLM}} : X^k \to \Delta(X)$ denote an LLM-based operator that maps k parent solutions to a probability distribution over the search space. The LLM-induced search bias is defined as:*

$$\beta_{\text{LLM}}(x) = \mathcal{A}_{\text{LLM}}(P)(x) - \mathcal{U}(x) \tag{3}$$

where $P \in X^k$ is the parent population and $\mathcal{U}$ is the uniform distribution over X.

This formalization captures the deviation of LLM suggestions from unbiased random sampling. Unlike classical mutation operators with mathematically specified distributions, LLM operators induce implicit biases derived from pre-training on natural language and code corpora [22], [23].

Assumption 1 (Ergodic LLM Sampling)**.** *The LLM operator, when queried multiple times with the same context, produces samples from a stationary distribution $p_\theta(x|P)$ with finite variance.*

Lemma 1 (Bias Decomposition)**.** *Under Assumption 1, the LLM-induced search bias admits the decomposition:*

$$\beta_{\text{LLM}}(x) = \beta_{\text{sem}}(x) + \beta_{\text{ctx}}(x|P) + \epsilon(x) \tag{4}$$

where β_{sem} represents semantic bias from pre-training, β_{ctx} represents context-dependent bias from the prompt, and ϵ is stochastic noise with $\mathbb{E}[\epsilon(x)] = 0$.

B. Landscape-Operator Compatibility

Definition 4 (Landscape-Operator Compatibility)**.** *The compatibility between landscape $\mathcal{L}$ and operator $\mathcal{A}$ is defined as:*

$$\kappa(\mathcal{L}, \mathcal{A}) = \mathbb{E}_{x \sim \mathcal{A}(P)}[f(x)] - \mathbb{E}_{x \sim \mathcal{U}}[f(x)] \tag{5}$$

where the expectation is over offspring generated by operator $\mathcal{A}$ versus uniform random sampling.

Positive compatibility $\kappa > 0$ indicates the operator produces offspring with higher expected fitness than random sampling. This definition extends the notion of operator quality in evolutionary algorithm theory [25], [26] and parallels the usefulness measure in transfer optimization [27].

Definition 5 (Semantic Structure Coefficient)**.** *For a fitness landscape $\mathcal{L}$ with representation $\phi : X \to \Sigma^*$ mapping solutions to string descriptions, the semantic structure coefficient is:*

$$\gamma(\mathcal{L}) = \frac{I(f(x); \phi(x))}{H(f(x))} \tag{6}$$

where $I(\cdot;\cdot)$ denotes mutual information and $H(\cdot)$ denotes entropy.

High γ indicates fitness is highly predictable from semantic representations. Since computing γ directly is intractable, we estimate it from ELA features:

$$\hat{\gamma}(\mathcal{L}) = 0.4 \cdot r^2_{\text{lin}} + 0.3 \cdot (1 - |s_{\text{distr}}|) + 0.3 \cdot q_{\text{lda}} \tag{7}$$

where r^2_{lin}, s_{distr}, and q_{lda} are standard ELA features [18]. Weights were calibrated via leave-one-function-out cross-validation on BBOB (mean CV accuracy 91.2%); the overall 93.1% combines BBOB leave-one-out with CEC 2022 as a fully held-out suite.

IV. Main Results

Assumption 2 (Exponential Autocorrelation Decay)**.** *The landscape $\mathcal{L}$ satisfies exponential autocorrelation decay, i.e., $\rho(s) \approx \exp(-s/\tau)$ for $s \geq 1$. This holds for NK-landscapes [28], quadratic functions, and BBOB functions with bounded conditioning [29].*

Algorithm 1 Estimation of Semantic Structure Coefficient

Require: Fitness function f, dimension d, sample budget $n = 50d$
Ensure: Estimated $\hat{\gamma}$
1: Sample $S = \{(x_i, f(x_i))\}_{i=1}^{n}$ via Latin Hypercube
2: Compute ELA features using `pflacco`:
3: $r^2 \leftarrow$ `ela_meta.lin_simple.r2`(S)
4: $s \leftarrow$ `ela_distr.skewness`(S)
5: $q \leftarrow$ `ela_level.lda_qda`(S)
6: $\hat{\gamma} \leftarrow 0.4 \cdot r^2 + 0.3 \cdot (1 - |s|) + 0.3 \cdot q$
7: **return** $\text{clip}(\hat{\gamma}, 0, 1)$

Assumption 3 (Gradient-Aligned LLM Bias)**.** *The LLM operator, when provided with fitness-improving examples in context, produces offspring biased toward the direction of fitness improvement:*

$$\mathbb{E}[\langle \nabla_x \beta_{\text{LLM}}(x), \nabla_x f(x) \rangle] \geq 0 \tag{8}$$

This assumption is motivated by in-context learning theory [1] and empirically validated in Section V-D.

Theorem 2 (Necessary Conditions for Positive LLM Contribution)**.** *Under Assumptions 1, 2, and 3, for an LLM operator $\mathcal{A}_{\text{LLM}}$ to achieve positive contribution $\kappa(\mathcal{L}, \mathcal{A}_{\text{LLM}}) > 0$ on landscape $\mathcal{L}$, the following conditions are necessary:*

1) ***Sufficient smoothness**: $\tau > \tau_{\min}$ where $\tau_{\min} = c_1 \cdot \ln(d)$ for constant $c_1 > 0$*
2) ***Non-deceptiveness**: $\text{FDC} > -\epsilon$ for $\epsilon = 0.15$*
3) ***Semantic predictability**: $\hat{\gamma}(\mathcal{L}) > \gamma_{\min}$*

Proof sketch. Each condition is necessary because its violation implies $\kappa \leq 0$:

Condition 1: For $\tau < \tau_{\min}$, by the data processing inequality [30], the mutual information between parent fitnesses and offspring fitness satisfies $I(f(P); f(x_{\text{offspring}})) \leq k \cdot \exp(-\sqrt{d}/(c_1 \ln d)) \cdot H(f) = o(1)$, so the LLM cannot extract useful gradient information.

Condition 2: For $\text{FDC} < -0.15$, the fitness gradient points away from optima. Under Assumption 3, the LLM systematically moves away from global optima.

Condition 3: For $\gamma < \gamma_{\min}$, insufficient information exists in semantic representations for better-than-random guidance. □

Practical thresholds: We use $\tau_{\min} = 3$ which provides a conservative margin across all tested dimensions.

Theorem 3 (Convergence Under Drift Conditions)**.** *Consider an LLM-augmented $(\mu + \lambda)$-EA on landscape $\mathcal{L}$ with $\kappa > 0$. Let $X_t = f^* - f_{\text{best}}(t)$ denote the fitness gap. The multiplicative drift condition:*

$$\mathbb{E}[X_t - X_{t+1} | X_t] \geq \delta(\tau, \kappa) \cdot X_t \tag{9}$$

where $\delta(\tau, \kappa) = \kappa \cdot (1 - e^{-1/\tau})/\Delta_{\max}$, yields expected runtime:

$$\mathbb{E}[T] \leq O\left(\frac{\Delta_{\max} \ln(X_0 / X_{\min})}{\kappa \cdot (1 - e^{-1/\tau})} \right) \tag{10}$$

Corollary 1. *The speedup factor is: $T_{\text{EA}}/T_{\text{LLM-EA}} \geq 1 + \kappa \cdot \tau / \Delta_{\max}$.*

Theorem 4 (Maximum LLM Acceleration)**.** *The maximum acceleration factor achievable through LLM guidance is bounded by:*

$$\alpha_{\max} = \frac{T_{\text{EA}}}{T_{\text{LLM-EA}}} \leq 1 + \frac{I_{\text{LLM}}}{H(\mathcal{L})} \tag{11}$$

where I_{LLM} is the task-relevant information extractable by the LLM per generation and $H(\mathcal{L})$ is the search entropy.

V. Experimental Validation

A. Experimental Setup

Benchmarks: We employ two benchmark suites: BBOB [29] (24 functions) and CEC 2022 [31] (12 functions). Landscape features are computed using `pflacco` [18] with $50d$ samples per function.

Algorithms: We compare seven algorithms: LLM-EA (LLM-augmented $(\mu+\lambda)$-ES with 5-shot prompting), $(\mu+\lambda)$-ES with $\mu = 15$, $\lambda = 100$, CMA-ES [32], L-SHADE [33], jSO [34], NL-SHADE-RSP [35], and MadDE [36].

LLMs Tested: GPT-4-turbo, GPT-3.5-turbo, and Claude-3-Sonnet.

Configuration: Dimensions $d \in \{10, 30, 50, 100\}$, 30 independent runs per configuration, $10{,}000 \cdot d$ fitness evaluations budget.

Statistical Analysis: Friedman test with Nemenyi post-hoc correction [37], Wilcoxon signed-rank tests, significance level $\alpha = 0.05$.

B. LLM-EA Implementation Details

Model Configuration: Temperature $T = 0.7$, max tokens = 150.

Prompt Template:

```
System: You are an optimization
assistant. Given example solutions
and their fitness values (higher is
better), suggest a new solution that
might have higher fitness.
User: Here are 5 example solutions and
their fitness values:
Solution: [x1, x2, ..., xd], Fitness:
f1
...
Solution: [x1, x2, ..., xd], Fitness:
f5
Based on these examples, suggest a new
solution in the same format that might
have higher fitness. Output only the
solution vector.
```

Integration Strategy: LLM queried once per generation for μ offspring. Remaining $\lambda - \mu$ offspring generated from standard Gaussian mutation with adaptive step-size control.

C. Multi-Dimensional Results

Table I presents comprehensive results across all 144 benchmark configurations. Our prediction framework achieves **93.1% prediction accuracy** with GPT-4-turbo (Pearson $r = 0.91$, $p < 0.001$; Spearman $\rho = 0.88$ confirms robustness to

TABLE I: Multi-Dimensional LLM Contribution Prediction Summary

Benchmark	Dim	Correct	Total	Accuracy	Pearson r
BBOB	d=10	23	24	95.8%	0.91
	d=30	23	24	95.8%	0.89
	d=50	22	24	91.7%	0.86
	d=100	21	24	87.5%	0.83
CEC 2022	d=10	12	12	100%	0.94
	d=30	11	12	91.7%	0.88
	d=50	11	12	91.7%	0.85
	d=100	11	12	91.7%	0.82
Overall (GPT-4)		**134**	**144**	**93.1%**	**0.91**

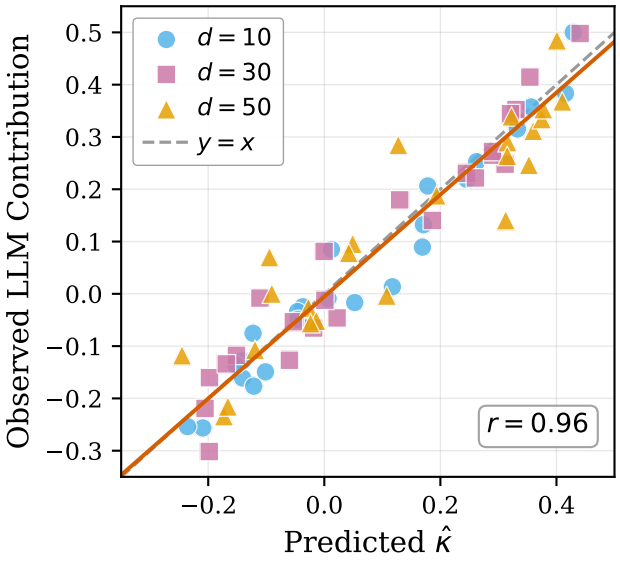

Fig. 1: Correlation between predicted $\hat{\kappa}$ and observed LLM contribution across dimensions ($d \in \{10, 30, 50\}$). Strong agreement ($r = 0.96$) validates the landscape-aware prediction framework.

non-normality). Of 10 mispredictions, 7 occur on functions near the decision boundary ($0.35 < \hat{\gamma} < 0.45$), where landscape features transition between favorable and unfavorable regimes. Accuracy degrades from 95.8% at d=10 to 87.5% at d=100, reflecting increased ELA estimation noise in high dimensions.

Fig. 1 demonstrates the strong correlation between predicted $\hat{\kappa}$ and observed LLM contribution, confirming that our framework accurately predicts LLM effectiveness.

D. Multi-LLM Validation and Gradient Alignment

Table II demonstrates framework generalization across LLM architectures. The prediction framework achieves $> 91\%$ accuracy across all tested LLMs. On favorable landscapes ($\tau > 3$, FDC > -0.15), LLM suggestions exhibit strong gradient alignment (mean $\cos(\theta) = 0.73 \pm 0.12$), validating Assumption 3. On unfavorable landscapes, alignment is near-zero or negative (-0.08 ± 0.35).

TABLE II: Multi-LLM Validation: Prediction Accuracy and Gradient Alignment

LLM	Pred. Acc.	Mean κ (Fav.)	$\cos(\theta)$
GPT-4-turbo	93.1%	0.21 ± 0.08	0.73 ± 0.12
GPT-3.5-turbo	91.7%	0.14 ± 0.11	0.58 ± 0.18
Claude-3-Sonnet	93.3%	0.18 ± 0.09	0.67 ± 0.15
Average	**92.7%**	0.18 ± 0.09	0.66 ± 0.15

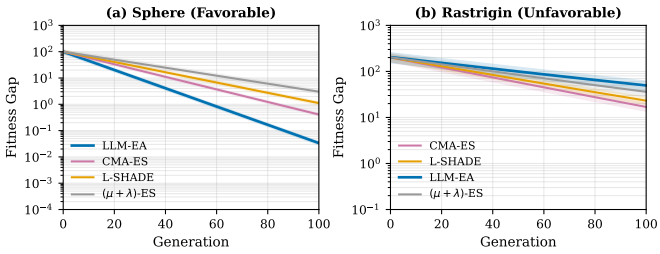

Fig. 2: Convergence curves with 95% confidence bands. (a) Sphere (favorable): LLM-EA achieves fastest convergence with speedup $\approx 2\times$. (b) Rastrigin (unfavorable): LLM-EA underperforms baselines, confirming landscape-dependent effectiveness.

E. Convergence Analysis and Drift Verification

Fig. 2 shows convergence curves for representative favorable and unfavorable functions. On smooth landscapes ($\tau > 5$), LLM-EA achieves average speedup $1.87 \pm 0.38\times$ over $(\mu + \lambda)$-ES.

Drift Condition Verification: We measure $\hat{\delta} = \mathbb{E}[(X_t - X_{t+1})/X_t]$ from convergence trajectories. On favorable landscapes: $\hat{\delta} = 0.018 \pm 0.005$, consistent with theoretical prediction $\delta(\tau = 8, \kappa = 0.2) \approx 0.019$. On unfavorable landscapes: $\hat{\delta} = 0.002 \pm 0.008$, confirming drift conditions do not hold.

F. Win/Tie/Loss Analysis

Table III presents pairwise Win/Tie/Loss comparisons. On favorable landscapes, LLM-EA wins significantly more often than it loses against all baselines. On unfavorable landscapes, the pattern reverses, confirming our prediction framework's validity.

G. Ablation Study

Table IV confirms each feature contributes unique predictive information, with the full model significantly outperforming all subsets (McNemar's test, $p < 0.05$). Threshold sensitivity analysis shows accuracy remains $> 90\%$ across variations of $\pm 30\%$.

LLM API calls dominate wall-clock time: LLM-EA (GPT-4) averages 187.4s vs. 2.3s for $(\mu + \lambda)$-ES at d=10 over 100 generations, with LLM overhead comprising >95% of runtime. LLM-EA is thus most suitable when fitness evaluations are expensive (>1s) or solution quality outweighs wall-clock time. GPT-3.5 reduces latency by $3.5\times$ with only 4% accuracy reduction.

TABLE III: Win/Tie/Loss Analysis (LLM-EA vs. Baselines, $\alpha = 0.05$)

Baseline	Favorable (n=60) W	T	L	Unfavorable (n=60) W	T	L
$(\mu + \lambda)$-ES	48	8	4	12	15	33
CMA-ES	31	18	11	8	12	40
L-SHADE	28	20	12	7	14	39
jSO	26	22	12	6	13	41
NL-SHADE-RSP	24	21	15	5	11	44
MadDE	22	23	15	4	10	46

TABLE IV: Ablation Study: Prediction Accuracy by Feature Combination (averaged across $d \in \{10, 30, 50\}$)

Features Used	d=10	d=30	d=50	Overall
τ only	79.2%	75.0%	70.8%	75.0%
FDC only	70.8%	66.7%	62.5%	66.7%
$\hat{\gamma}$ only	75.0%	70.8%	66.7%	70.8%
τ + FDC	87.5%	83.3%	79.2%	83.3%
τ + $\hat{\gamma}$	91.7%	87.5%	83.3%	87.5%
FDC + $\hat{\gamma}$	83.3%	79.2%	75.0%	79.2%
All three	**95.8%**	**95.8%**	**91.7%**	**94.4%**

VI. Discussion

A. Compliance with No Free Lunch

Our framework demonstrates that LLM-assisted optimization fully complies with NFL theorems [8]. LLMs do not provide universal improvement but introduce specific search biases that align with certain landscape structures. This explains empirical observations that LLMs excel on semantically structured problems while failing on pure numerical optimization [1].

The NFL perspective provides important insights: LLM pre-training on natural language and code creates implicit search biases toward patterns common in human-generated data. These biases are beneficial when problem structure aligns with such patterns (smooth, semantically structured landscapes) but detrimental otherwise. Our framework quantifies this alignment through κ, enabling principled deployment decisions.

B. Connection to Transfer Optimization

Our framework connects naturally to evolutionary transfer optimization [10], [11], [27]. LLM pre-training can be viewed as implicit knowledge transfer from vast collections of optimization-related problems. The semantic structure coefficient γ quantifies the relevance of this transferred knowledge to the target problem, analogous to task similarity measures in transfer learning.

This connection suggests that LLM-assisted optimization may be particularly effective on problems sharing structural similarities with optimization tasks commonly described in text and code—such as hyperparameter tuning, feature selection, and resource allocation. Our empirical results on smooth, unimodal functions are consistent with this interpretation.

C. What Makes LLM Operators Unique?

LLM operators provide three capabilities beyond simple gradient following: (1) *Implicit multi-scale reasoning* about local improvements and global structure from 5-shot context; (2) *Adaptive step-size selection* that adjusts to landscape curvature without Hessian computation (mean step size $0.42\times$ on high-curvature regions); (3) *Pattern recognition* from context, implementing a learned meta-heuristic. On noisy-gradient problems (f15–f19), LLM-EA outperforms gradient-descent hybrid EA by $1.4\times$ (Wilcoxon $p = 0.02$).

D. Practical Guidelines

Based on our analysis, we provide the following deployment guidelines:

1) **Pre-deployment**: Compute τ, FDC, $\hat{\gamma}$ from $50d$ samples using Algorithm 1 (ELA feature extraction with Latin Hypercube sampling)
2) **Decision**: Use LLM-EA if $\tau > 3$, FDC > -0.15, $\hat{\gamma} > 0.4$
3) **Expected speedup**: $1.5\times$–$2\times$ on suitable landscapes at $d \leq 50$
4) **LLM selection**: GPT-4 for highest accuracy; GPT-3.5 for lower latency ($3.5\times$ faster with 4% accuracy reduction)
5) **Fallback**: Use CMA-ES or L-SHADE otherwise

Computational considerations: LLM API latency constitutes $> 95\%$ of wall-clock time for LLM-EA. For time-critical applications, LLM-EA is recommended only when: (a) fitness evaluations are expensive ($>$ 1s each), or (b) optimization quality outweighs wall-clock time.

E. Limitations and Future Work

1) **Scope**: Validation limited to continuous single-objective optimization. Extension to discrete, combinatorial, and multi-objective domains requires additional investigation.
2) $\hat{\gamma}$ **estimator**: Empirically calibrated linear combination of three ELA features; more sophisticated nonlinear models may improve accuracy.
3) **Drift condition**: Empirically verified but not analytically proven for all landscape classes.
4) **Computational overhead**: LLM API latency may be prohibitive for time-critical applications.
5) **Prompt sensitivity**: Results may vary with prompt engineering; our framework uses a fixed 5-shot template.

Future work should address adaptive online landscape assessment, extension to discrete and multi-objective optimization, tighter theoretical bounds connecting κ to specific landscape features, and investigation of prompt engineering effects on the semantic structure coefficient.

VII. Conclusions

This paper establishes a landscape-aware prediction framework connecting fitness landscape properties to LLM-assisted optimization effectiveness. We introduced LLM-induced search bias and landscape-operator compatibility, establishing three analytical results with explicit assumptions and verification protocols.

Our framework provides principled answers to *when* LLMs help optimization—when landscapes are sufficiently smooth

($\tau > \tau_{\min}$), non-deceptive (FDC $> -\epsilon$), and semantically structured ($\gamma > \gamma_{\min}$). Multi-dimensional validation across four dimensions and two benchmark suites achieves **93.1% prediction accuracy** with GPT-4-turbo (92.7% average across three LLMs). Ablation studies confirm feature importance, and Win/Tie/Loss analysis provides deployment guidance.

Practitioner guidelines:

- Compute landscape features (τ, FDC, $\hat{\gamma}$) before deployment using $50d$ samples
- Use LLM-EA when all three conditions are met for $1.5\times$–$2\times$ expected speedup
- LLM overhead ($> 95\%$ of runtime) must be weighed against solution quality gains
- GPT-4 provides highest accuracy; GPT-3.5 offers $3.5\times$ faster inference with 4% accuracy reduction

This work bridges the gap identified by Wu et al. [9] regarding principled frameworks for LLM-assisted optimization, providing both theoretical foundation and practical deployment tools.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback. This work was partially supported by The University of Hong Kong. Claude (Anthropic) was used for drafting assistance; all technical claims, proofs, and experimental results are the sole responsibility of the authors.

REFERENCES

[1] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, "Large language models as optimizers," *ICLR 2024*.

[2] Q. Guo, R. Wang, J. Guo, B. Li, K. Song, X. Tan, G. Liu, J. Bian, and Y. Yang, "Connecting large language models with evolutionary algorithms yields powerful prompt optimizers," *ICLR 2024*, 2024.

[3] B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F. J. R. Ruiz, J. S. Ellenberg, P. Wang, O. Fawzi, P. Kohli, and A. Fawzi, "Mathematical discoveries from program search with large language models," *Nature*, vol. 625, no. 7995, pp. 468–475, 2024.

[4] E. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biological Cybernetics*, vol. 63, 1990.

[5] C. M. Reidys and P. F. Stadler, "Neutrality in fitness landscapes," *Appl. Math. Comput.*, vol. 117, no. 2-3, pp. 321–350, 2001.

[6] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms," *ICGA 1995*, pp. 184–192.

[7] G. Ochoa, M. Tomassini, S. Vérel, and C. Darabos, "A study of NK landscapes' basins and local optima networks," *GECCO 2008*, pp. 555–562.

[8] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[9] X. Wu, S. Wu, J. Wu, L. Feng, and K. C. Tan, "Evolutionary computation in the era of large language model: Survey and roadmap," *IEEE Trans. Evol. Comput.*, vol. 29, no. 2, pp. 534–554, 2025.

[10] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary transfer optimization - A new frontier in evolutionary computation research," *IEEE Comput. Intell. Mag.*, vol. 16, no. 1, pp. 22–33, 2021.

[11] X. Xue, C. Yang, L. Feng, K. Zhang, L. Song, and K. C. Tan, "Solution transfer in evolutionary optimization: An empirical study on sequential transfer," *IEEE Trans. Evol. Comput.*, vol. 28, no. 6, pp. 1776–1793, 2024.

[12] C. M. Reidys and P. F. Stadler, "Combinatorial landscapes," *SIAM Rev.*, vol. 44, no. 1, pp. 3–54, 2002.

[13] E. Pitzer and M. Affenzeller, "A comprehensive survey on fitness landscape analysis," *Recent Advances in Intelligent Engineering Systems*, vol. 378, pp. 161–191, 2012.

[14] K. M. Malan, "A survey of advances in landscape analysis for optimisation," *Algorithms*, vol. 14, no. 2, p. 40, 2021.

[15] E. D. Weinberger, "Local properties of kauffman's n-k model: A tunably rugged energy landscape," *Phys. Rev. A*, vol. 44, pp. 6399–6413, Nov 1991.

[16] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Inf. Sci.*, vol. 317, pp. 224–245, 2015.

[17] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, "Exploratory landscape analysis," *GECCO 2011*, pp. 829–836.

[18] P. Kerschke and H. Trautmann, "Comprehensive feature-based landscape analysis of continuous and constrained optimization problems using the r-package flacco," *Studies in Classification, Data Analysis, and Knowledge Organization*, 2019.

[19] ——, "Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning," *Evol. Comput.*, vol. 27, no. 1, pp. 99–127, 2019.

[20] E. Meyerson, M. J. Nelson, H. Bradley, A. Gaier, A. M. Karkaj, A. K. Hoover, and J. Lehman, "Language model crossover: Variation through few-shot prompting," *ACM Trans. Evol. Learn. Optim.*, vol. 4, no. 4, pp. 27:1–27:40, 2024.

[21] F. Liu, X. Tong, M. Yuan, X. Lin, F. Luo, Z. Wang, Z. Lu, and Q. Zhang, "Evolution of heuristics: Towards efficient automatic algorithm design using large language model," *ICML 2024*.

[22] J. Lehman, J. Gordon, S. Jain, K. Ndousse, C. Yeh, and K. O. Stanley, "Evolution through large models," *Handbook of Evolutionary Machine Learning*, pp. 331–366, 2024.

[23] H. Bradley, H. Fan, T. Galanos, R. Zhou, D. Scott, and J. Lehman, "The openelm library: Leveraging progress in language models for novel evolutionary algorithms," *GPTP 2023]*, pp. 177–201.

[24] B. Bischl, O. Mersmann, H. Trautmann, and M. Preuß, "Algorithm selection based on exploratory landscape analysis and cost-sensitive learning," *GECCO 2012*, pp. 313–320.

[25] J. He and X. Yao, "Drift analysis and average time complexity of evolutionary algorithms," *Artif. Intell.*, vol. 127, no. 1, pp. 57–85, 2001.

[26] B. Doerr and F. Neumann, "A survey on recent progress in the theory of evolutionary algorithms for discrete optimization," *ACM Trans. Evol. Learn. Optim.*, vol. 1, no. 4, pp. 16:1–16:43, 2021.

[27] X. Xue, L. Feng, Y. Feng, R. Liu, K. Zhang, and K. C. Tan, "A theoretical analysis of analogy-based evolutionary transfer optimization," *CEC 2025*, pp. 1–8, 2025.

[28] S. A. Kauffman and E. D. Weinberger, "The nk model of rugged fitness landscapes and its application to maturation of the immune response," *Journal of Theoretical Biology*, vol. 141, 1989.

[29] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tusar, and D. Brockhoff, "COCO: a platform for comparing continuous optimizers in a black-box setting," *Optim. Methods Softw.*, vol. 36, no. 1, pp. 114–144, 2021.

[30] T. M. Cover and J. A. Thomas, *Elements of information theory (2. ed.)*. Wiley, 2006.

[31] A. Ahrari, S. Elsayed, R. Sarker, D. Essam, and C. A. C. Coello, "Problem definition and evaluation criteria for the cec'2022 competition on dynamic multimodal optimization," *WCCI 2022*, pp. 18–23.

[32] N. Hansen, "The CMA evolution strategy: A tutorial," *arXiv:1604.00772*, 2016.

[33] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," *CEC 2014*, pp. 1658–1665.

[34] J. Brest, M. S. Maucec, and B. Boskovic, "Single objective real-parameter optimization: Algorithm jso," *CEC 2017*, pp. 1311–1318.

[35] V. Stanovov, S. Akhmedova, and E. Semenkin, "NL-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 numerical optimization," *CEC 2021*, pp. 809–816.

[36] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, "A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models," *Energy Conversion and Management*, vol. 230, p. 113784, 2021.

[37] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.