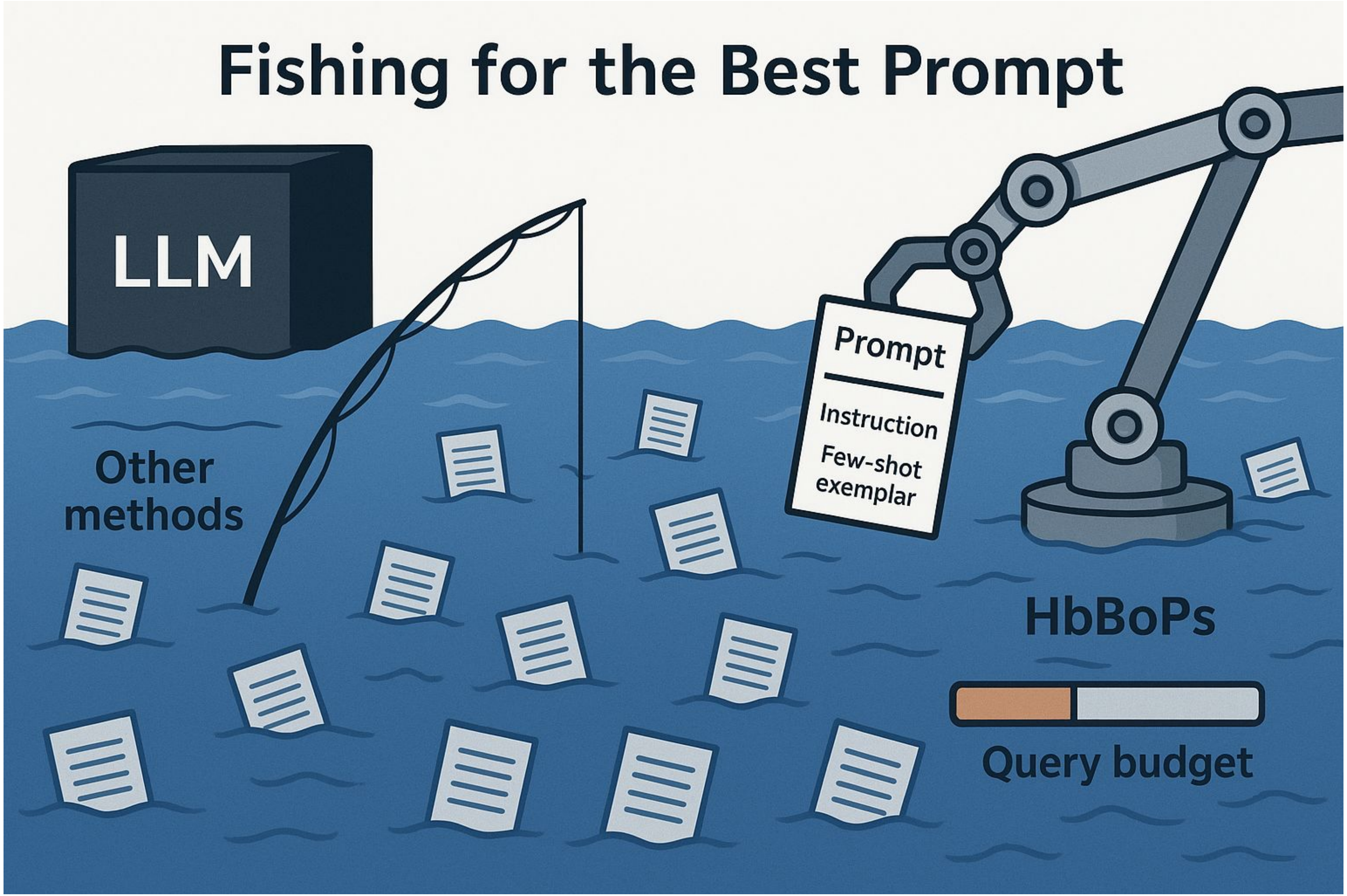
Hyperband-based Bayesian Optimization for Black-box Prompt Selection

Lennart Schneider - Martin Wistuba - Aaron Klein - Jacek Golebiowski - Giovanni Zappella - Felice Antonio Merra

Work done during an internship at AWS, Berlin, Germany

ICML 2025


Fishing for the Best Prompt
LLM
Other methods
Prompt
Instruction
Few-shot exemplar
HbBoPs
Query budget

Presentation Overview

1. Problem statement
2. Taxonomy and related work
3. Overview of HBBoPs methodology
4. Experimental setup and benchmarks
5. Results and analysis
6. Conclusions and future directions

Problem statement

Problem statement

- Assume prompts are composed of **instructions** and **few-shot exemplars**.
- We want to identify the prompt that **performs best in expectation** on a **downstream task**.
- Black-box optimization proxy: Evaluate prompt on a **validation set**.

$$\arg\min_{p \in \mathcal{P}} \mathbb{E}_{(x,y) \sim \mathbb{P}_{xy}} \left[l(y, h_p(x)) \right]$$

$$f(p) := \frac{1}{n_{\text{valid}}} \sum_{i=1}^{n_{\text{valid}}} l(y_i, h_p(x_i))$$

Taxonomy and related work

Taxonomy

Black-box:

- Only access to model outputs via API.
- Requires query-efficient, derivative-free methods.

White-box:

- Full access to the internals of the LLM, including gradients.
- Enables gradient-based prompt optimization or selection.

Static:

- A single prompt is chosen offline to generalize across all test instances.
- Prioritizes robustness and average-case performance.

Dynamic:

- Prompts are selected or adapted per test instance, often online.
- Allows for instance-specific reasoning and improved accuracy.

Selection:

- Choose the best-performing prompt from a (predefined) finite set.
- Emphasis is on efficient evaluation and ranking, not generation.

Optimization:

- Generating or refining new prompts.
- Techniques include gradient-based updates (in white-box) or evolutionary/search methods (in black-box).

Static black-box prompt selection: Related work

MIPROv2 (Opsahl-Ong et al., 2024)

- Combines instructions and few-shot exemplars from a finite prompt pool.
- Uses Tree-structured Parzen Estimator (TPE) with categorical indices.
- Limitations:
  - Lacks semantic modeling of prompts.
  - Evaluation not query-efficient; relies on full/random validation sets.

EASE (Wu et al., 2024)

- Uses NeuralUCB with embeddings of prompt text blocks.
- (Optional) optimal transport heuristic to reduce exemplar space.
- Limitations:
  - Does not make use of separate building blocks of prompts.
  - Evaluation not query-efficient; relies on full/random validation sets.

TRIPLE (Shi et al., 2024)

- Uses Successive Halving (SH) and Generalized Successive Elimination (GSE).
- Employs embeddings to model expected performance (for GSE)
- Limitations:
  - Sensitive to initial budgets.
  - Does not make use of separate building blocks of prompts.
  - Evaluates all prompts initially, limiting sample-efficiency.

→ Lack of a method that is both sample-efficient and query-efficient

Overview of HbBoPs methodology

Idea

Sample-efficiency via BO proposal:

- Prompts are natural language, yet composed of building blocks.
- How can we learn a surrogate model mapping prompts to downstream performance?

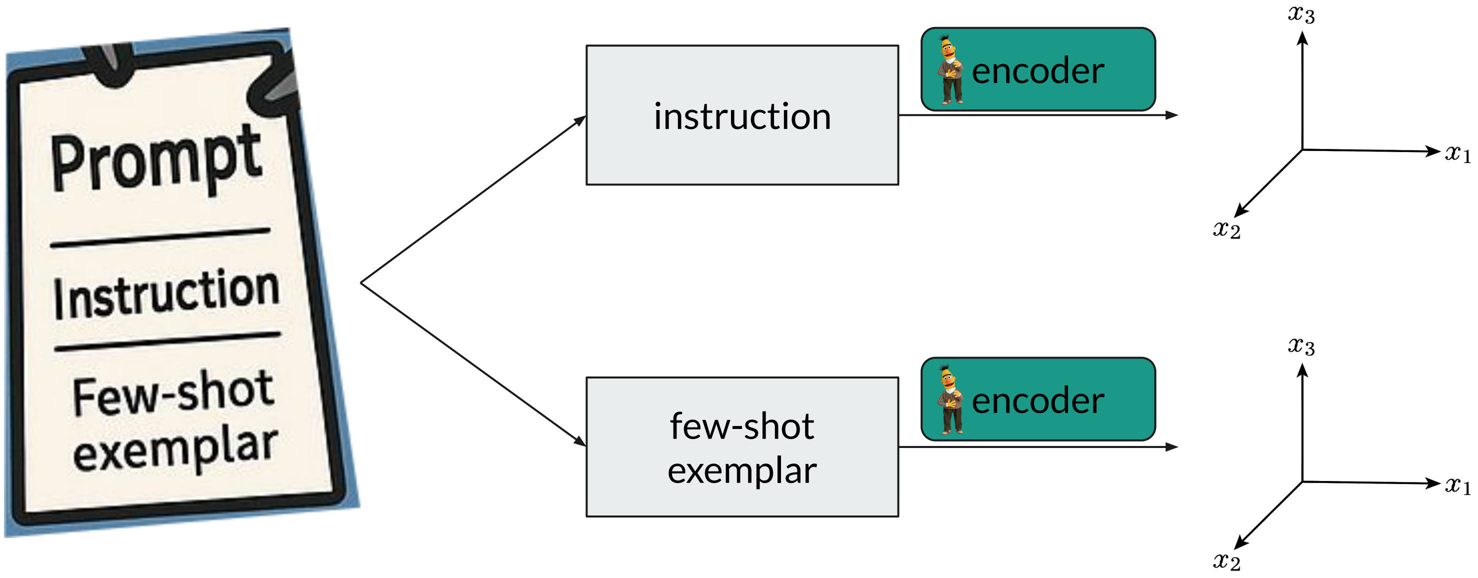
→ structural-aware deep kernel Gaussian Process

Query-efficiency via Hyperband (Li et al. 2018):

- Evaluating prompts on a validation set results in a natural fidelity: the number of validation samples.
- In contrast to HPO or NAS, the fidelity, however, only affects the noise of the objective without impacting trend.

→ adapt Hyperband to prompt selection

Structural-aware deep kernel Gaussian Process



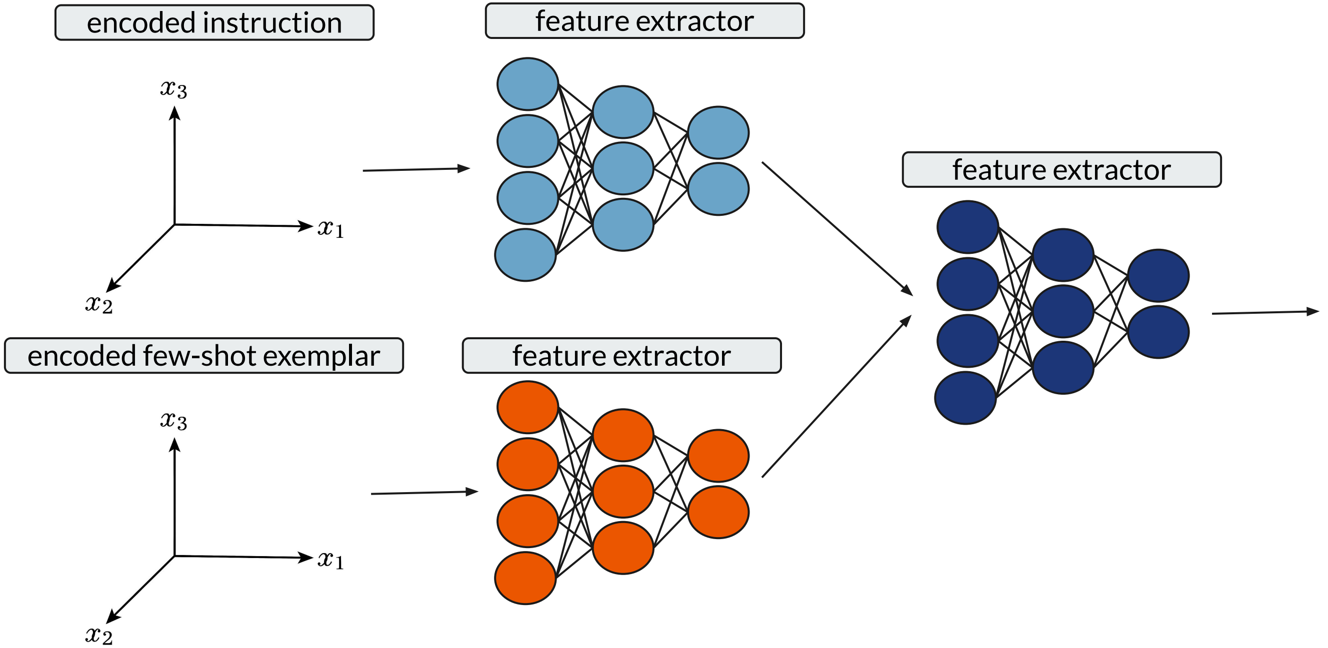
$\phi_{\mathrm{enc}(\cdot)}$:

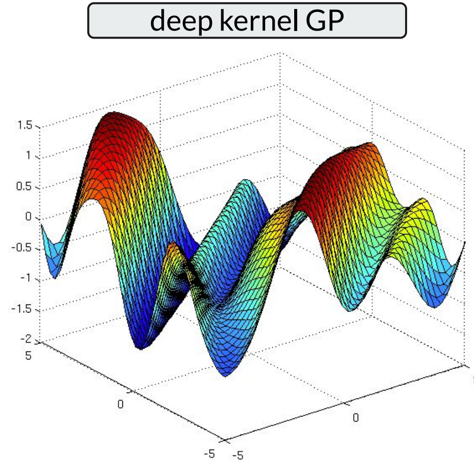
$\mathtt{Lin(d,\ 64)} \rightarrow \mathtt{ReLU()} \rightarrow \mathtt{Lin(64,\ 32)} \rightarrow \mathtt{ReLU()}$

$\phi_{(\phi_{\mathrm{enc}(i)}, \phi_{\mathrm{enc}(e)})}$:

$\mathtt{Lin(32} \cdot \mathtt{2,\ 32)} \rightarrow \mathtt{ReLU()} \rightarrow \mathtt{Lin(32,\ 10)}$

Structural-aware deep kernel Gaussian Process





Source: [The Kernel Cookbook by David Duvenaud]

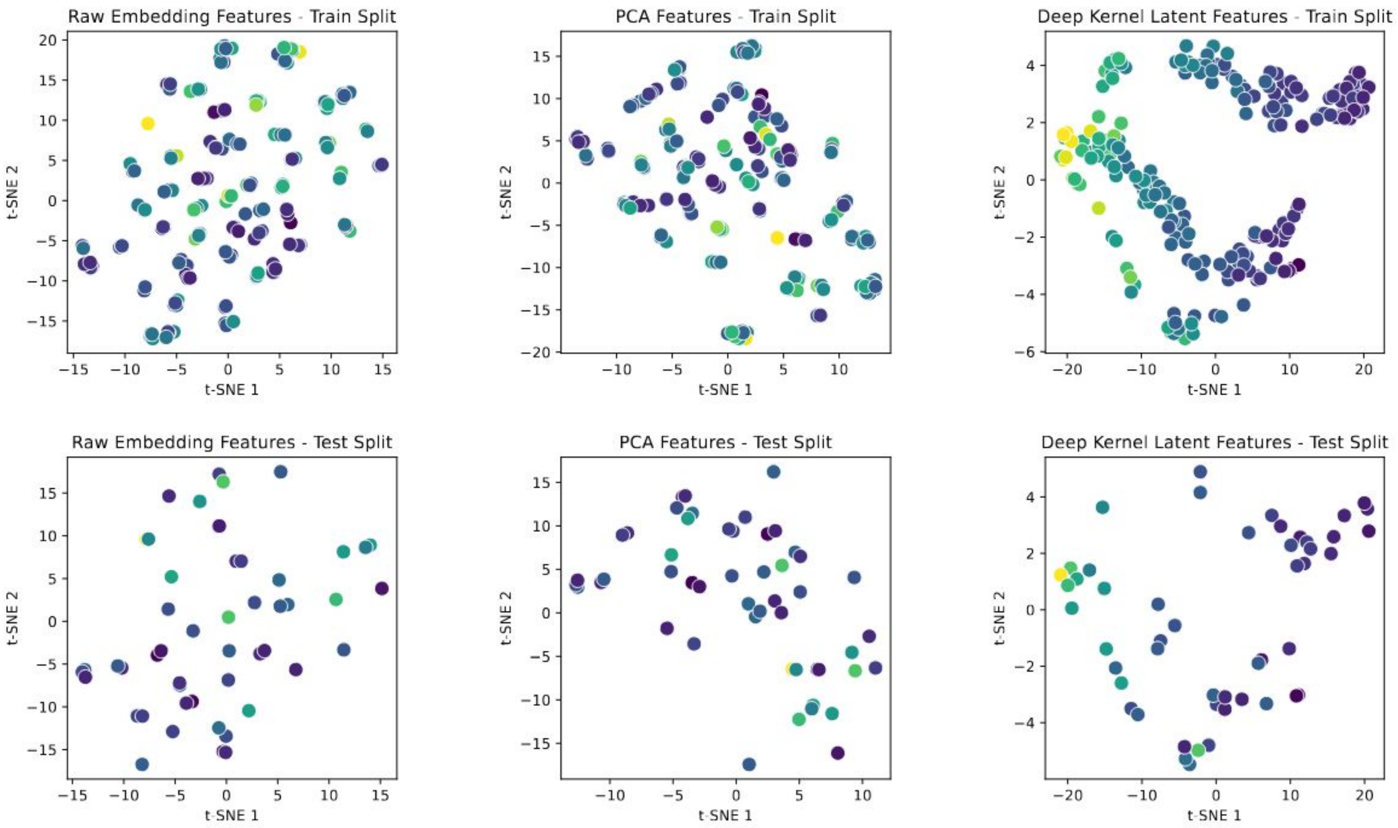


Figure 3. Visualization of the 768 dimensional BERT [CLS] token embeddings of prompts via a two component t-SNE. Left: Raw, unprocessed features. Middle: Features of a 10 component PCA solution. Right: Latent features (10 dimensional) from the feature extractor of our structural-aware DK-GP. Top row: Train split. Bottom row: Test split. Color indicates the performance of prompts for LLAMA3 8B Instruct on *GSM8K*.

Hyperband for prompt selection

Bracket (s)	Stage (i)	#Instances (b)	#Prompts (n)
3	0	10	8
3	1	20	4
3	2	40	2
3	3	80	1
2	0	20	6
2	1	40	3
2	2	80	1
1	0	40	4
1	1	80	2
0	0	80	4

- How to determine the incumbent?
  → best performing prompt on highest fidelity
- Purely random instances for evaluation within stages of a bracket vs. “fixed” random instances?
  → fixed
- Superset structures vs. no superset structure of instances when moving from one stage to another within a bracket?
  → superset structure
- **Note**: if LLM evaluation is close to deterministic, they can be cached an re-used when moving from one stage to another stage within a bracket

HbBoPs

- Combine Hyperband for prompt selection with a BO proposal based on the structural-aware deep kernel GP in the spirit of BOHB (Falkner et al. 2018)
- Acquisition function based on EI:

$$\alpha_{\text{EI}}(p|\mathcal{D}_{t|b}) := \mathbb{E}[\max\{v_{\min,b} - f(\mathbf{z}_p), 0\}]$$

$$p_{t+1} = \arg\max_{p \in \mathcal{P}} \alpha_{\text{EI}}(p|\mathcal{D}_{t|b}),$$

Algorithm 1 HbBoPs

```
input n_valid, b_min (lower limit to #validation instances), η (halving parameter)
  r = n_valid / b_min
  s_max = ⌊log_η(r)⌋
  B = (s_max + 1) n_valid
  for s ∈ {s_max, s_max − 1, ..., 0} do
    n = ⌈ (B / n_valid) (η^s / (s+1)) ⌉
    b = n_valid η^(−s)
    P = {}, V = {}
    for j ∈ {0, ..., n − 1} do
      p = get_prompt()
      v = get_validation_error(p, b)
      P ← P ∪ {p}, V ← V ∪ {v}
    end for
    P = top_k(P, V, ⌊n/η⌋)
    for i ∈ {1, ..., s} do
      n_i = ⌊n η^(−i)⌋
      b_i = b η^i
      V = {get_validation_error(p, b_i) : p ∈ P}
      P = top_k(P, V, ⌊n_i/η⌋)
    end for
  end for
output Prompt with the lowest validation error evaluated
  on the whole validation set
```

Experimental setup and benchmarks

Benchmark tasks

- **AI2's Reasoning Challenge (ARC)** - Multiple-choice question answering (Clark et al., 2018)
- **GSM8K** - Multi-step math problems (Cobbe et al., 2021)
- 8 tasks from **BIG-bench / Instruction Induction** (BBII): *antonyms, larger animal, negation, second word letter, sentiment, object counting, orthography starts with, word unscrambling* (Srivastava et al., 2023; Honovich et al., 2023)

Prompt pool

Instructions (5 per task):

- APE (forward mode; Zhou et al. 2023) using Claude 3 Sonnet based on 10 I/O examples.

Few-shot exemplars (50 per task):

- 25 sets of 5 I/O examples sampled from the tas’s training set.
- Each set permuted twice to test ordering sensitivity.

→ Final prompt space via Cartesian product

LLMS

- Claude 3 Haiku
- LLAMA3 8B Instruct
- Mistral 7B Instruct

Evaluation protocol

- Evaluation budget: 25 full-fidelity evaluations per method per (task, LLM) pair.
- Cost metric: Number of LLM calls used (model-agnostic and interpretable).
- Repetitions: Each experiment is repeated 30 times for statistical reliability.
- Prompt evaluation metric: Based on exact match scoring function.

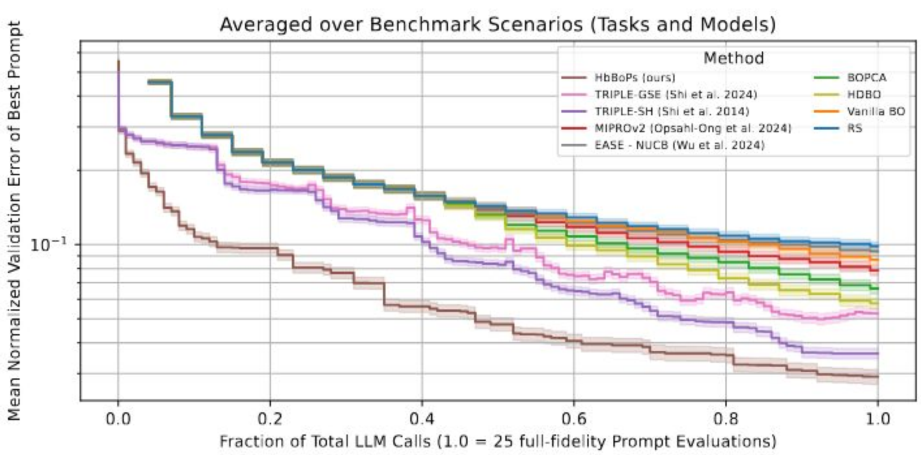
Baselines and competitors

Table 1. Overview of baselines, competitors and our HbBoPs in the static black-box prompt selection setting.

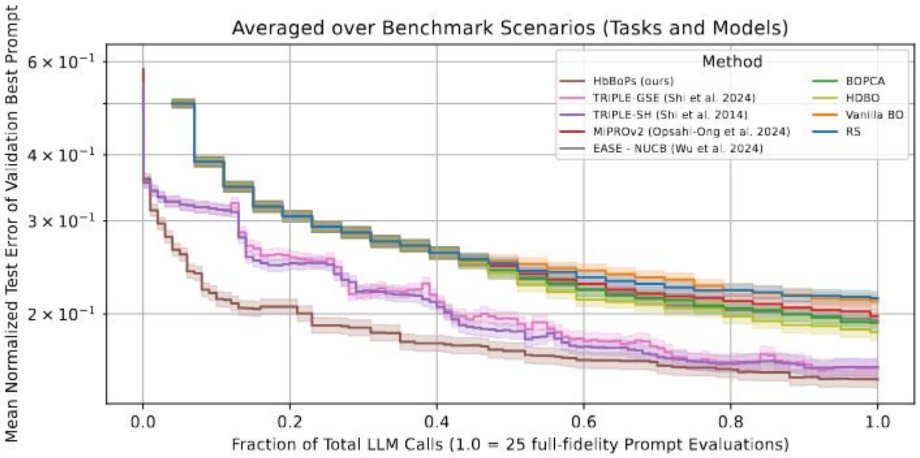
Method	Fidelity Level	Efficiency		Surrogate Model	Bandit Algorithm	Prompt Representation
		sample	query			
RS	Full	-	-	-	-	p
Vanilla BO	Full	✓	-	vanilla GP	-	$enc(p)$
HDBO	Full	✓	-	GP (Hvarfner et al., 2024)	-	$enc(p)$
BOPCA	Full	✓	-	vanilla GP	-	PCA$(enc(p))$ (Zhang et al., 2024)
EASE (Wu et al., 2024)	Full	✓	-	NN	NUCB	$enc(p)$
MIPROv2 (Opsahl-Ong et al., 2024)	Full	✓	-	TPE	-	ID_i ID_e
TRIPLE-SH (Shi et al., 2024)	Multi	-	✓	-	SH	p
TRIPLE-GSE (Shi et al., 2024)	Multi	-	✓	LM/GLM	GSE	$enc(p)$
HbBoPs (ours)	Multi	✓	✓	structural-aware DK-GP	HB	$enc(i), enc(e)$

Results and analysis

Main results



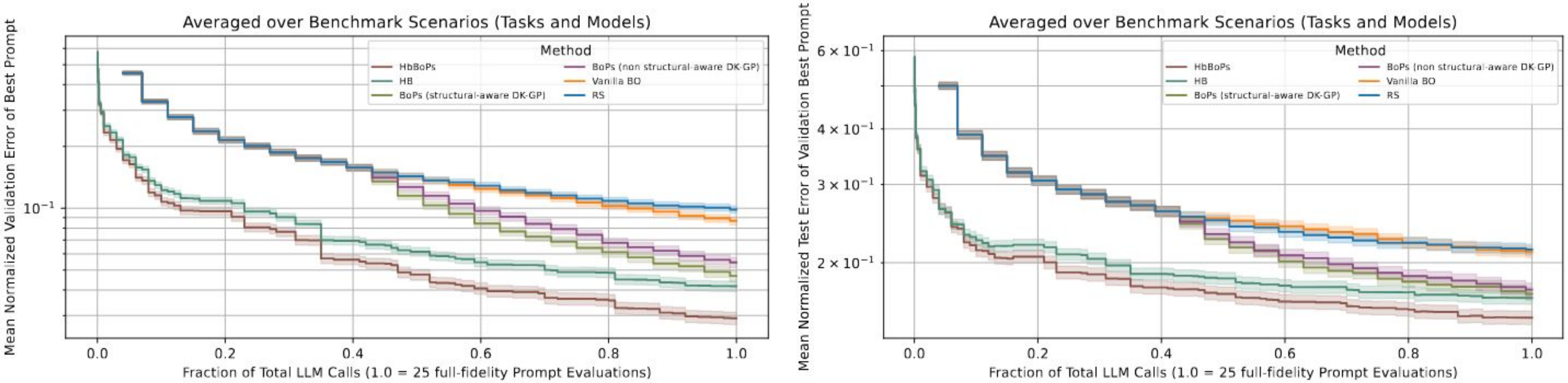
(a) Validation



(b) Test

Figure 1. Normalized error (log scale) of the best prompt per method, averaged over benchmarks. Lower is better. Ribbons represent SE.

Ablation: Components of HbBoPs



(a) Validation (b) Test

Figure 2. Normalized error (log scale) of the best prompt per HbBoPs ablation variant, RS, and vanilla BO, averaged over benchmarks. Lower is better. Ribbons represent SE.

Sensitivity Analysis: Encoder Model

Table 3. Normalized validation and test error of `HbBoPs` with different encoders at different fractions of total LLM calls averaged over all 30 benchmarks. SE in parentheses.

		Fraction of Total LLM Calls		
		0.25	0.50	1.00
BERT	Valid	0.081 (0.004)	0.048 (0.003)	0.029 (0.002)
	Test	0.190 (0.006)	0.170 (0.006)	0.150 (0.005)
MPNet	Valid	0.083 (0.004)	0.049 (0.003)	0.031 (0.002)
	Test	0.193 (0.006)	0.173 (0.006)	0.158 (0.006)
DistillRoBERTa	Valid	0.071 (0.003)	0.045 (0.002)	0.026 (0.002)
	Test	0.185 (0.006)	0.166 (0.006)	0.150 (0.005)

Conclusions and future directions

Conclusions

- HbBoPs enables efficient black-box prompt selection using structural-aware modeling and adaptive fidelity scheduling.
- Outperforms state-of-the-art methods (e.g., MIPROv2, EASE, TRIPLE) in performance and efficiency.
- Uses Deep Kernel GP to model downstream prompt performance (instructions + exemplars).
- Uses Hyperband to allocate evaluation resources cost-effectively.
- Robust across 10 tasks and 3 LLMs under tight evaluation budgets.
- Avoids full evaluation of all prompts, enhancing scalability.
- Offers a strong baseline for static black-box prompt selection.
- Prompt selection / optimization can be an interesting venue for AutoML methods.

Future directions

- Extend to richer prompt space (output guidance, formatting constraints, ...).
- Extend to multi-objective setting (number of few-shot examples in exemplar and prompt length).
- Integrate into end-to-end prompt optimization pipelines.
- Investigate robustness to noisy performance estimates in low-fidelity settings.

References

Opsahl-Ong, K., Ryan, M. J., Purtell, J., Broman, D., Potts, C., Zaharia, M., and Khattab, O. Optimizing instructions and demonstrations for multi-stage language model programs. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 9340–9366, 2024.

Wu, Z., Lin, X., Dai, Z., Hu, W., Shu, Y., Ng, S.-K., Jaillet P., and Low, B. K. H. Prompt optimization with EASE? Efficient ordering-aware automated selection of exemplars. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), Advances in Neural Information Processing Systems, volume 37, pp. 122706–122740, 2024.

Shi, C., Yang, K., Chen, Z., Li, J., Yang, J., and Shen, C. Efficient prompt optimization through the lens of best arm identification. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), Advances in Neural Information Processing Systems, volume 37, pp. 99646–99685, 2024.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18(185):1–52, 2018.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In Gretton, A. and Robert, C. C. (eds.), Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, volume 51, pp. 370–378, 2016.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, 2019.

Falkner, S., Klein, A., and Hutter, F. BOHB: Robust and efficient hyperparameter optimization at scale. In Dy, J. and Krause, A. (eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80, pp. 1437–1446, 2018.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? Try ARC, the AI2 Reasoning Challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Srivastava, A. et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. Transactions on Machine Learning Research, 2023.

Honovich, O., Shaham, U., Bowman, S. R., and Levy, O.Instruction induction: From few examples to natural language task descriptions. In 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023, pp. 1935–1952, 2023.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. Large language models are human-level prompt engineers. In The Eleventh International Conference on Learning Representations, 2023.

Appendix

Algorithm 1 `HbBoPs`

input n_{valid}, $b_{\min}$ (lower limit to #validation instances), η (halving parameter)

$r = n_{\text{valid}}/b_{\min}$
$s_{\max} = \lfloor \log_\eta(r) \rfloor$
$B = (s_{\max} + 1)n_{\text{valid}}$
for $s \in \{s_{\max}, s_{\max} - 1, \ldots, 0\}$ **do**
  $n = \left\lceil \frac{B}{n_{\text{valid}}} \frac{\eta^s}{(s+1)} \right\rceil$
  $b = n_{\text{valid}}\eta^{-s}$
  $P = \{\}, V = \{\}$
  for $j \in \{0, \ldots, n-1\}$ **do**
    $p = \text{get\_prompt}()$
    $v = \text{get\_validation\_error}(p, b)$
    $P \leftarrow P \cup \{p\}, V \leftarrow V \cup \{v\}$
  end for
  $P = \text{top\_k}(P, V, \lfloor n/\eta \rfloor)$
  for $i \in \{1, \ldots, s\}$ **do**
    $n_i = \lfloor n\eta^{-i} \rfloor$
    $b_i = b\eta^i$
    $V = \{\text{get\_validation\_error}(p, b_i) : p \in P\}$
    $P = \text{top\_k}(P, V, \lfloor n_i/\eta \rfloor)$
  end for
end for

output Prompt with the lowest validation error evaluated on the whole validation set

Table 5. Characteristics of tasks used in the experiments.

Task	**Setting**	n_{train}	n_{valid}	n_{test}
AI2 ARC	multiple choice question answering	1094	291	1144
GSM8K	grade school math questions	6154	1319	1319
antonyms	find antonym of word	2073	519	100
larger animal	select larger of two animals	2422	606	100
negation	negate a sentence	723	181	100
object counting	count number of objects	560	140	100
orthography starts with	output all words starting with a given letter	2400	600	100
second word letter	output the second letter of a word	2644	662	100
sentiment	sentiment analysis of movie rating	933	234	100
word unscrambling	build a word from scrambled letters	5627	1407	100

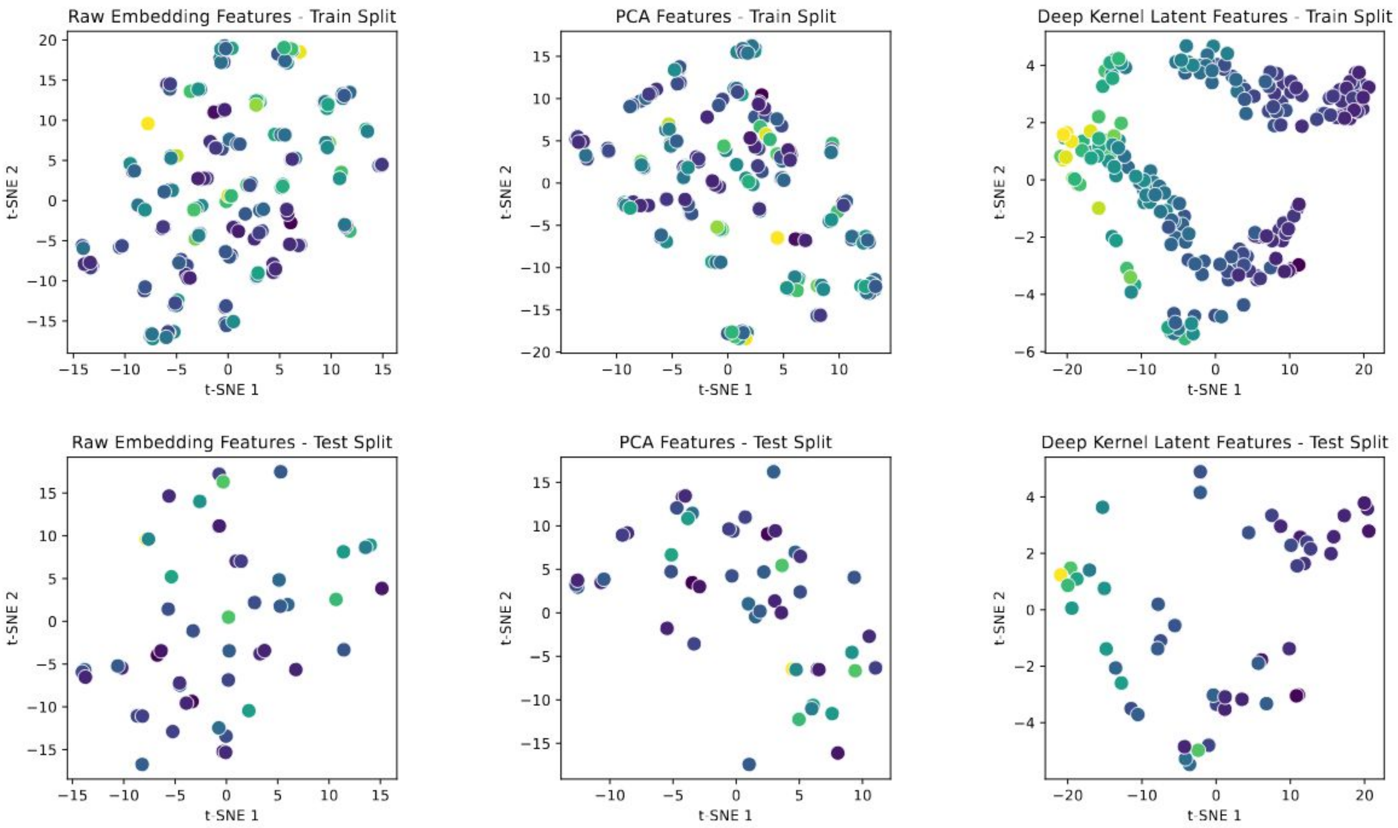
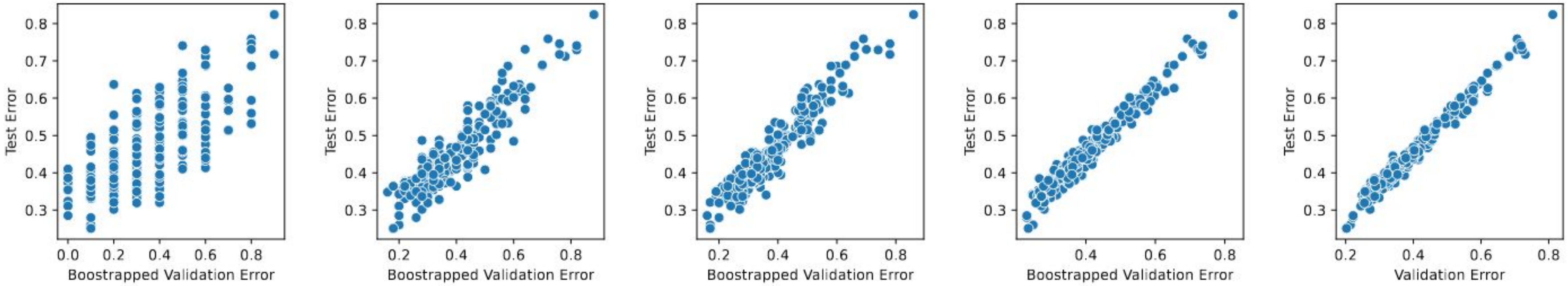


Figure 3. Visualization of the 768 dimensional BERT [CLS] token embeddings of prompts via a two component t-SNE. Left: Raw, unprocessed features. Middle: Features of a 10 component PCA solution. Right: Latent features (10 dimensional) from the feature extractor of our structural-aware DK-GP. Top row: Train split. Bottom row: Test split. Color indicates the performance of prompts for LLAMA3 8B Instruct on *GSM8K*.



(a) $k = 10$. (b) $k = 50$. (c) $k = 100$. (d) $k = 500$. (e) Full validation set.

Figure 4. Scatter plots of the validation and test errors of 250 prompts evaluated with LLAMA3 8B Instruct on *GSM8K* using differently sized ($k = 10, 50, 100, 500$) bootstrap samples of validation instances (a) to (d) or the full validation set (e).

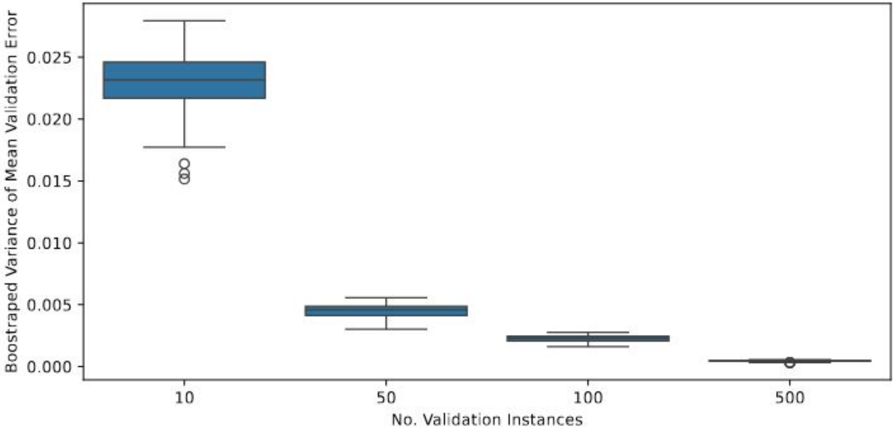
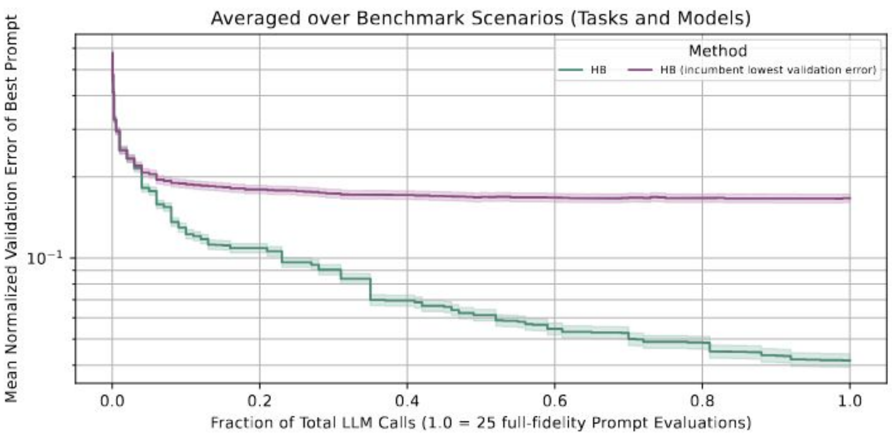


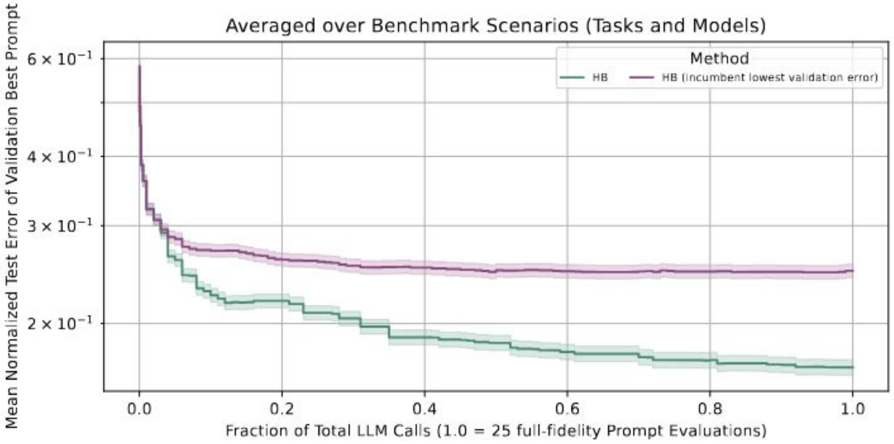
Figure 5. Box plots of the bootstrapped variance estimates of the mean validation error of 250 prompts evaluated with LLAMA3 8B Instruct on *GSM8K* varying the number of validation instances used to estimate the mean validation error.

Table 4. Exemplary HB schedule for black-box prompt selection assuming a minimum budget of $b_{\min} = 10$ validation instances, a maximum number of $n_{\text{valid}} = 80$ validation instances being available in total, and a halving parameter of $\eta = 2.0$.

Bracket (s)	**Stage** (i)	**#Instances** (b)	**#Prompts** (n)
3	0	10	8
3	1	20	4
3	2	40	2
3	3	80	1
2	0	20	6
2	1	40	3
2	2	80	1
1	0	40	4
1	1	80	2
0	0	80	4

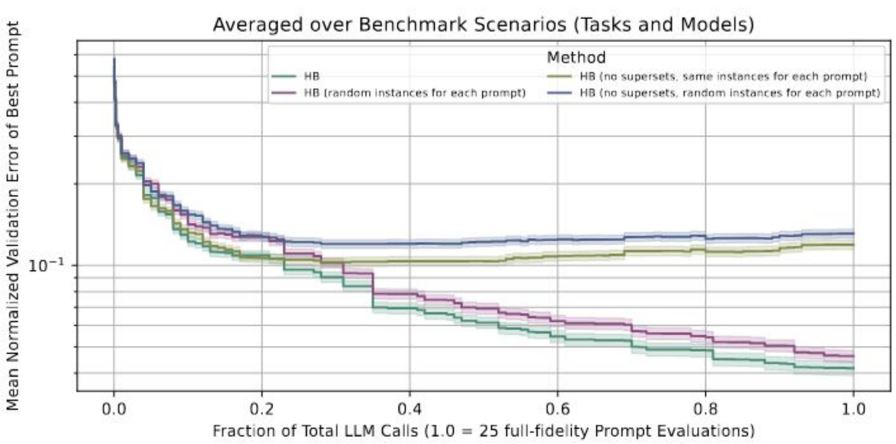


(a) Validation

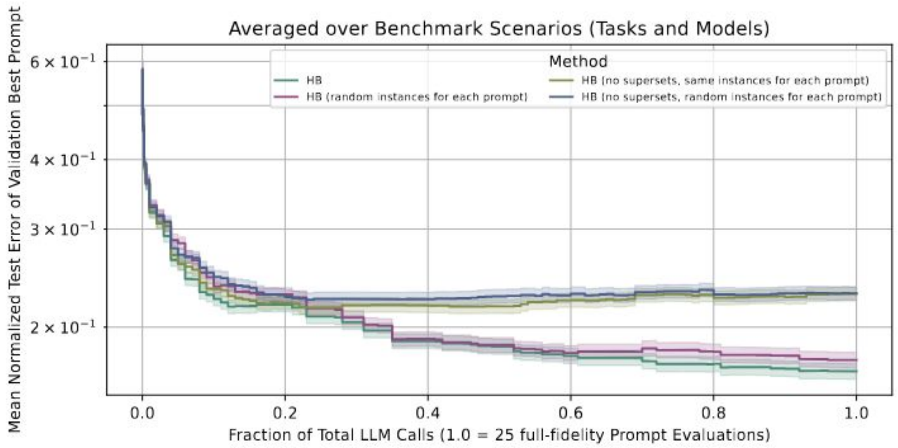


(b) Test

Figure 6. Normalized error (log scale) of the best prompt found by each HB incumbent selection mechanism, averaged over benchmarks. Lower is better. Ribbons represent SE.



(a) Validation



(b) Test

Figure 7. Normalized error (log scale) of the best prompt found by each HB validation instances sampling variant, averaged over benchmarks. Lower is better. Ribbons represent SE.

Table 4. Exemplary HB schedule for black-box prompt selection assuming a minimum budget of $b_{\text{min}} = 10$ validation instances, a maximum number of $n_{\text{valid}} = 80$ validation instances being available in total, and a halving parameter of $\eta = 2.0$.

Bracket (s)	**Stage** (i)	**#Instances** (b)	**#Prompts** (n)
3	0	10	8
3	1	20	4
3	2	40	2
3	3	80	1
2	0	20	6
2	1	40	3
2	2	80	1
1	0	40	4
1	1	80	2
0	0	80	4