BBox-Adapter

Lightweight Adapting for Black-Box Large Language Models

Haotian Sun[1]*, Yuchen Zhuang[1]*, Wei Wei[2], Chao Zhang[1], Bo Dai[1]

[1] Georgia Institute of Technology
[2]Accenture

Paper Code Website

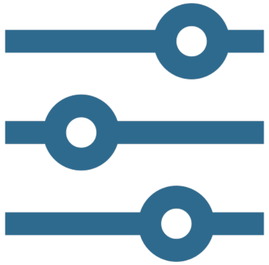
ICML International Conference On Machine Learning

Georgia Tech

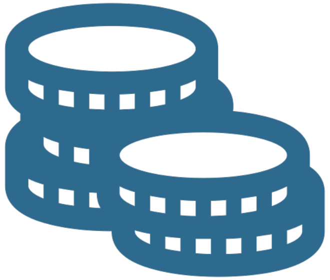
accenture

- Adapting Blackbox LLMs *only through* their APIs raises problems with **privacy, transparency, and cost**.

uploading training data via APIs…

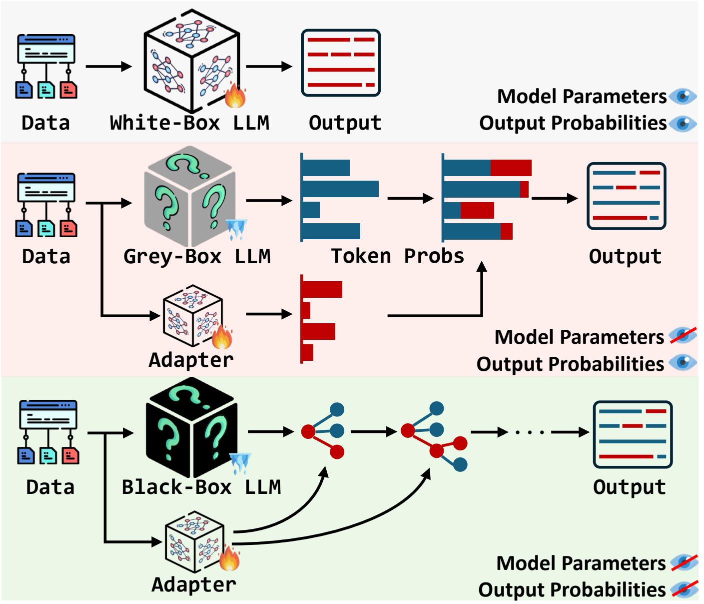


a restricted set of adjustable hyperparameters…

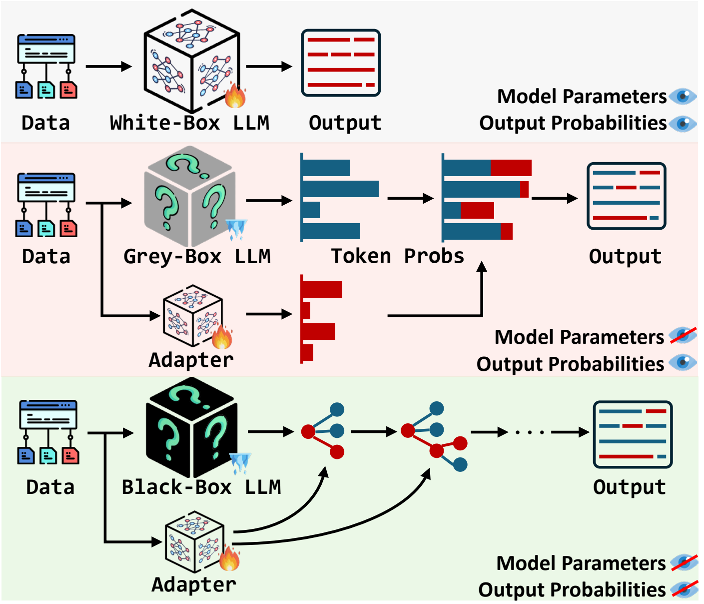


fine-tuning APIs much higher compared to inference…

- Adapting Blackbox LLMs *only through* their APIs raises problems with **privacy, transparency, and cost**.
- Existing methods fail to support this real black-box LLM setting, where neither model parameters nor output probabilities can be accessed in the most recent LLM APIs.

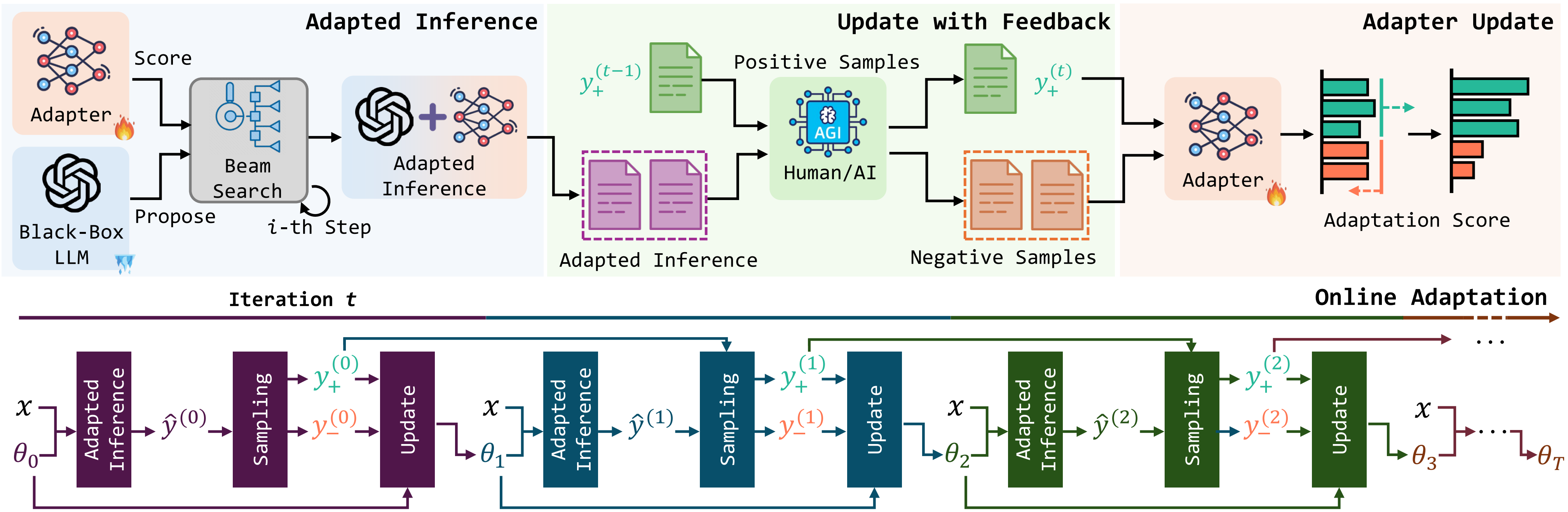


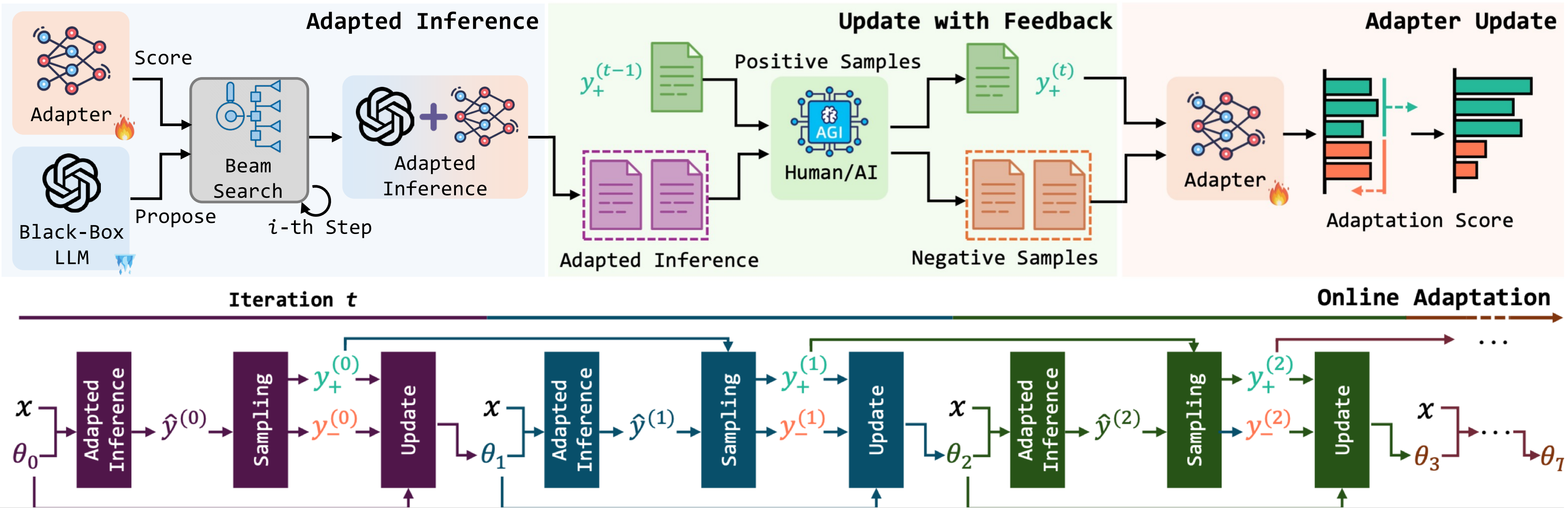
- Adapting Blackbox LLMs *only through* their APIs raises problems with **privacy, transparency, and cost**.
- Existing methods fail to support this real black-box LLM setting, where neither model parameters nor output probabilities can be accessed in the most recent LLM APIs.



Methods	w/o Model Parameters	w/o High-Dimensional Representation	w/o Token Probabilities	w/o Retrieval Corpus	w/ Smaller Adapter
White-Box LLM Fine-Tuning					
Fine-Tuning (Devlin et al., 2019)	✗	✗	✗	✓	✗
Instruction-Tuning (Wei et al., 2021)	✗	✗	✗	✓	✗
Continual Pre-Training (Gururangan et al., 2020)	✗	✗	✗	✓	✗
Adapter (Houlsby et al., 2019)	✗	✗	✗	✓	✓
Prefix-Tuning (Liu et al., 2022)	✗	✗	✗	✓	✓
LoRA (Hu et al., 2021)	✗	✗	✗	✓	✓
Grey-Box LLM Adaptation					
LMaaS (Sun et al., 2022)	✓	✗	✗	✓	✓
kNN-Adapter (Huang et al., 2023)	✓	✓	✗	✗	✓
CombLM (Ormazabal et al., 2023)	✓	✓	✗	✓	✓
IPA (Lu et al., 2023)	✓	✓	✗	✓	✓
Proxy-Tuning (Liu et al., 2024)	✓	✓	✗	✓	✓
Black-Box LLM Adaptation					
BBOX-ADAPTER (Ours)	✓	✓	✓	✓	✓

BBox-Adapter offers a novel lightweight adaption solution for customizing commercial black-box LLMs with *only* APIs.





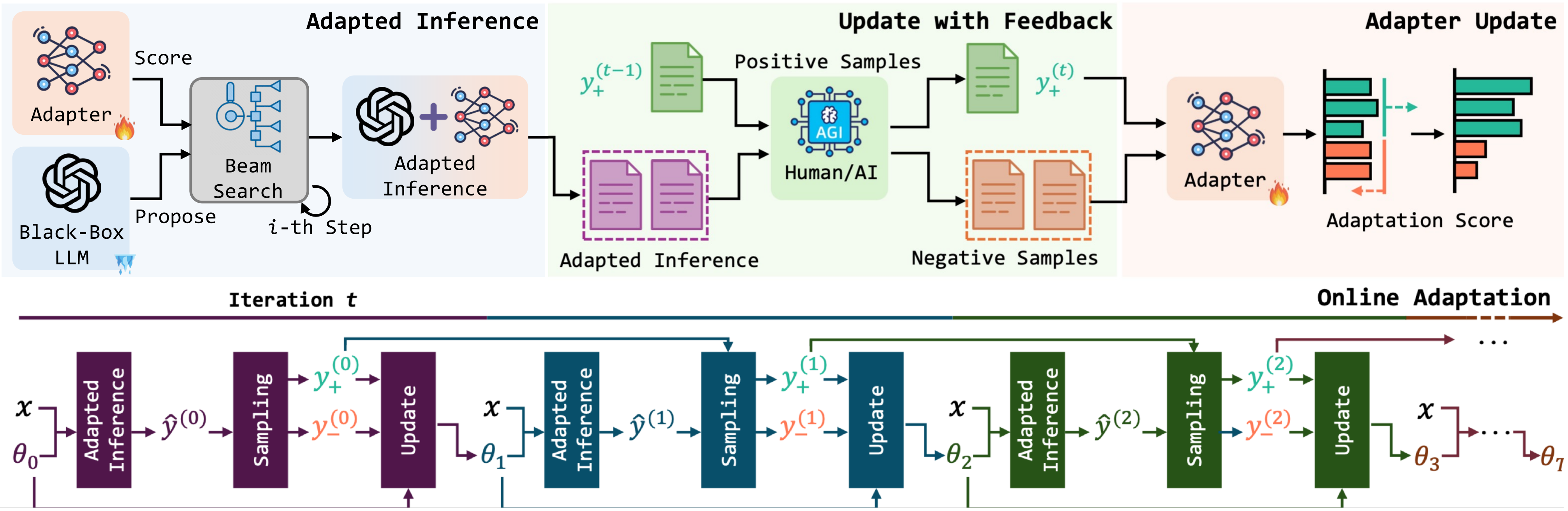
Adaptation as Energy-based Models

$$p_\theta(\mathbf{y}|\mathbf{x}) = p_{\text{LLM}}(\mathbf{y}|\mathbf{x}) \frac{\exp(g_\theta(\mathbf{x},\mathbf{y}))}{Z_\theta(\mathbf{x})}$$

$$Z_\theta(\mathbf{x}) = \int p_{\text{LLM}}(\mathbf{y}|\mathbf{x}) \exp(g_\theta(\mathbf{x},\mathbf{y})) d\mathbf{y}$$

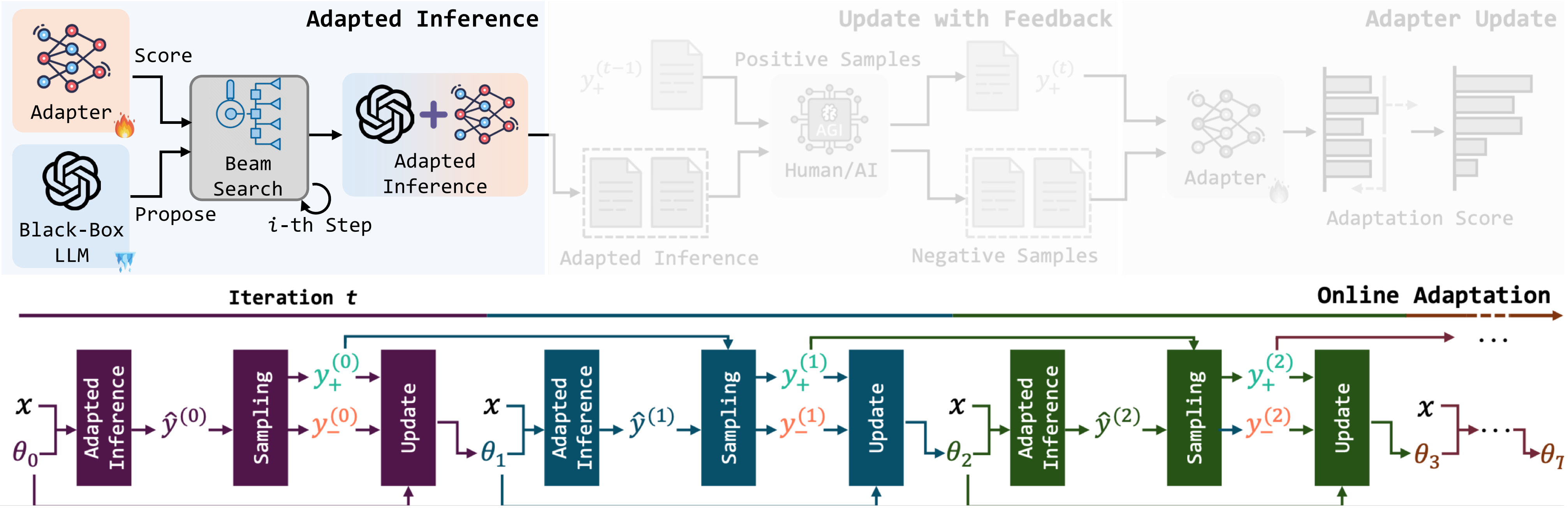
NCE loss

$$\nabla_\theta \ell(\theta) = \nabla_\theta \{ -\mathbb{E}_{\mathbf{y}_+ \sim p_{\text{data}}(\mathbf{y}|\mathbf{x})}[g_\theta(\mathbf{x},\mathbf{y}_+)] + \alpha \mathbb{E}[g_\theta(\mathbf{x},\mathbf{y}_+)^2] + \mathbb{E}_{\mathbf{y}_- \sim p_\theta(\mathbf{y}|\mathbf{x})}[g_\theta(\mathbf{x},\mathbf{y}_-)] + \alpha \mathbb{E}[g_\theta(\mathbf{x},\mathbf{y}_-)^2] \}.$$



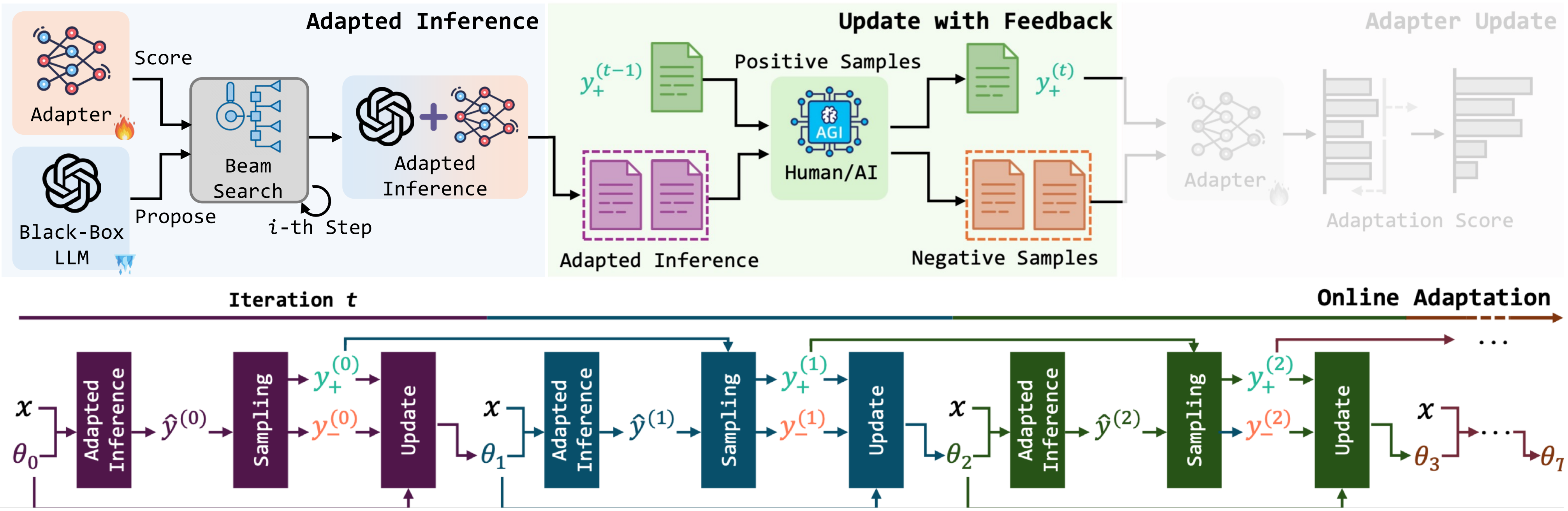
Online adaptation draws training samples from dynamic distributions to optimize its backend adapter.

1) *Sampling from Adapted Inference.*
2) *Updating Training Data with Feedback.*
3) *Update Adapter Parameters.*



Adapted Inference employs a black-box Language Model (LM) to generate proposals and an adapter for evaluation. The process simplifies complex tasks into sentence-level searches by selecting and evaluating candidate sequences.

$$
\begin{aligned}
p_\theta(\mathbf{y}|\mathbf{x}) &= p_\theta(\mathbf{s}^{1:L}|\mathbf{x}) = p_{\mathrm{LLM}}(\mathbf{s}^{1:L}|\mathbf{x}) \exp(g_\theta(\mathbf{s}^{1:L}, \mathbf{x})) \\
&= \exp(g_\theta(\mathbf{s}^{1:L}, \mathbf{x})) \prod_l p_{\mathrm{LLM}}(\mathbf{s}^l|\mathbf{x}, \mathbf{s}^{1:l-1}).
\end{aligned}
$$

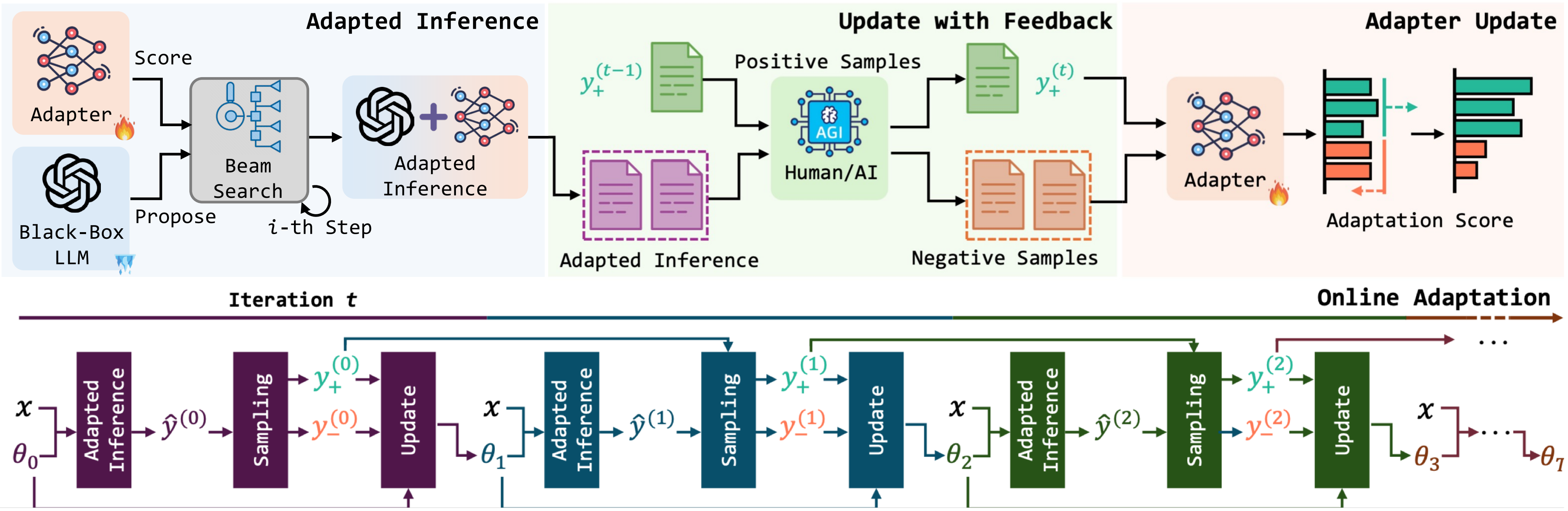


Updating Training Data with Feedback

Positive samples are selected based on ground-truth solutions or AI feedback from advanced LLMs like GPT-4, which simulate human judgment. We also employ outcome supervision to categorize further adapted inferences into correct outcomes (positive) and incorrect (negative) and update the respective sets accordingly.

$$\mathbf{y}_{i+}^{(t)} = \mathrm{SEL}(\mathbf{y}_{i+}^{(t-1)}, \{\hat{\mathbf{y}}_{i,m}\}_{m=1}^{M}).$$

$$\mathbf{y}_{i-}^{(t)} = \{\hat{\mathbf{y}}_{i,m} | \hat{\mathbf{y}}_{i,m} \neq \mathbf{y}_{i+}^{(t)}\}_{m=1}^{M}.$$



Adapter Update. With the updated positive and negative samples, we update the adapter parameters using the NCE loss, which minimizes energy for positive samples and increases it for negative ones.

$$
\begin{aligned}
\nabla_\theta \ell(\theta) = & \nabla_\theta \{-\mathbb{E}_{\mathbf{y}_+ \sim p_{\text{data}}(\mathbf{y}|\mathbf{x})}[g_\theta(\mathbf{x}, \mathbf{y}_+)] + \alpha \mathbb{E}[g_\theta(\mathbf{x}, \mathbf{y}_+)^2] \\
& + \mathbb{E}_{\mathbf{y}_- \sim p_\theta(\mathbf{y}|\mathbf{x})}[g_\theta(\mathbf{x}, \mathbf{y}_-)] + \alpha \mathbb{E}[g_\theta(\mathbf{x}, \mathbf{y}_-)^2]\}.
\end{aligned}
$$

Dataset (→)	StrategyQA		GSM8K		TruthfulQA		ScienceQA	
Adapter (↓) / Metrics (→)	Acc. (%)	Δ (%)	Acc. (%)	Δ (%)	True + Info (%)	Δ (%)	Acc. (%)	Δ (%)
gpt-3.5-turbo (OpenAI, 2022)	66.59	-	67.51	-	77.00	-	72.90	-
Azure-SFT (Peng et al., 2023)	76.86	+10.27	69.94	+2.43	95.00	+18.00	79.00	+6.10
BBox-Adapter (Ground-Truth)	71.62	+5.03	73.86	+6.35	79.70	+2.70	78.53	+5.63
BBox-Adapter (AI Feedback)	69.85	+3.26	73.50	+5.99	82.10	+5.10	78.30	+5.40
BBox-Adapter (Combined)	**72.27**	**+5.68**	**74.28**	**+6.77**	**83.60**	**+6.60**	**79.40**	**+6.50**

- Empirical experiments show that **BBox-Adapter** consistently outperforms gpt-3.5-turbo by an average of **6.39%** across all datasets, highlighting its efficacy in adapting black-box LLMs to various tasks.

Plugger (→)	BBox-Adapter (gpt-3.5-turbo)							
Dataset (→)	StrategyQA		GSM8K		TruthfulQA		Average	
Black-Box LLMs (↓) / Metrics (→)	Acc. (%)	Δ (%)	Acc. (%)	Δ (%)	True + Info (%)	Δ (%)	Acc. (%)	Δ (%)
davinci-002	44.19	-	23.73	-	31.50	-	33.14	-
davinci-002 (Plugged)	**59.61**	**+15.42**	**23.85**	**+0.12**	**36.50**	**+5.00**	**39.99**	**+6.85**
Mixtral-8×7B	59.91	-	47.46	-	40.40	-	49.26	-
Mixtral-8×7B (Plugged)	**63.97**	**+4.06**	**47.61**	**+0.15**	**49.70**	**+9.30**	**53.76**	**+4.50**

- **Plug-and-Play**. Our trained adapter demonstrates an average performance improvement of **6.85%** and **4.50%** across all datasets.

Dataset (→)	StrategyQA			GSM8K		
Adapter (↓) / Metric (→)	**Acc.(%)**	**Training Cost ($)**	**Inference Cost ($)/1k Q**	**Acc.(%)**	**Training Cost ($)**	**Inference Cost ($)/1k Q**
`gpt-3.5-turbo`	66.59	-	0.41	67.51	-	1.22
Azure-SFT (Peng et al., 2023)	76.86	153.00	7.50	69.94	216.50	28.30
BBOX-ADAPTER (Single-step)	69.87	2.77	2.20	71.13	7.54	3.10
BBOX-ADAPTER (Full-step)	71.62	3.48	5.37	74.28	11.58	12.46

- **Cost-efficiency**. BBox-Adapter achieves **5.90%** improvement over the base model with **31.30** times less training cost and **1.84** times less inference cost than the official SFT service.

Case study on GSM8K

Q: An airport has only 2 planes that fly multiple times a day. Each day, the first plane goes to Greece for three-quarters of its flights, and the remaining flights are split equally between flights to France and flights to Germany. The other plane flies exclusively to Poland, and its 44 trips only amount to half the number of trips the first plane makes throughout each day. How many flights to France does the first plane take in one day?

CoT Result

The first plane makes 4 trips to Greece (3/4 of its flights), and the remaining 1/4 is split equally between France and Germany.
So, it makes 1/4 * 4 = 1 trip to France.
The second plane makes 44 trips, which is half the number of trips the first plane makes.
So, the first plane makes 44 * 2 = 88 trips in total.
Therefore, the first plane takes 1 trip to France in one day.
The answer is 1 trip to France. ❌

Adapted Result

The second plane makes half the number of trips as the first plane, so the first plane makes 44 x 2 = <<44*2=88>>88 trips in one day.
The first plane goes to Greece for three-quarters of its flights, which means 88 x 3/4 = <<88*3/4=66>>66 flights to Greece.
The remaining flights are split equally between France and Germany, so the first plane makes (88-66)/2 = 11 flights to France.
The answer is 11 flights to France. ✅

Search Path with Bbox-Adapter

Step 1

- Let's first find out how many flights the second plane makes per day.
- Let's start by figuring out how many flights the first plane makes in one day.
- **The second plane makes half the number of trips as the first plane, so the first plane makes 44 x 2 = <<44*2=88>>88 trips in one day.**

Step 2

- There are a total of 44 x 2 = <<44*2=88>>88 flights made by the first plane in one day.
- **The first plane goes to Greece for three-quarters of its flights, which means 88 x 3/4 = <<88*3/4=66>>66 flights to Greece.**
- If the second plane's 44 trips amount to half the number of trips the first plane makes, the first plane makes 44*2=88 trips per day.

Step 3

- **The remaining flights are split equally between France and Germany, so the first plane makes (88-66)/2 = 11 flights to France.**
- The remaining 88 - 66 = <<88-66=22>>22 flights are split equally between France and Germany.
- The remaining flights are split equally between flights to France and flights to Germany, so there are (88-66)/2 = <<(88-66)/2=11>>11 flights to France.

Step 4

- **#### The answer is 11 flights to France.**
- #### The answer is 11.
- #### The answer is 11.

Thank You!

BBox-Adapter: Lightweight Adapting for Black-Box Large Language Models

Haotian Sun[1]*, Yuchen Zhuang[1]*, Wei Wei[2], Chao Zhang[1], Bo Dai[1]

[1] Georgia Institute of Technology
[2] Accenture

Paper Code Website