Bayesian Optimization with Conformal Prediction Sets

Samuel Stanton[1,2]
Prescient Design, Genentech[1]

Wesley Maddox[2]

Andrew Gordon Wilson[2]
New York University[2]

Why standard BO fails

- I. Model misspecification
- -> homoscedastic likelihood, but true noise is heteroscedastic
- -> Matern kernel cannot capture discontinuities
- Result: bayesian credible sets have POOR empirical coverage
- E.g. "95% credible interval" may only cover 60–70% of true values in practice

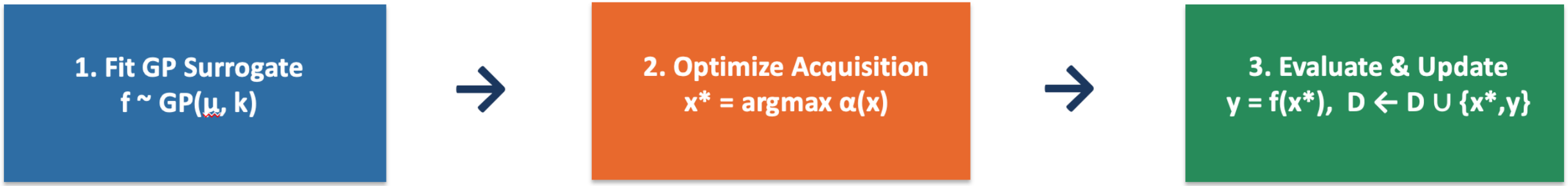
Why standard BO fails

- II. Feedback Covariate Shift
- -> **BayesOpt actively steers queries toward high-value regions:**
- Result: Query distribution $p'(x) \neq$ Training distribution $p(x)$

- Credible sets are calibrated for training distribution,
- **NOT for the query distribution.**
- Coverage degrades precisely where the optimizer
- **is spending its evaluation budget!**

BO

- Common Acquisition functions
  EI, UCB…
- LOOP:



- Problem: All standard acquisition functions rely on the GP posterior p(f | D) — which may be poorly calibrated due to misspecification and covariate shift.

Conformal Prediction

- Distribution-free prediction sets with coverage guarantees
- Coverage Guarantee (model-free, no distributional assumptions):

$$\mathbb{P}[\mathbf{y}_n \in \mathcal{C}_\alpha(\mathbf{x}_n)] \geq 1 - \alpha,$$

How it works

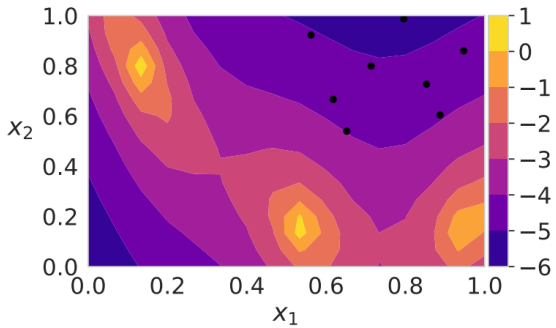
- 1. Define score function $\log p(\mathbf{y}_i|\mathbf{x}_i, \hat{\mathcal{D}})$
  Higher score → y is more consistent with the data
- 2. Compute scores for all calibration points
- 3. Build prediction set — include ŷ if score rank ≥ α quantile:

$$\mathcal{C}_\alpha(\mathbf{x}_n) := \left\{ \hat{\mathbf{y}}_n \in \mathcal{Y} \,\middle|\, \mathbf{h}^\top \mathbf{w} > \alpha \right\},$$

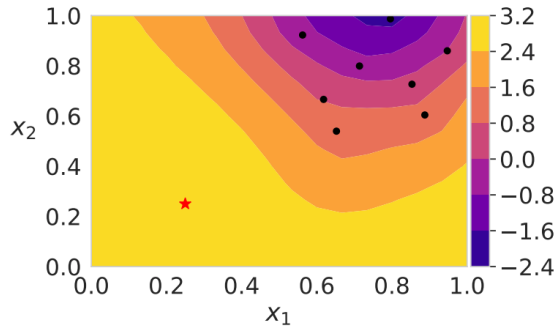
$$\text{where } h_i := \mathbb{1}\left\{ s(\mathbf{x}_i, \mathbf{y}_i, \hat{\mathcal{D}}) \leq s(\mathbf{x}_n, \hat{\mathbf{y}}_n, \hat{\mathcal{D}}) \right\},$$

- 4. Importance weights correct for covariate shift

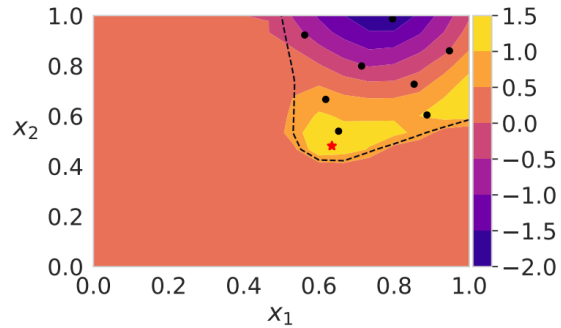
$$w_i \;\propto\; r(\mathbf{x}_i) \;=\; p'(\mathbf{x}_i|\hat{\mathcal{D}}_{-i})/p(\mathbf{x}_i)$$



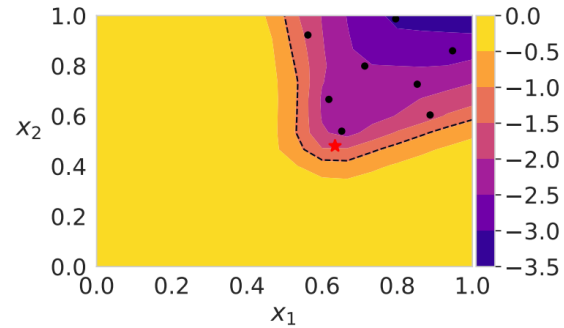
(a) Objective fn.



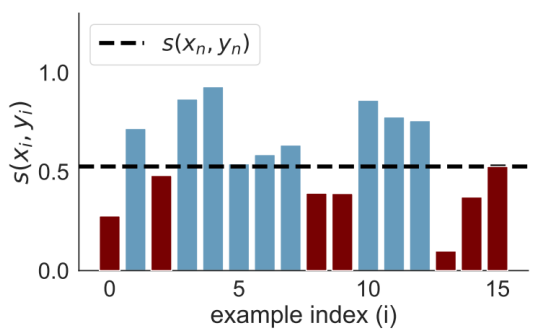
(b) UCB



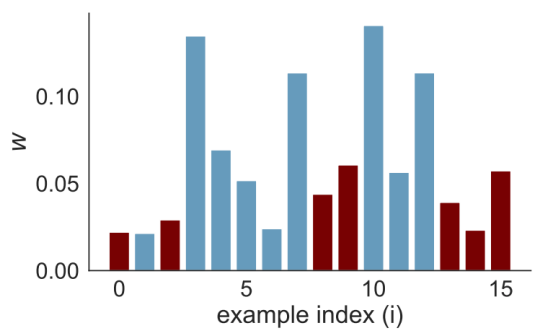
(c) CUCB



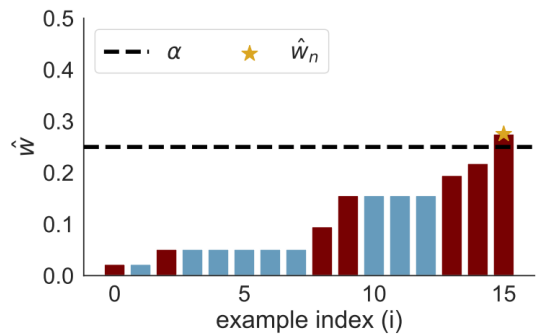
(d) Normalized density ratio



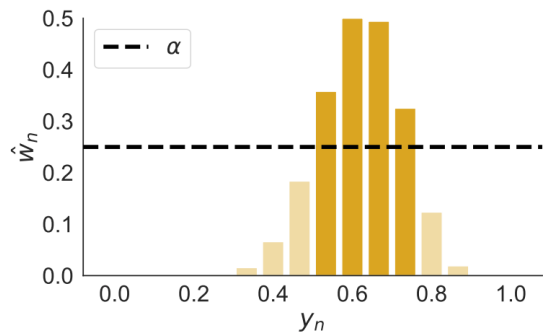
(a) conformal scores $\mathbf{s}$



(b) imp. weights $\mathbf{w}$



(c) IW partial sums $\overline{\mathbf{w}}$



(d) $\mathcal{C}_\alpha(\mathbf{x}_n)$

Conformal Bayes Posterior

- Standard Bayes Posterior (mixture form)

$$p(f(\mathbf{x})|\mathcal{D}) = \int_{\hat{\mathbf{y}}\in\mathcal{Y}} p(f(\mathbf{x})|\hat{\mathcal{D}})p(\hat{\mathbf{y}}|\mathbf{x},\mathcal{D})d\hat{\mathbf{y}}.$$

- Conformal Mixing Density

Replace p(ŷ | x, D) with conformal mixing density p_α(ŷ | x, D)

$$p_\alpha(\hat{\mathbf{y}}|\mathbf{x},\mathcal{D}) := \begin{cases} (1-\alpha)/Z_1 & \text{if } \hat{\mathbf{y}} \in C_\alpha(\mathbf{x}), \\ \alpha p(\hat{\mathbf{y}}|\mathbf{x},\mathcal{D})/Z_2 & \text{else,} \end{cases}$$

$$Z_1 = \int_{\hat{y}\in C_\alpha(x)} d\hat{y} \qquad Z_2 = \int_{\hat{y}\notin C_\alpha(x)} p(\hat{y} \mid x,\mathcal{D})\, d\hat{y}$$

$$p_\alpha(f(\mathbf{x})|\mathcal{D}) := \frac{1-\alpha}{Z_1} \int_{\hat{\mathbf{y}} \in C_\alpha(\mathbf{x})} p(f|\hat{\mathcal{D}}) d\hat{\mathbf{y}}$$

$$+ \frac{\alpha}{Z_2} \int_{\hat{\mathbf{y}} \in \mathcal{Y} \setminus C_\alpha(\mathbf{x})} p(f(\mathbf{x})|\hat{\mathcal{D}}) p(\hat{\mathbf{y}}|\mathbf{x}, \mathcal{D}) d\hat{\mathbf{y}}$$

where Z1, Z2 are normalization constants

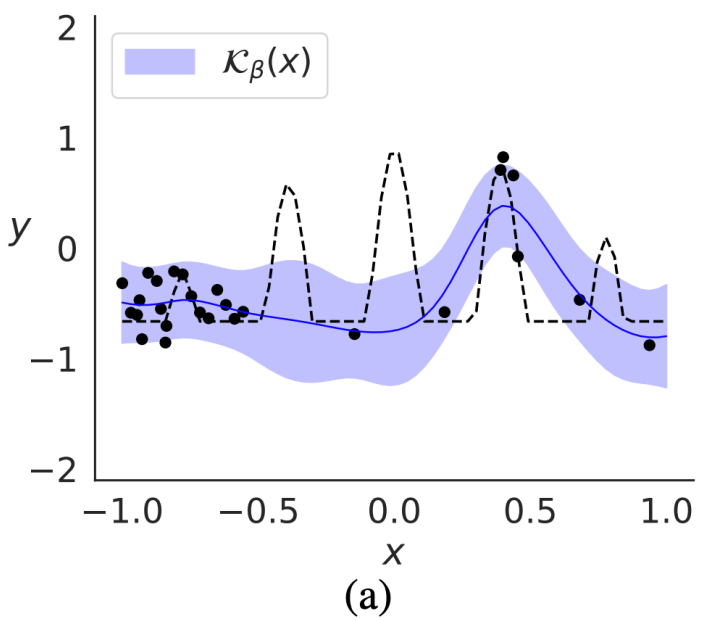
Inside C_α(x): treat all outcomes as equally plausible

→ The coverage guarantee says any of them could be the truth

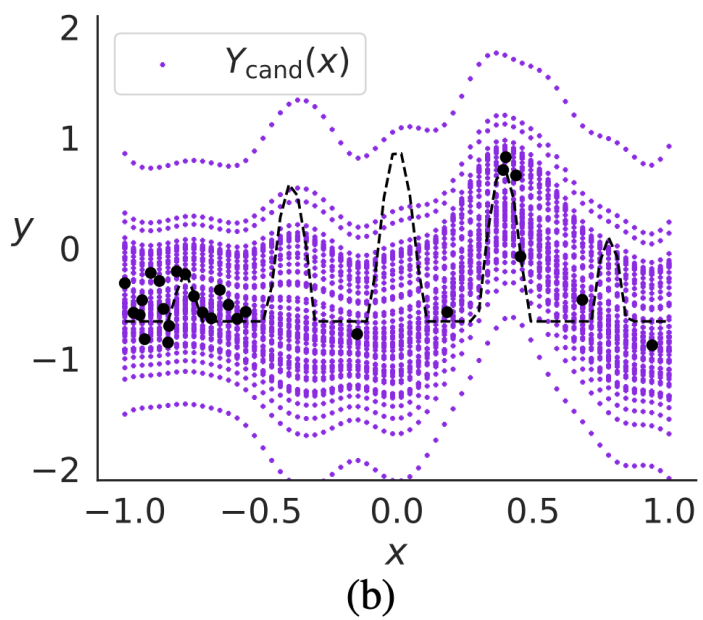
Outside C_α(x): retain GP predictive weighting

→ Very unlikely outcomes still down-weighted by the model

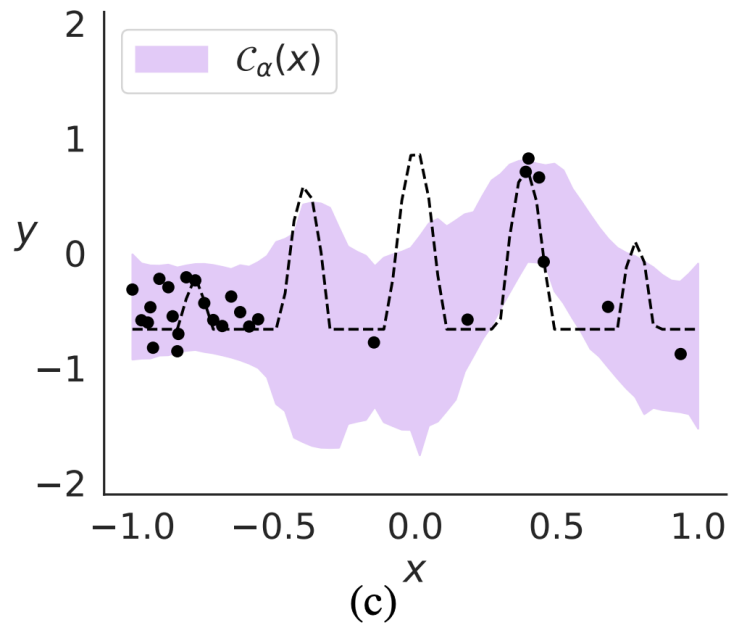
As α → 1: p_α → standard Bayes posterior



(a)



(b)



(c)

n=27 noisy observation,
α=1−β=0.19

Challenge 1: Making Conformal Prediction Differentiable

The Problem

Indicator $1[\hat{y} \in C_\alpha(x)]$ is a Heaviside step function

→ **Gradient is zero almost everywhere**
→ **Cannot optimize acquisition via gradient descent**

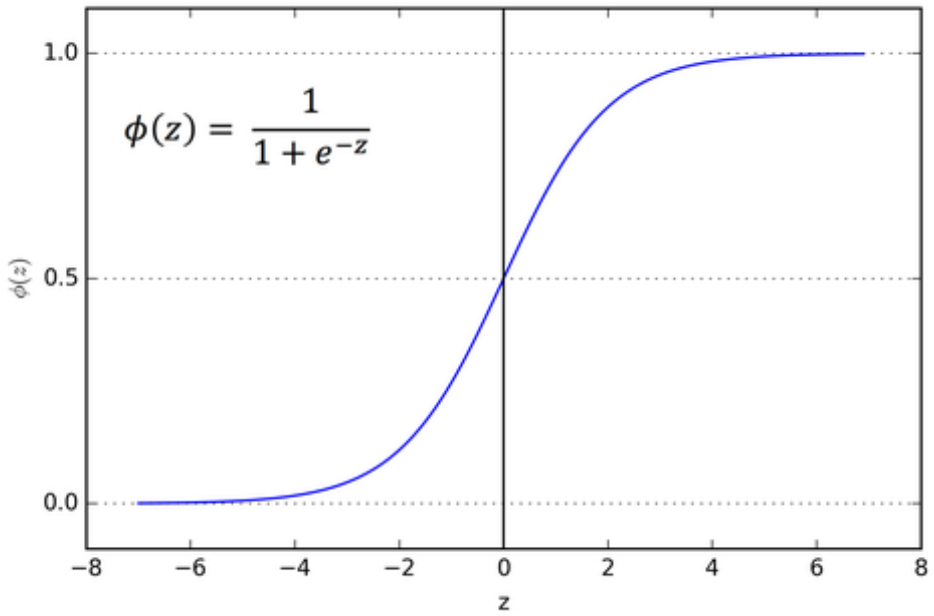
The Solution: Sigmoid Relaxation

Replace Heaviside with soft sigmoid mask:

$$m_j = \sigma(\tau^{-1} \cdot (w - \alpha))$$

$\tau \to 0$: exact conformal | τ moderate: differentiable

Sigmoid function ->



Algorithm 1 Differentiable conformal prediction masks

Data: train data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=0}^{n-1}$, test point $\mathbf{x}_n$, imp. weights $\mathbf{w}$, label candidates Y_{cand}, score function s, miscoverage tolerance α, relaxation strength $\tau > 0$.

$m_j = 0, \forall j \in \{0, \ldots, k-1\}$.

for $\hat{\mathbf{y}}_j \in Y_{\text{cand}}$ **do**

$\hat{\mathcal{D}} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_n, \hat{\mathbf{y}}_j)\}$

$\mathbf{s} \leftarrow [s(\mathbf{x}_0, \mathbf{y}_0, \hat{\mathcal{D}}) \ \cdots \ s(\mathbf{x}_n, \hat{\mathbf{y}}_j, \hat{\mathcal{D}})]^\top$.

$\mathbf{h} \leftarrow \texttt{sigmoid}(\tau^{-1}(\mathbf{s} - s_n))$.

$\overline{w} \leftarrow \mathbf{h}^\top \mathbf{w}$.

$m_j \leftarrow \texttt{sigmoid}(\tau^{-1}(\overline{w} - \alpha))$.

end

Result: $\mathbf{m}$

Challenge 2: Estimating the Density Ratio

- Query distribution p′(x|D) is implicit — not available in closed form!
- We need: $r(x) = \frac{p'(x \mid \mathcal{D})}{p(x)}$

Step 1: Sample from p′(x|D)

Run SGLD (Stochastic Gradient Langevin Dynamics) on the acquisition surface to generate samples approximating the implicit query distribution

$$p'(x \mid \mathcal{D}) \propto \exp\{a_\alpha(x, \mathcal{D})\}$$

→

Step 2: Train Binary Classifier

Label training data D as class 0, SGLD samples as class 1. Train classifier q_θ(z|x) to distinguish them

→

Step 3: Estimate Density Ratio

r(x) ∝ p(z=1|x) / p(z=0|x) (derived from Bayes' theorem on the classifier output)

→

Step 4: Stabilize with EMA

Use Exponential Moving Average of classifier weights to stabilize the SGLD chain (inspired by Bootstrapped DQN)

Let

$$r(x) = \frac{p'(x \mid \mathcal{D})}{p(x)}.$$

Introduce a binary label z such that

- $z = 0$ means x is drawn from the training distribution, so $p(x \mid z = 0) = p(x)$,
- $z = 1$ means x is drawn from the query distribution, so $p(x \mid z = 1) = p'(x \mid \mathcal{D})$.

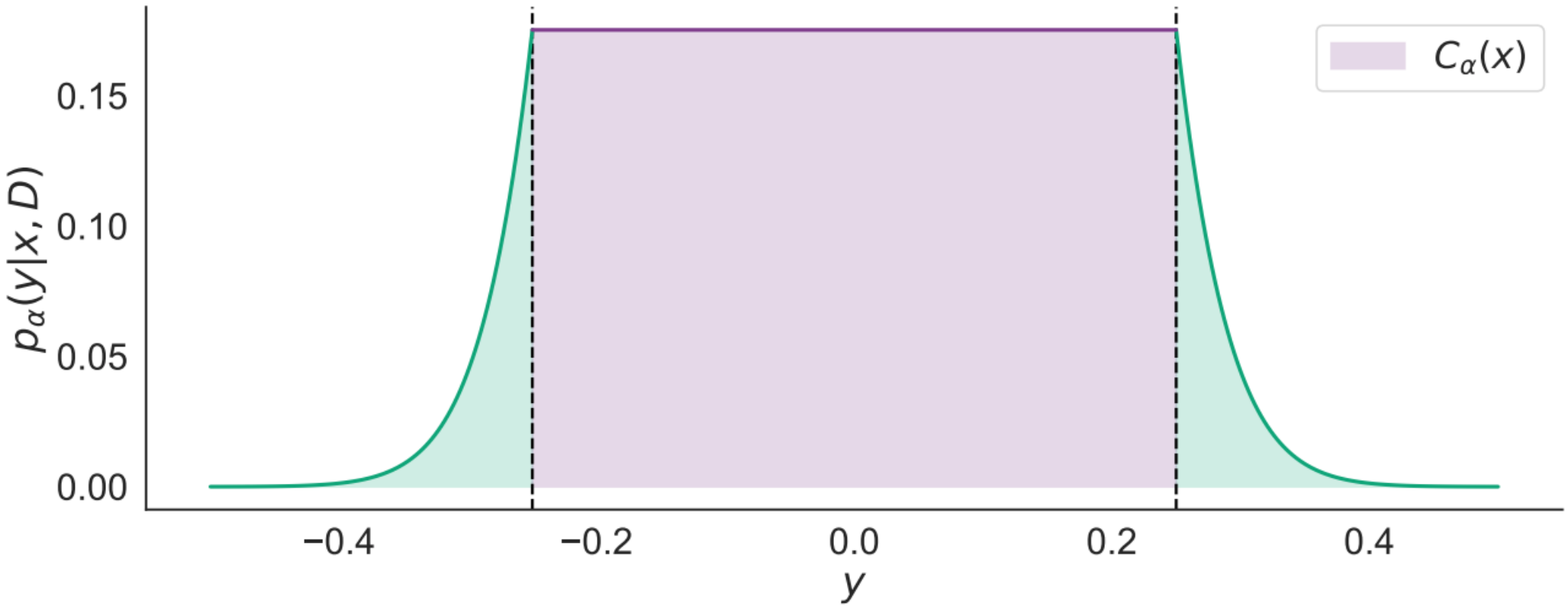
$$r(x) = \frac{p(x \mid z = 1)}{p(x \mid z = 0)}.$$

$$p(x \mid z = 1) = \frac{p(z = 1 \mid x)\, p(x)}{p(z = 1)},$$

$$p(x \mid z = 0) = \frac{p(z = 0 \mid x)\, p(x)}{p(z = 0)}.$$

$$r(x) = \frac{\frac{p(z=1|x)\, p(x)}{p(z=1)}}{\frac{p(z=0|x)\, p(x)}{p(z=0)}}. \quad r(x) = \frac{p(z = 0)}{p(z = 1)} \frac{p(z = 1 \mid x)}{p(z = 0 \mid x)}$$

train a probabilistic classifier pˆ(z | x) to discriminate the sample labels

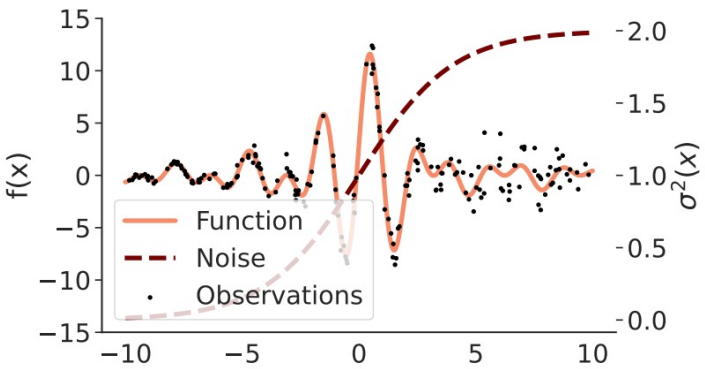

$C_\alpha(x)$
0.15
0.10
0.05
0.00
$p_\alpha(y|x, D)$
−0.4
−0.2
0.0
0.2
0.4
y

Conformal Acquisition

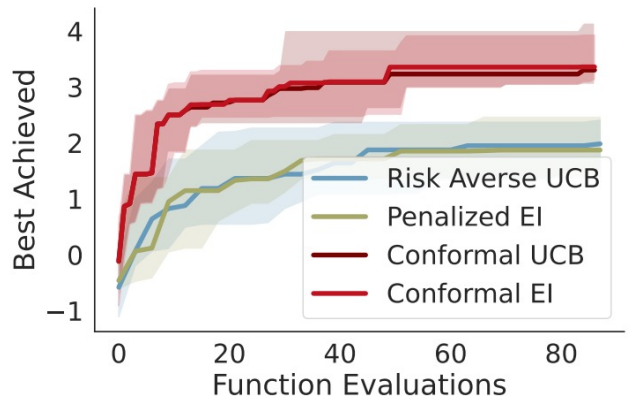
$$
\begin{aligned}
a_\alpha(\mathbf{x}, \mathcal{D}) &= \int u(\mathbf{x}, f, \mathcal{D}) p_\alpha(f|\mathcal{D}) df, \\
&\approx (1-\alpha)\mathbf{u}^\top \mathbf{v} + \alpha \mathbf{u}^\top \mathbf{v}', \\
\textbf{where } \mathbf{u} &= [u(\mathbf{x}, f^{(0)}, \mathcal{D}) \ \cdots \ u(\mathbf{x}, f^{(k-1)}, \mathcal{D})]^\top, \\
\mathbf{v}_i &= \frac{m_i}{p(\hat{\mathbf{y}}_i|\mathbf{x}, \mathcal{D})} \left(\sum_j \frac{m_j}{p(\hat{\mathbf{y}}_j|\mathbf{x}, \mathcal{D})} \right)^{-1}, \\
\mathbf{v}_i' &= (1-m_i)(\mathbf{1}^\top(\mathbf{1}-\mathbf{m}))^{-1},
\end{aligned}
$$

Conformal BayesOpt: Full Alogorithm

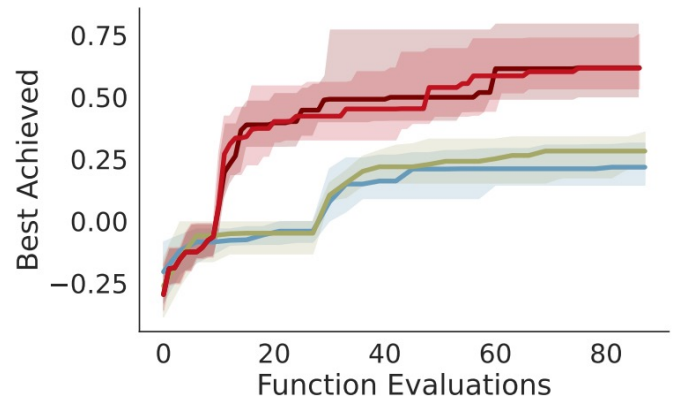
Initialize	Set initial query x_n, fit GP on D
Estimate r(x)	SGLD sampling + classifier training → density ratio
Sample Y_cand	Draw candidate labels from GP posterior $p(y \mid x, D)$
Compute Masks	Sigmoid-relaxed conformal masks m_j for each $\hat{y}_j$
Conformal Acq.	Evaluate C-UCB / C-EI / C-NEI using conformal Bayes posterior
SGLD Step	Update x_n via gradient step on acquisition surface
Query & Update	Evaluate f(x_n), add to D, update GP → repeat



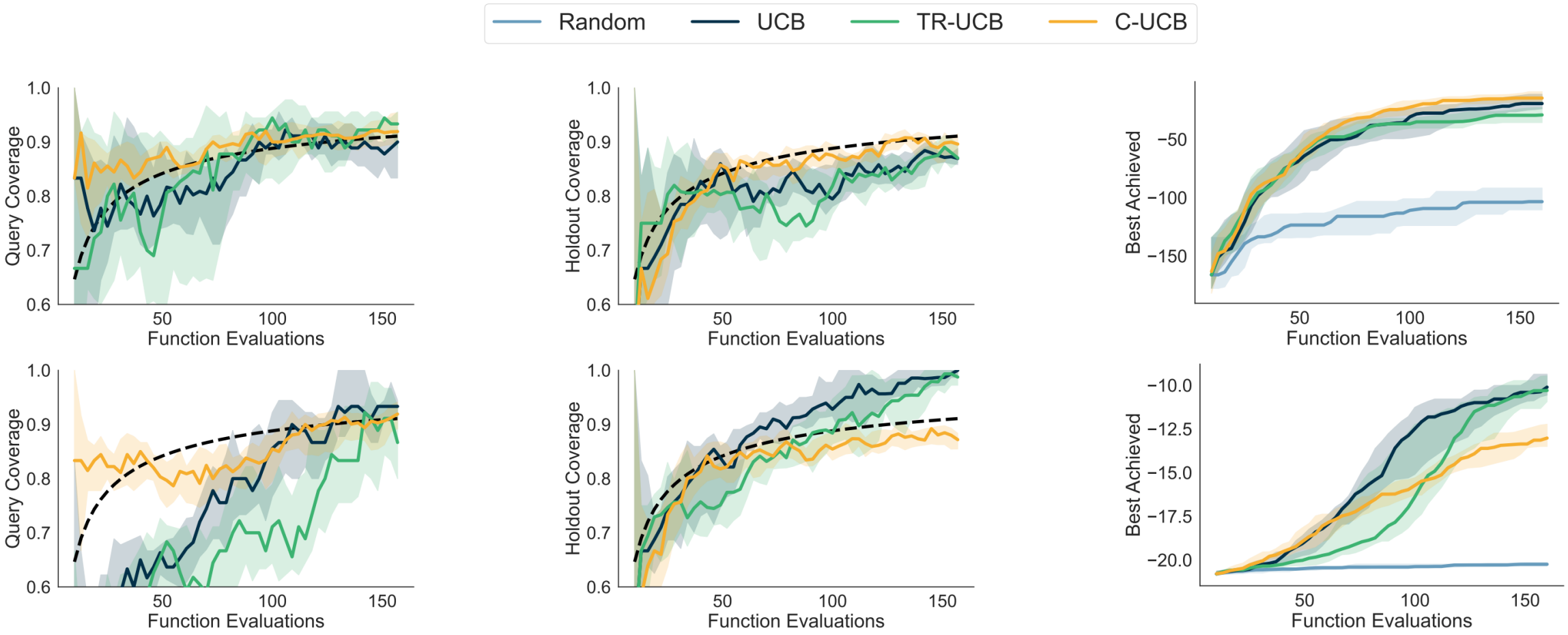
(a) `sinc` objective fn.



(b) `sinc`



(c) `double-knot`


Random
UCB
TR-UCB
C-UCB
Query Coverage
Holdout Coverage
Best Achieved
Function Evaluations
1.0
0.9
0.8
0.7
0.6
−50
−100
−150
−10.0
−12.5
−15.0
−17.5
−20.0
50
100
150

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao[*,1], Jeffrey Zhao[2], Dian Yu[2], Nan Du[2], Izhak Shafran[2], Karthik Narasimhan[1], Yuan Cao[2]

[1]Department of Computer Science, Princeton University
[2]Google Research, Brain team
[1]{shunyuy,karthikn}@princeton.edu
[2]{jeffreyzhao,dianyu,dunan,izhak,yuancao}@google.com

Motivation

- Humans naturally combine verbal reasoning with physical actions
  – Track progress, handle exceptions, decide when to seek external information

- LLMs can reason (Chain-of-Thought) but reasoning is a static black box
  – No grounding → hallucination, error propagation

- LLMs can act (plan generation, tool use) but lack high-level reasoning
  – Lose track of goals, fail to recover from unexpected states

- Can we combine reasoning and acting synergistically?

The ReAct Framework

Core idea: augment agent's action space Â = A ∪ L

- A = environment actions L = language space (thoughts)
- Thoughts do not affect the environment but update the context for future steps

Thought–Action–Observation loop:

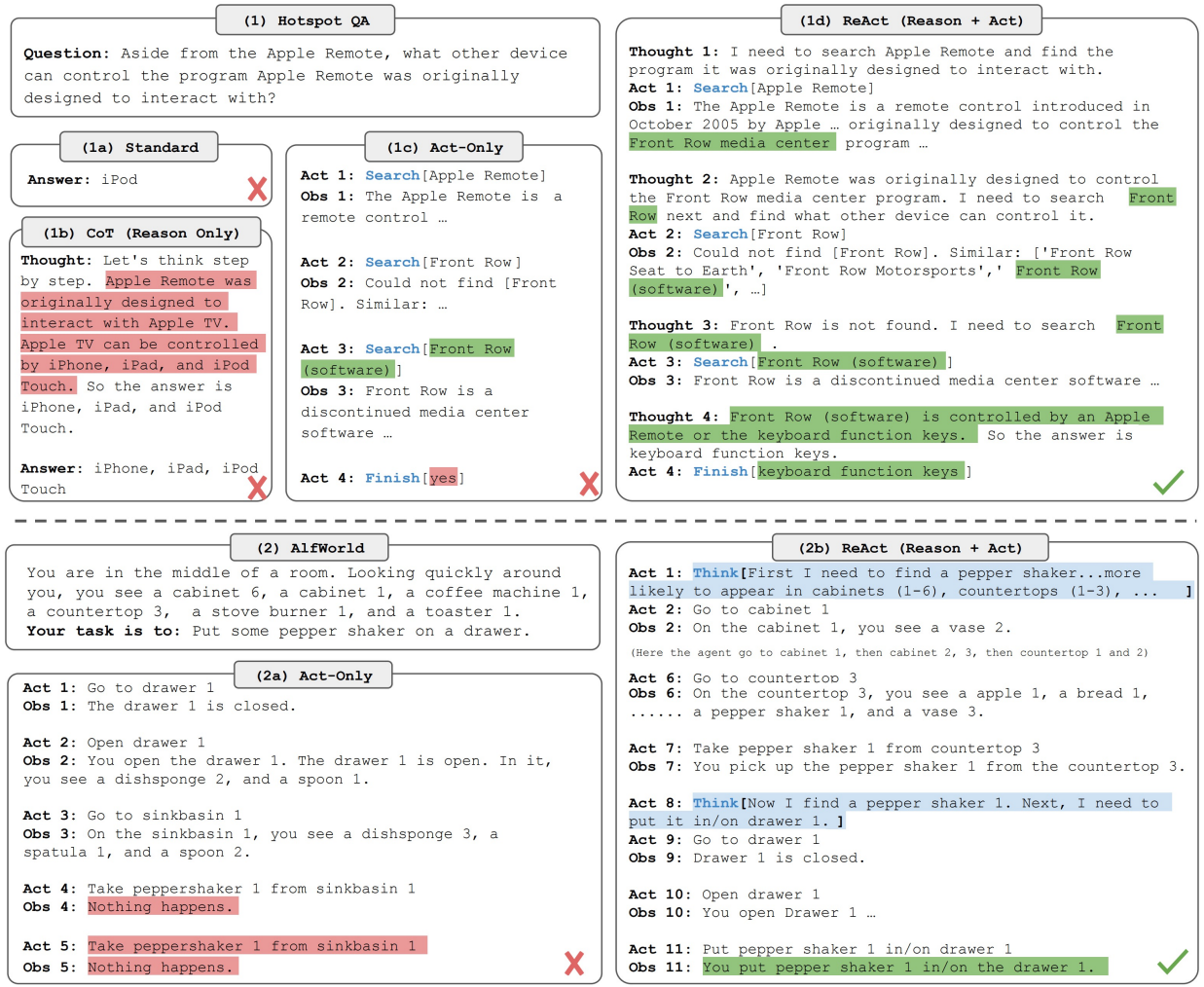
Thought → reason about current state, plan next step

Action → interact with external environment (e.g., Wikipedia search)

Observation → receive feedback from environment

Implementation: few-shot prompting of frozen PaLM-540B

– Each in-context example is a human-written trajectory of (Thought, Action, Obs) steps



Experimental Setup

Four benchmarks:

Knowledge-intensive reasoning
- HotpotQA – multi-hop QA over Wikipedia (metric: EM)
- FEVER – fact verification: SUPPORTS / REFUTES / NOT ENOUGH INFO (metric: Acc)

Action space: search[entity], lookup[string], finish[answer]

Interactive decision making
- ALFWorld – text-based household game, 6 task types, 134 eval games
- WebShop – online shopping with 1.18M products, 500 test instructions

Baselines: Standard prompting, Chain-of-Thought (CoT), CoT-SC, Act-only
Base model: PaLM-540B (few-shot prompting); also PaLM-8B/62B for fine-tuning

Results: Knowledge-Intensive Tasks (HotpotQA & FEVER)

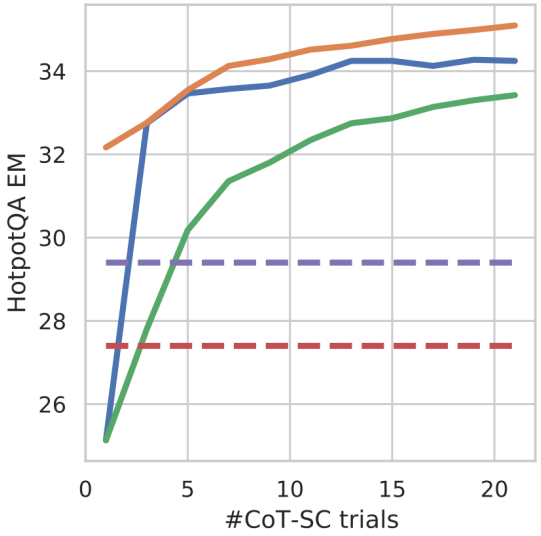
• ReAct > Act-only on both tasks: reasoning helps guide search and synthesize answers

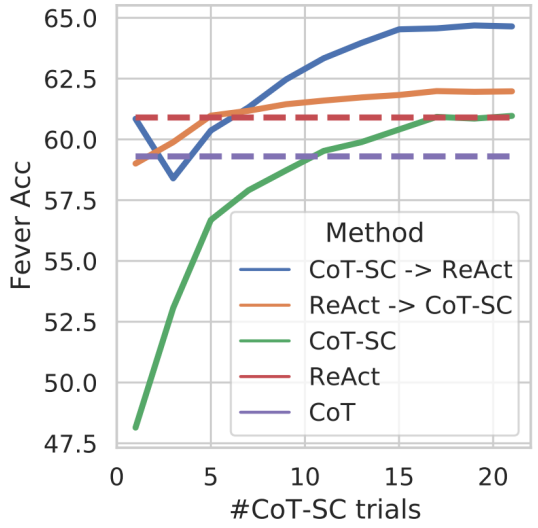
• ReAct vs CoT:
– FEVER: ReAct 60.9 vs CoT 56.3 (external facts crucial for verification)
– HotpotQA: ReAct 27.4 vs CoT 29.4 (CoT has more flexible reasoning structure)

• Best overall: ReAct + CoT-SC hybrid
– CoT-SC→ReAct: 64.6 Acc on FEVER
– ReAct→CoT-SC: 35.1 EM on HotpotQA
– Reaches 21-sample CoT-SC performance with only 3–5 samples

Prompt Method[a]	HotpotQA (EM)	Fever (Acc)
Standard	28.7	57.1
CoT (Wei et al., 2022)	29.4	56.3
CoT-SC (Wang et al., 2022a)	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT-SC → ReAct	34.2	**64.6**
ReAct→ CoT-SC	**35.1**	62.0
Supervised SoTA[b]	67.5	89.5

Table 1: PaLM-540B prompting results on HotpotQA and Fever.





Analysis: Success & Failure Modes on HotpotQA

Human study on 50 correct + 50 incorrect trajectories each for ReAct and CoT:

Successes:

- ReAct: 94% correct reasoning & facts (CoT: 86%)
- ReAct: only 6% hallucinated (CoT: 14%)

Failures:

- CoT failures: 56% due to hallucination — major weakness
- ReAct failures: 47% reasoning errors (incl. repetitive loops), 23% search errors
- ReAct hallucination in failures: 0%

Key trade-off: ReAct is more factual but less flexible in reasoning structure

	Type	Definition	ReAct	CoT
Success	True positive	Correct reasoning trace and facts	94%	86%
	False positive	Hallucinated reasoning trace or facts	6%	14%
Failure	Reasoning error	Wrong reasoning trace (including failing to recover from repetitive steps)	47%	16%
	Search result error	Search return empty or does not contain useful information	23%	-
	Hallucination	Hallucinated reasoning trace or facts	0%	56%
	Label ambiguity	Right prediction but did not match the label precisely	29%	28%

Fine-tuning & Scaling Results

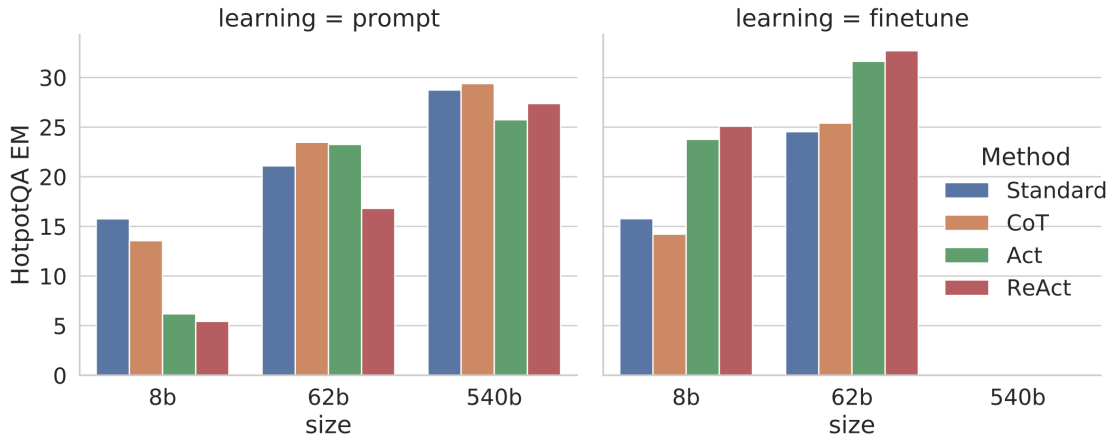
Bootstrapping fine-tuning: use 3,000 correct ReAct trajectories to fine-tune PaLM-8B / 62B

Key findings:

• Prompting: ReAct performs worst among 4 methods on small models
  (harder to learn both reasoning and acting from few examples)

• Fine-tuned ReAct becomes best method:
  – PaLM-8B fine-tuned ReAct > all PaLM-62B prompting methods
  – PaLM-62B fine-tuned ReAct > all PaLM-540B prompting methods

• Fine-tuning Standard/CoT is worse: teaches models to memorize facts
  ReAct/Act teaches models how to act — a more generalizable skill



Results: Interactive Decision Making – ALFWorld

ALFWorld: 6 household task types (Pick, Clean, Heat, Cool, Look, Pick 2)
134 unseen eval games, 6 prompt permutations per method

Results:
• ReAct (best of 6): 71% overall success vs Act (best): 45% vs BUTLER: 37%
• Even worst ReAct trial (48%) beats best Act and BUTLER
• ReAct advantage consistent across all 6 trials (relative gain 33–90%, avg 62%)

Why Act fails: cannot decompose goals or track environment state without thoughts

ReAct vs. Inner Monologue (ReAct-IM):
• IM-style dense external feedback: 53% overall
• Internal reasoning (ReAct) crucial for subgoal decomposition & commonsense

Method	Pick	Clean	Heat	Cool	Look	Pick 2	All
`Act` (best of 6)	88	42	74	67	72	**41**	45
`ReAct` (avg)	65	39	83	76	55	24	57
`ReAct` (best of 6)	**92**	58	**96**	86	**78**	**41**	**71**
`ReAct-IM` (avg)	55	59	60	55	23	24	48
`ReAct-IM` (best of 6)	62	**68**	87	57	39	33	53
BUTLER$_g$ (best of 8)	33	26	70	76	17	12	22
BUTLER (best of 8)	46	39	74	**100**	22	24	37

in a task-specific setup. For robustness, we construct 6 prompts for each task type through each permutation of 2 annotated trajectories from the 3 we annotate. `Act` prompts are constructed using

Results: Interactive Decision Making – WebShop

WebShop: 1.18M real products, 12k human instructions, 500 test episodes
Metrics: Score (% desired attributes covered) & Success Rate (SR)

Results:
• Act (one-shot): Score 62.3, SR 30.1 — already on par with IL and IL+RL
• ReAct: Score 66.6, SR 40.0 — +10% absolute improvement in SR
• Human expert: Score 82.1, SR 59.6 — still large gap to humans

Why ReAct helps: bridges gap between noisy observations and actions
– Reasoning identifies relevant product attributes from verbose text
– E.g., 'item has options 39x18x18inch and blue and seems good to buy'

Remaining gap: humans explore more products and reformulate queries

Method	Score	SR
Act	62.3	30.1
ReAct	**66.6**	**40.0**
IL	59.9	29.1
IL+RL	62.4	28.7
Human Expert	82.1	59.6

Conclusion

ReAct: a simple prompting paradigm that interleaves reasoning traces and actions

Contributions:
1. New paradigm — Thought–Action–Observation loop for general task solving
2. Strong performance across 4 diverse benchmarks with 1–6 in-context examples
3. Systematic analysis of reasoning vs. acting trade-offs
4. Interpretability: humans can inspect reasoning and correct agent behavior

Limitations:
- Reasoning errors (repetitive loops) under prompting setup
- Search failures hard to recover from
- Still far from supervised SoTA on knowledge tasks (27–35 vs. 67–89)

Future directions: RL training, more tasks, better decoding strategies