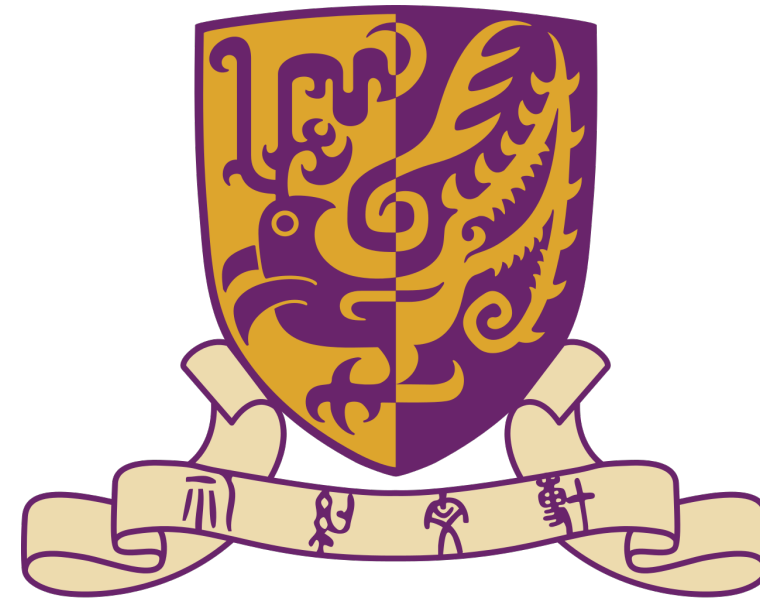


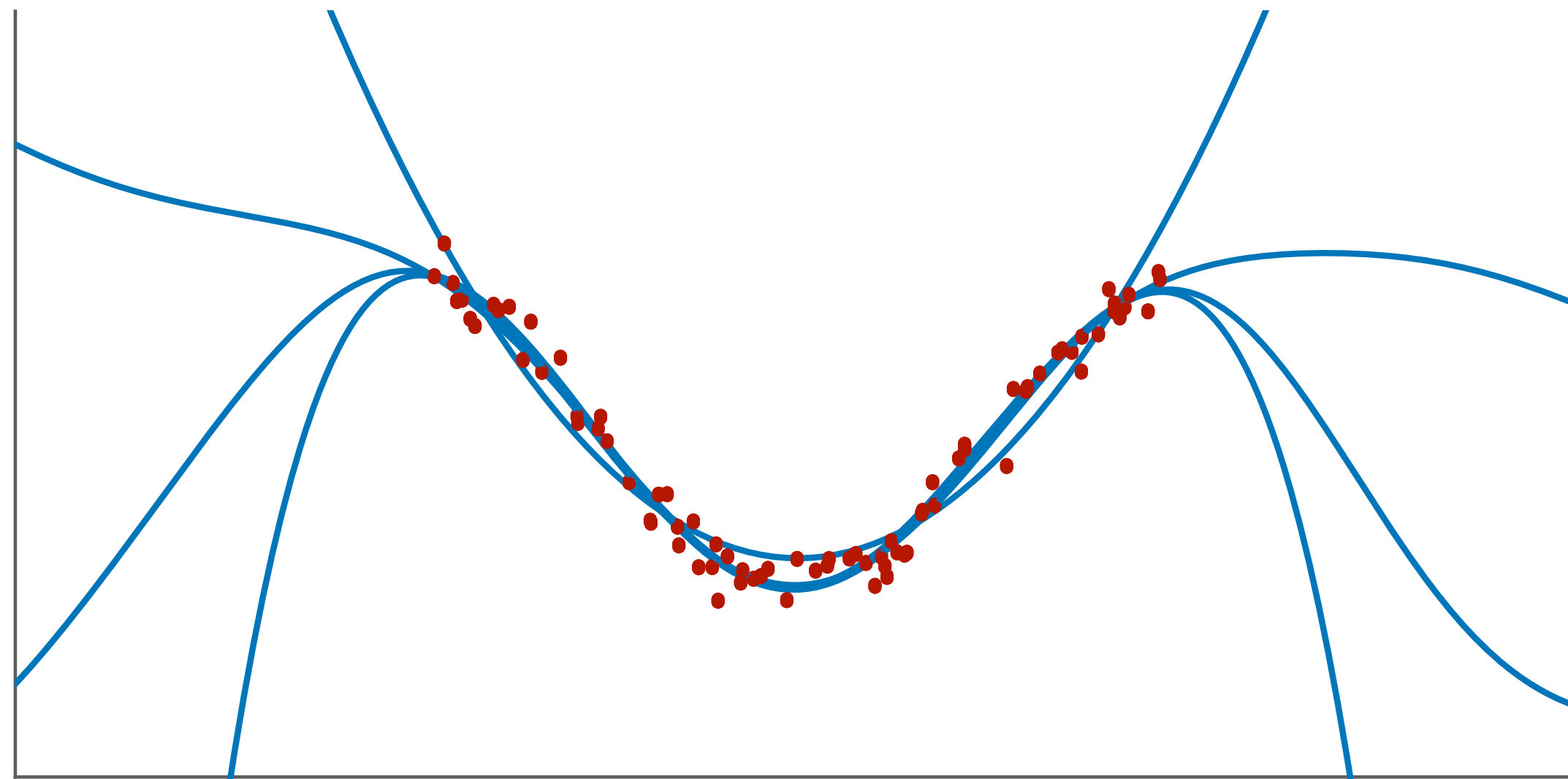


Kernel Autoresearch for Open-Ended Model Discovery

Richard Cornelius Suwandi, Feng Yin, Kevin Murphy

One World Seminar on Mathematics of Machine Learning

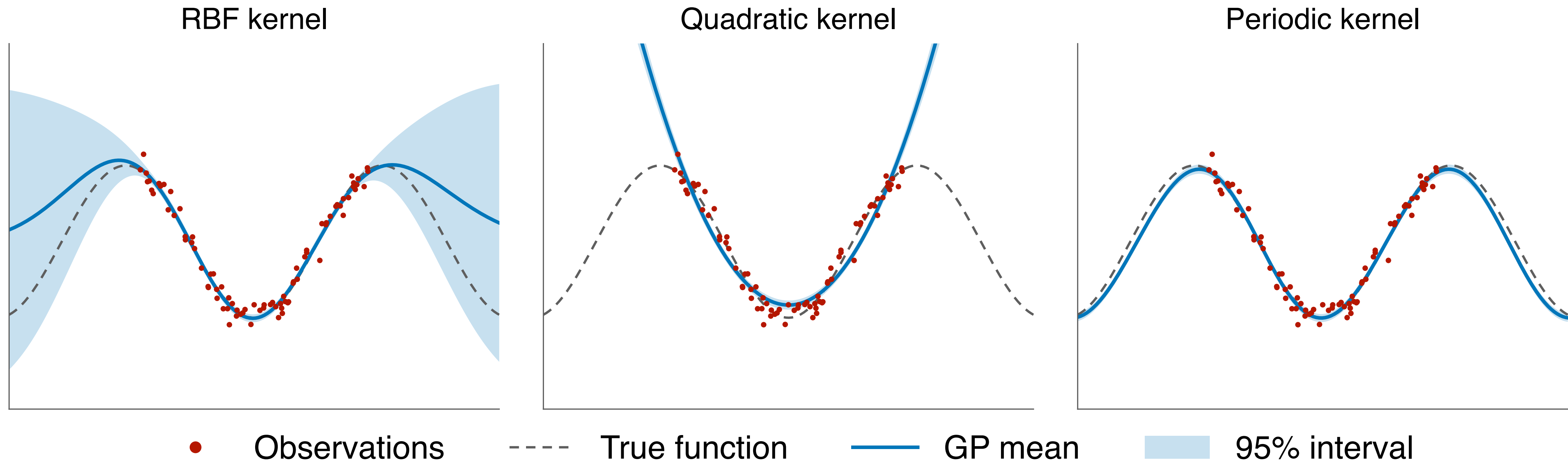
Which function should we prefer?



- Suppose we observe some **noisy samples** of an unknown function
- There are many possible functions that can fit these points
- **Inductive bias** determines which of these functions the model prefers

How can we encode **inductive bias** in a model?

Kernels as inductive biases



- Different kernels encode different **inductive biases**
- A misspecified kernel yields **biased** predictions and **miscalibrated** uncertainty

How should we choose a kernel given the data?

Compose kernels from a fixed library

*Structure Discovery in Nonparametric Regression
through Compositional Kernel Search*

[Duvenaud et al., 2013]

“built compositionally by adding and multiplying a small number of base kernels”

Method: Search over a grammar of kernel sums and products using a model-selection score

Compose kernels from a fixed library

*Structure Discovery in Nonparametric Regression
through Compositional Kernel Search*

[Duvenaud et al., 2013]

Differentiable Compositional Kernel Learning for Gaussian Processes

[Sun et al., 2018]

“each unit of the network corresponds to a valid kernel”

Method: Optimize differentiable kernel compositions with gradient-based training

Compose kernels from a fixed library

*Structure Discovery in Nonparametric Regression
through Compositional Kernel Search*

[Duvenaud et al., 2013]

Differentiable Compositional Kernel Learning for Gaussian Processes

[Sun et al., 2018]

*Adaptive Kernel Design for Bayesian Optimization
Is a Piece of CAKE with LLMs*

[Suwandi et al., 2025]

“LLMs as the crossover and mutation operators”

Method: Evolve kernel compositions using LLMs and in-context learning as data arrive

These methods over combinations, but only using a **fixed set of base kernels**

Learn flexible kernels from data

Gaussian Process Kernels for Pattern Discovery and Extrapolation

[Wilson and Adams, 2013]

“model a spectral density [...] with a Gaussian mixture”

Method: Learn a mixture of frequency components to represent a stationary kernel

Learn flexible kernels from data

Gaussian Process Kernels for Pattern Discovery and Extrapolation

[Wilson and Adams, 2013]

Deep Kernel Learning

[Wilson et al., 2016]

“we transform the inputs of a spectral mixture base kernel with a deep architecture”

Method: Jointly learn neural features and kernel parameters through the GP marginal likelihood

Learn flexible kernels from data

Gaussian Process Kernels for Pattern Discovery and Extrapolation

[Wilson and Adams, 2013]

Deep Kernel Learning

[Wilson et al., 2016]

Few-Shot Bayesian Optimization with Deep Kernel Surrogates

[Wistuba and Grabocka, 2021]

“train a **shared deep surrogate model** to **quickly adapt** (with few response evaluations)”

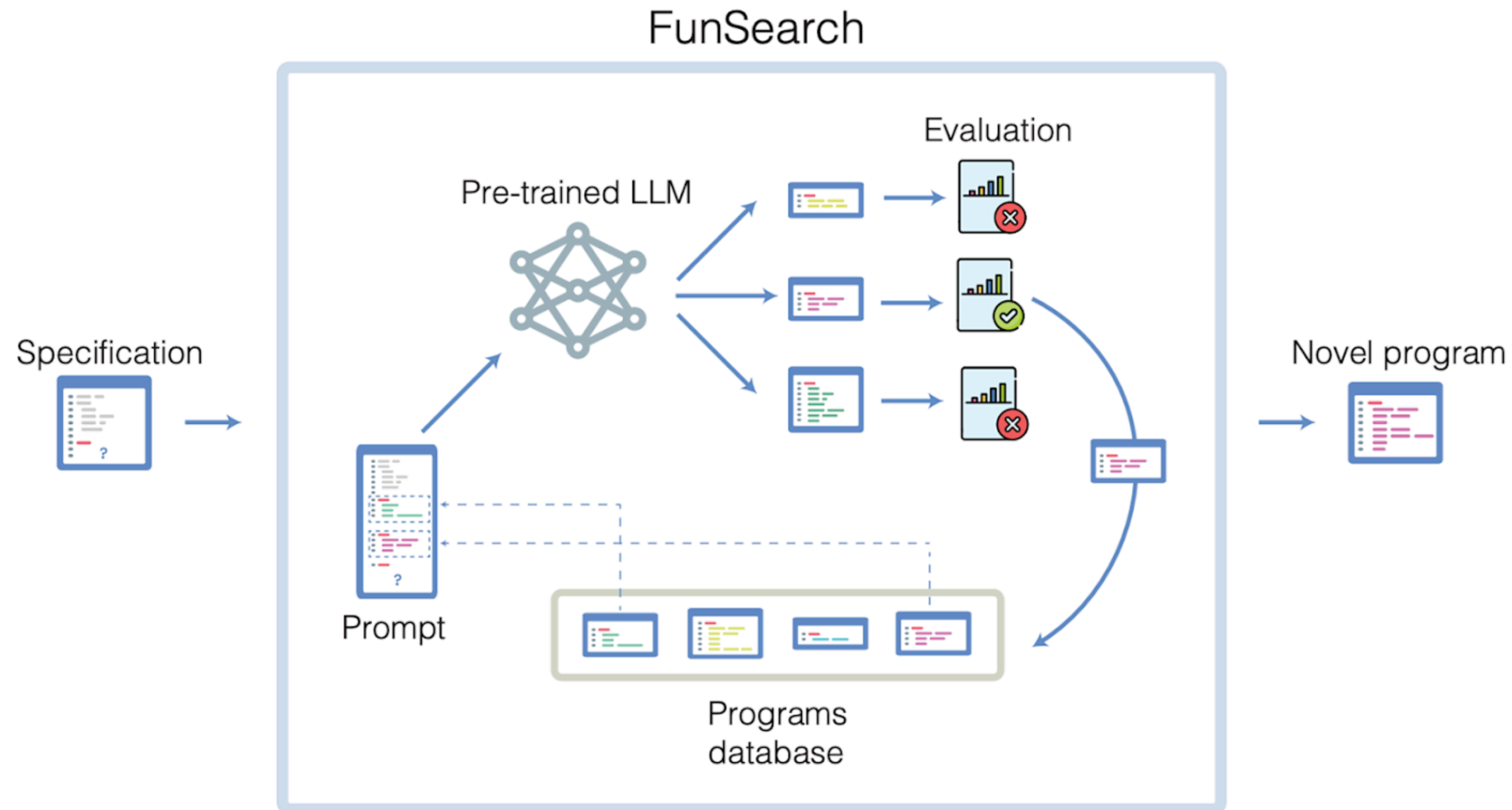
Method: Meta-learn a deep kernel across tasks, then adapt it to a new task

These method learn kernels from data, but only within a **fixed family or architecture**

Coding agents for algorithm discovery

Recent works have shown that coding agents can discover **new algorithms**

[Romera-Paredes et al, 2024; Novikov et al., 2025]



Kernel as programs

Concurrent work uses LLMs to write kernels as code

*Automated Kernel Discovery Towards
Understanding High-dimensional
Bayesian Optimization*

Yun et al., 2026

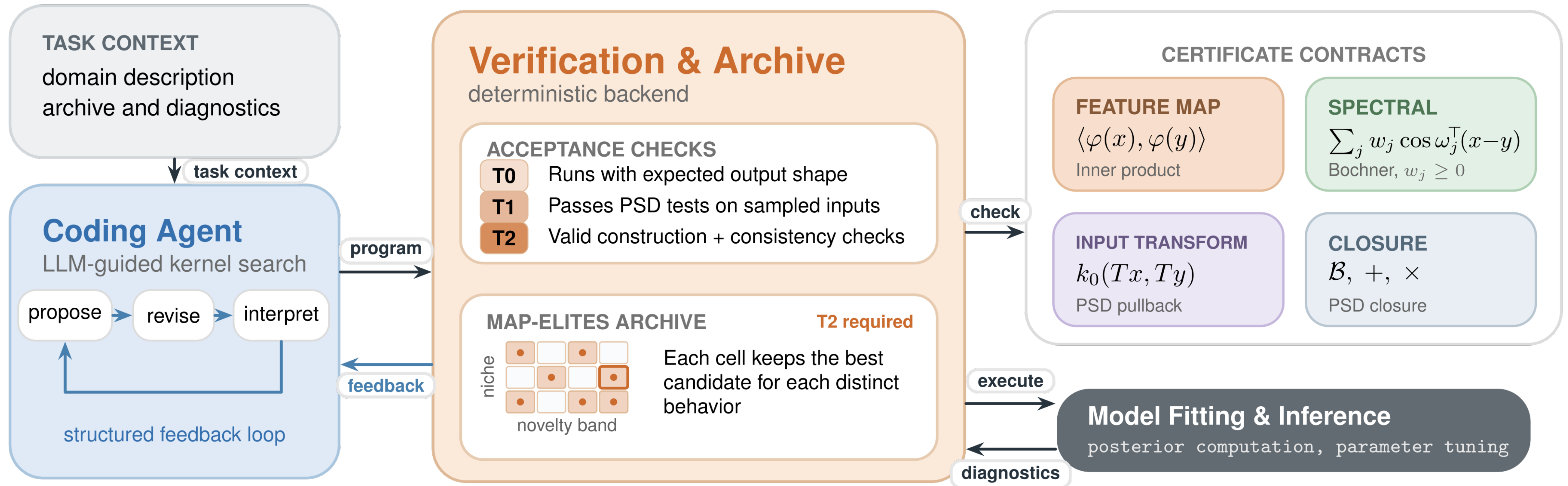
“searches a broader kernel space beyond predefined composition rules”

Method: Generate kernel code and screen the sampled Gram matrices with a Cholesky factorization.

- Kernel validity is undecidable for unrestricted programs [Rice, 1953]
- Numerical checks on **sampled** inputs cannot establish validity over the **entire** input domain
- In our stress tests, **22–58%** of LLM-written kernels that pass sampled checks fail at other scales or dims

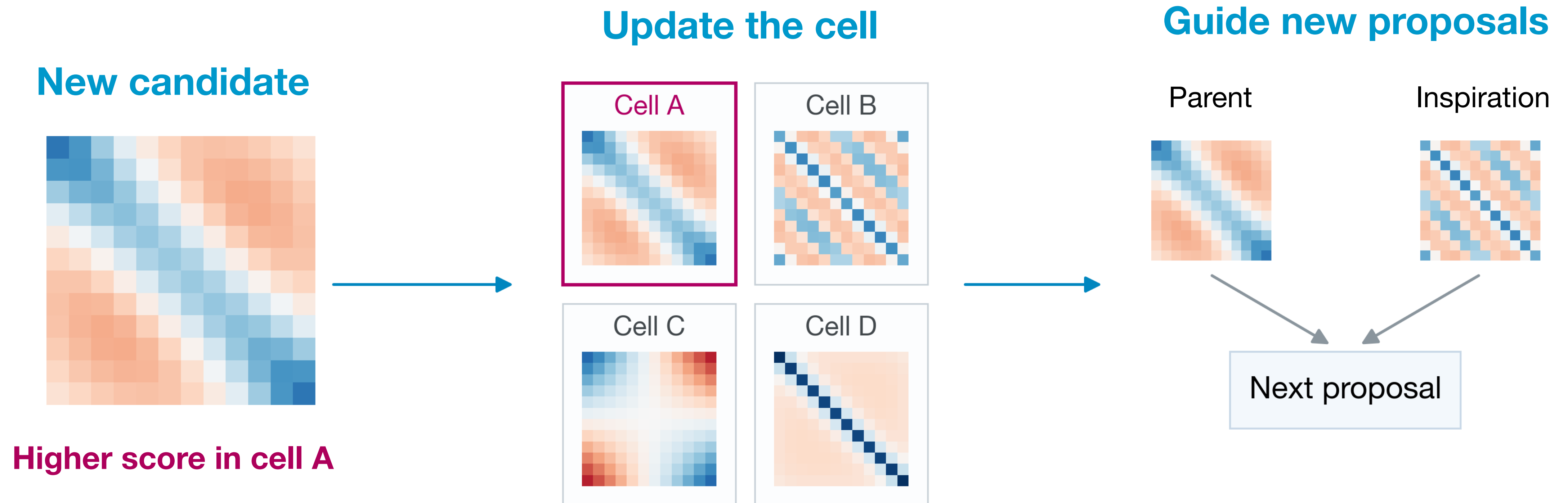
Can we search **beyond** a fixed grammar and still **guarantee** validity?

Kernel Autoresearch (Kernaut)



Quality-diversity archives

We use MAP-Elites to keep **diverse, strong** kernels during search [Mouret & Clune, 2015]



Novelty screening

- The agent may rediscover a "known" kernel under a new name/notation/implementation
- To detect this, we compare their **kernel fingerprints**:

$$d_{\text{nov}}(\kappa) = \min_{r \in \mathcal{R}} \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \left\| \bar{K}_{\kappa}(X_d) - \bar{K}_r(X_d) \right\|_F$$

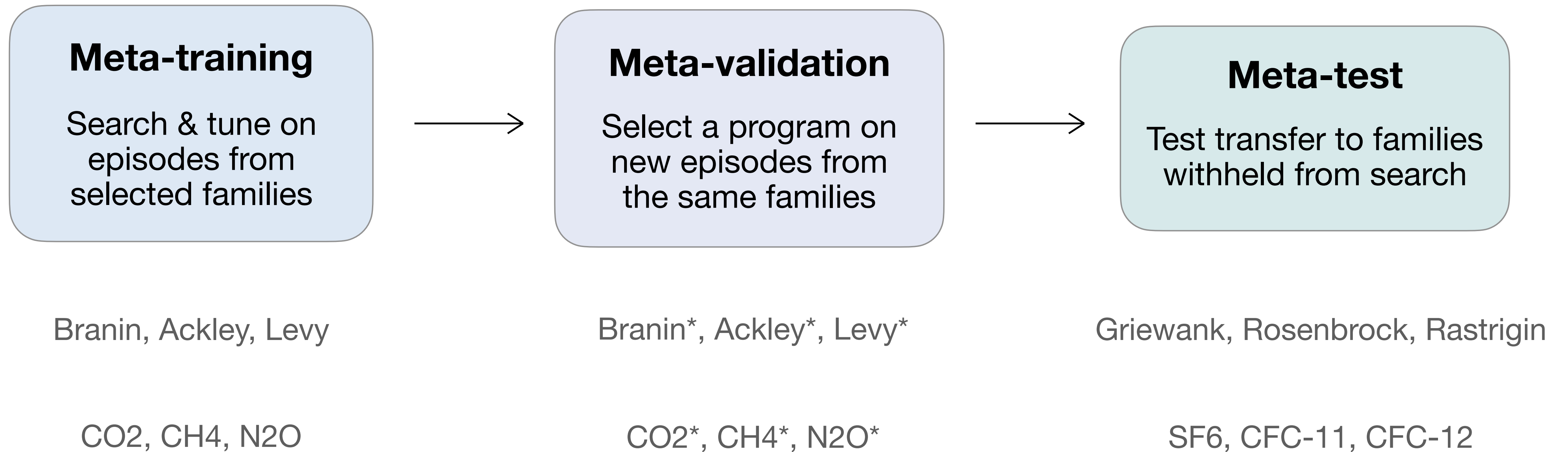
The diagram illustrates the components of the novelty distance formula $d_{\text{nov}}(\kappa)$ using color-coded brackets and labels:

- nearest known kernel** (red bracket): Points to the $\min_{r \in \mathcal{R}}$ term.
- average over probe dimensions** (grey bracket): Points to the $\frac{1}{|\mathcal{D}|}$ term.
- candidate fingerprint** (blue bracket): Points to the $\bar{K}_{\kappa}(X_d)$ term.
- known fingerprint** (purple bracket): Points to the $\bar{K}_r(X_d)$ term.

- The **reference set** includes standard kernels and kernels discovered earlier in the search

Meta-evaluation

- We seek kernel programs that **transfer to unseen tasks** without repeating discovery
- We adopt a **meta-evaluation** framework inspired by [Goldie et al., 2026]



Evaluation metrics

- We score predictions with the **continuous ranked probability score (CRPS)**:

$$\text{CRPS}(\underbrace{\mathcal{N}(\mu, \sigma^2)}_{\text{GP predictive distribution}}, \underbrace{y}_{\text{observed value}}) = \sigma \left[z(2\Phi(z) - 1) + 2\phi(z) - \frac{1}{\sqrt{\pi}} \right], \quad z = \underbrace{\frac{y - \mu}{\sigma}}_{\text{standardized error}}$$

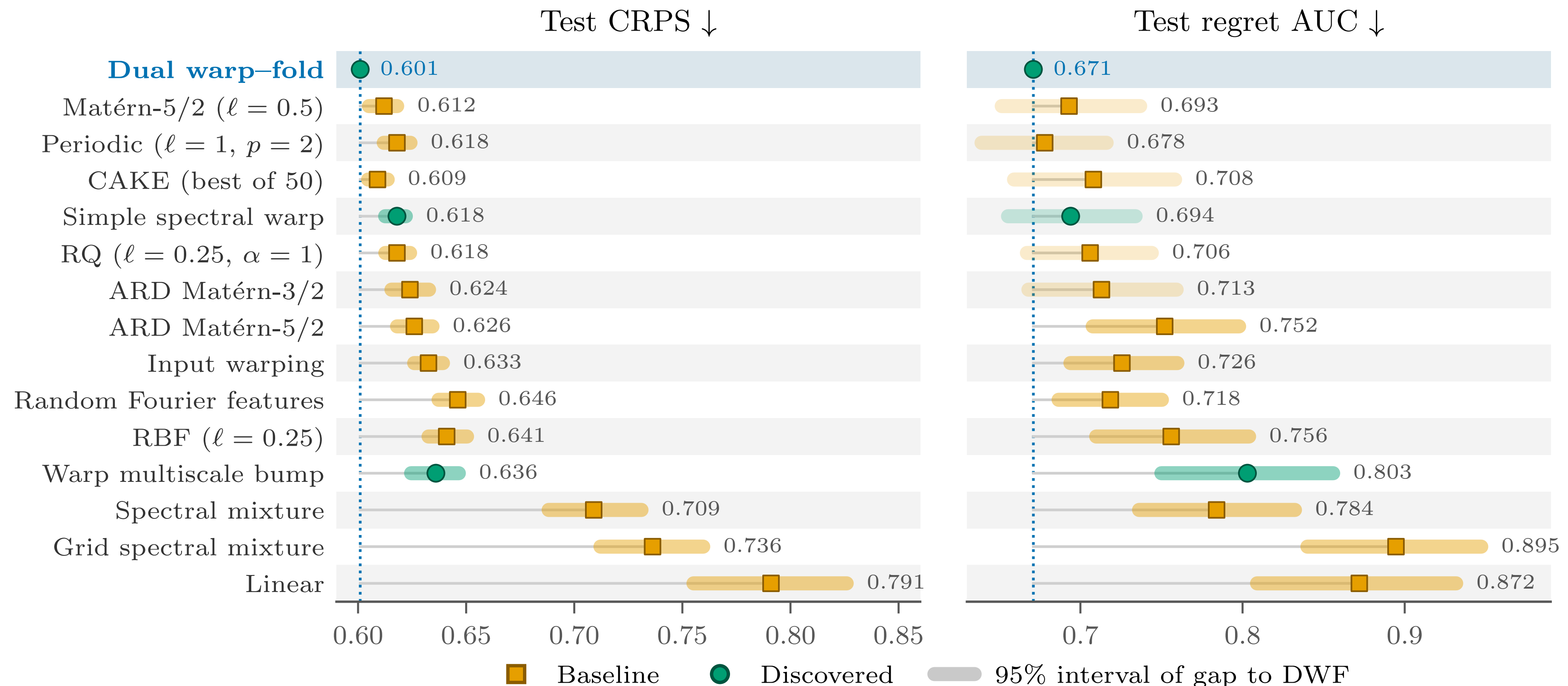
- CRPS evaluates the full predictive distribution, so lower values mean **more accurate and better-calibrated predictions** [Gneiting and Raftery, 2007]
- For black-box optimization, we also report the **regret AUC**:

$$\text{RegretAUC} = \frac{1}{T} \sum_{t=1}^T \frac{f^* - y_t^*}{f^* - y_0^*}$$

average over T steps

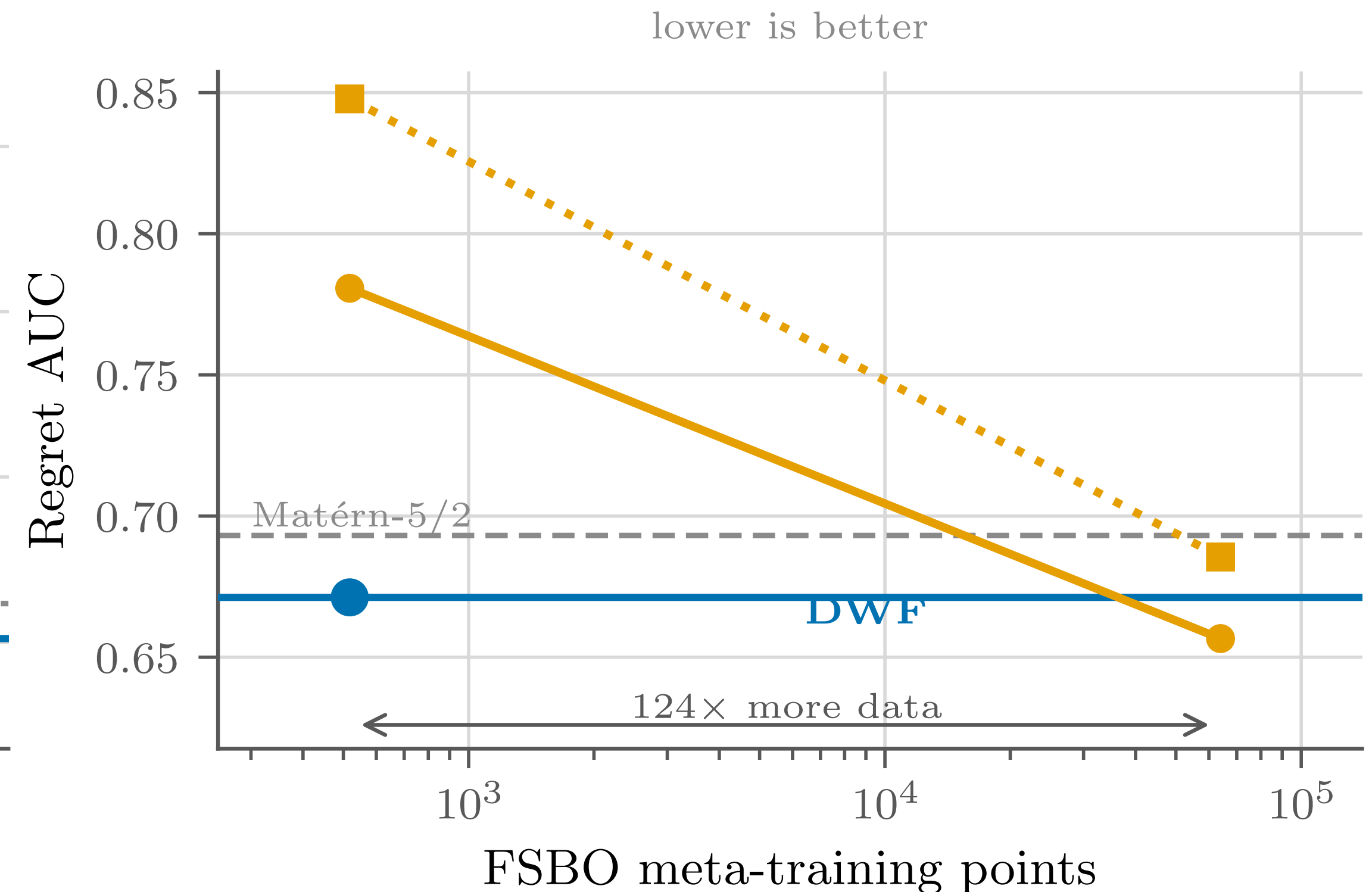
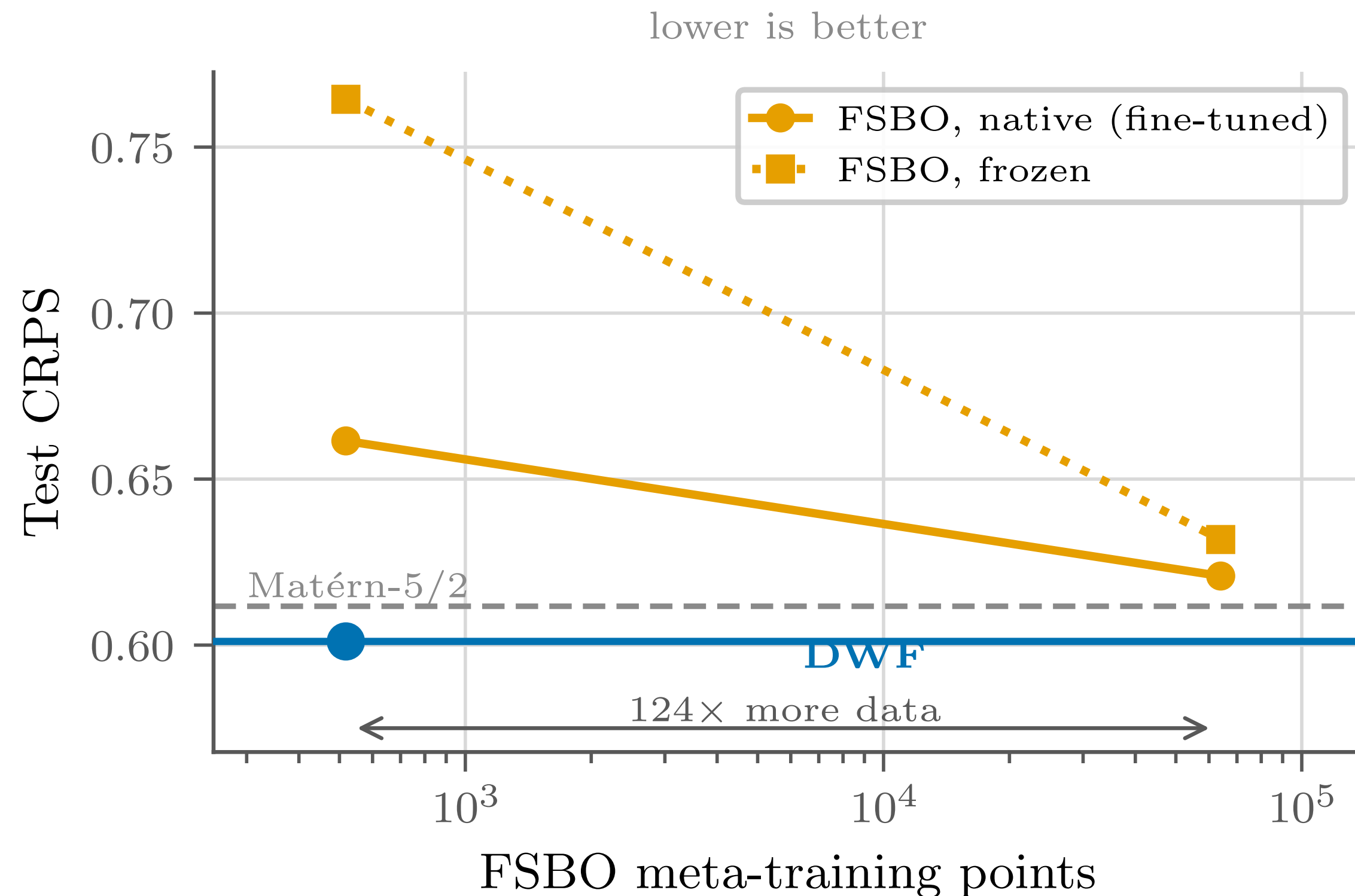
gap to the optimum after step t, relative to the initial gap

Transfer to unseen black-box optimization functions



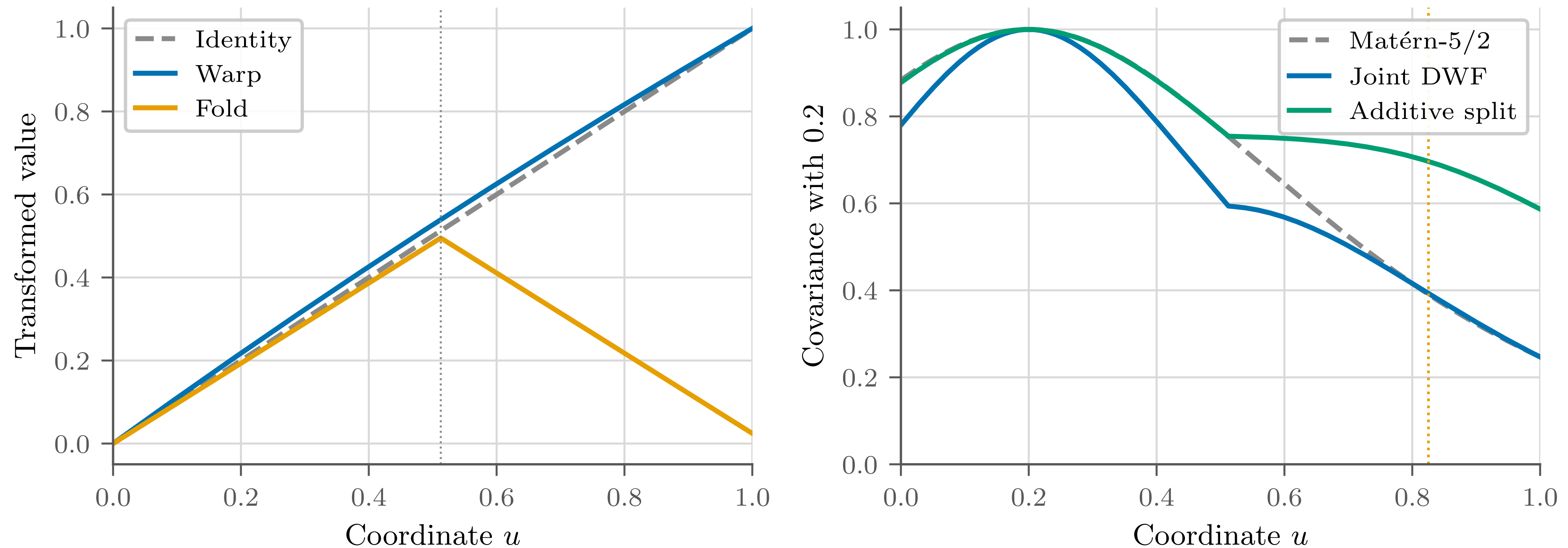
- Test tasks come from six function families withheld from search, with coordinates randomly permuted, reflected, and warped in every episode
- The discovered kernel, **dual warp-fold (DWF)**, achieves the lowest test error and regret among all fixed and learned kernels

Transfer to unseen black-box optimization functions



- With the **same meta-training points**, DWF beats FSBO, a meta-learned deep neural kernel, on both predictive error and regret, while FSBO needs about **124× more data** to approach it
- A short program with the appropriate structure can therefore be substantially **more data-efficient** than a learned neural representation

Discovered kernel: Dual-Warp Fold (DWF)

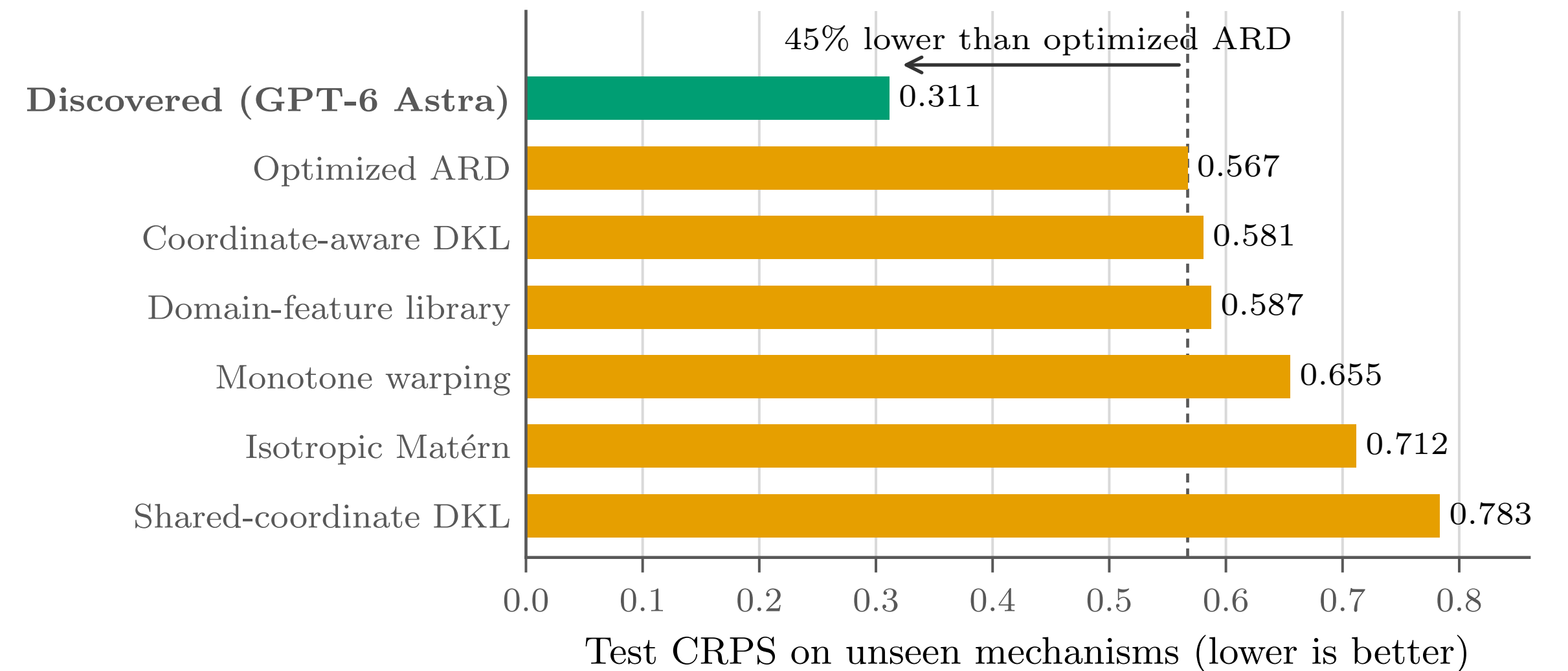


- Combines a gentle warp and a triangular fold before applying a standard Matérn kernel
- Similarity depends on input location as well as distance, a structure that sums and products of stationary kernels **cannot express**
- Since the program is **interpretable**, researchers can analyze and refine it: a **human-refined version** reduces the held-out CRPS by a further 5.7%

Transfer to new enzyme mechanisms

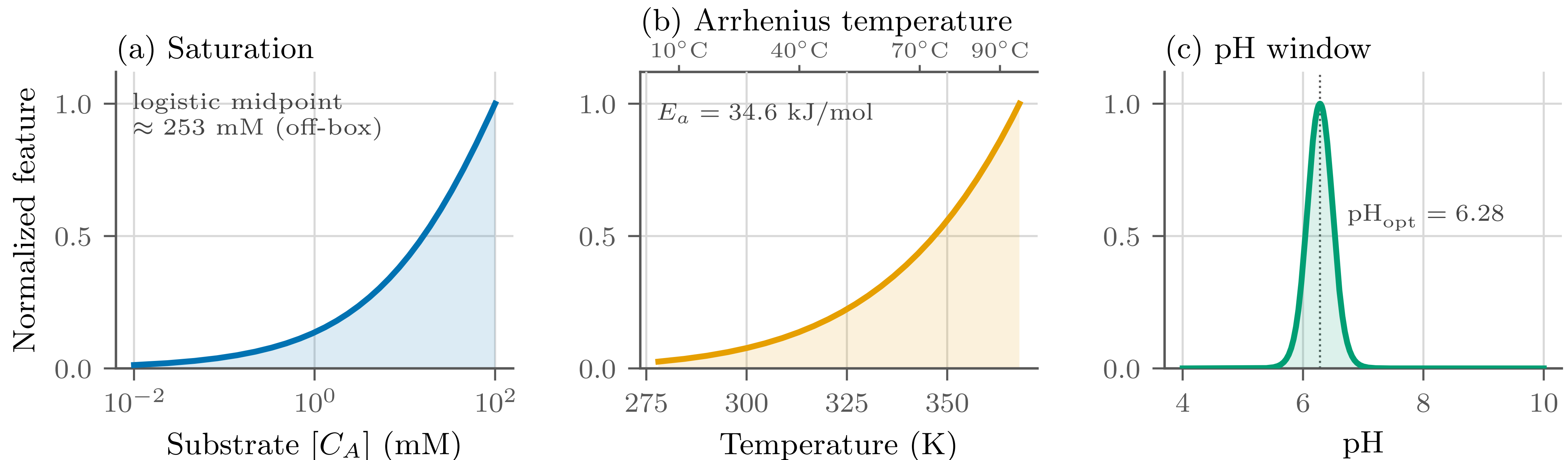
(a) Enzyme kinetics: CRPS reduction vs. Matérn-5/2 (%)

	Ensemble									GPT-6 Astra		
	E1	E2	E3	E4	E5	E6	E7	E8	E9	F1	F2	F3
Ordered bi-bi	17	3	16	10	11	-3	-2	-1	-12	22	38	10
Reversible MM	32	14	17	12	24	6	5	6	-17	43	80	24
Allosteric activation	27	24	18	33	21	-2	-1	2	-8	46	87	18
Fractal kinetics	33	10	19	26	19	0	0	2	-8	44	70	17
Metal activation	13	1	14	6	10	0	0	1	-10	18	27	4
All mechanisms	23	10	16	17	16	0	0	2	-11	33	57	13
	0.62	0.65	0.67	0.63	0.65	0.74	0.74	0.73	0.79	0.49	0.30	0.65
	Validation CRPS											



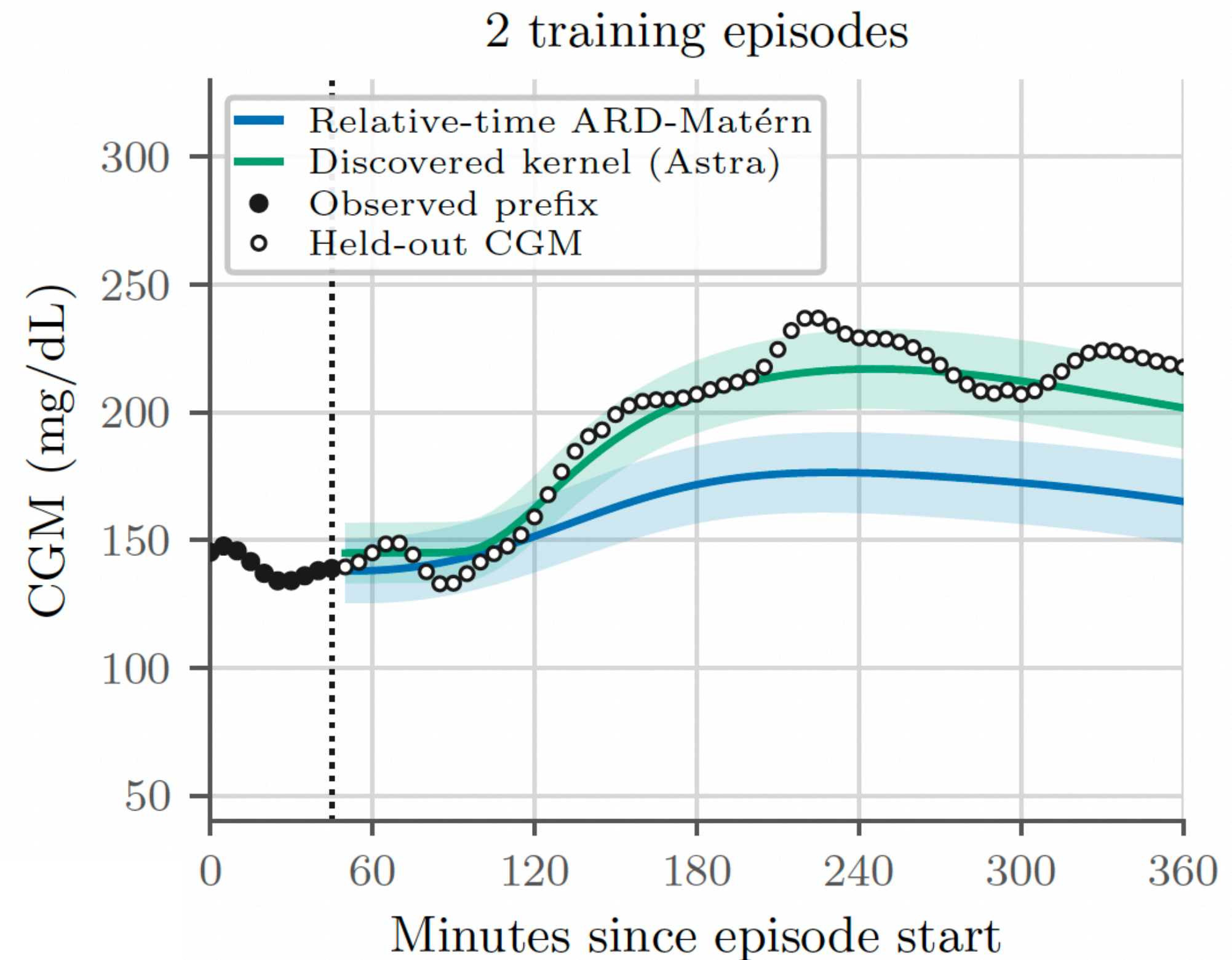
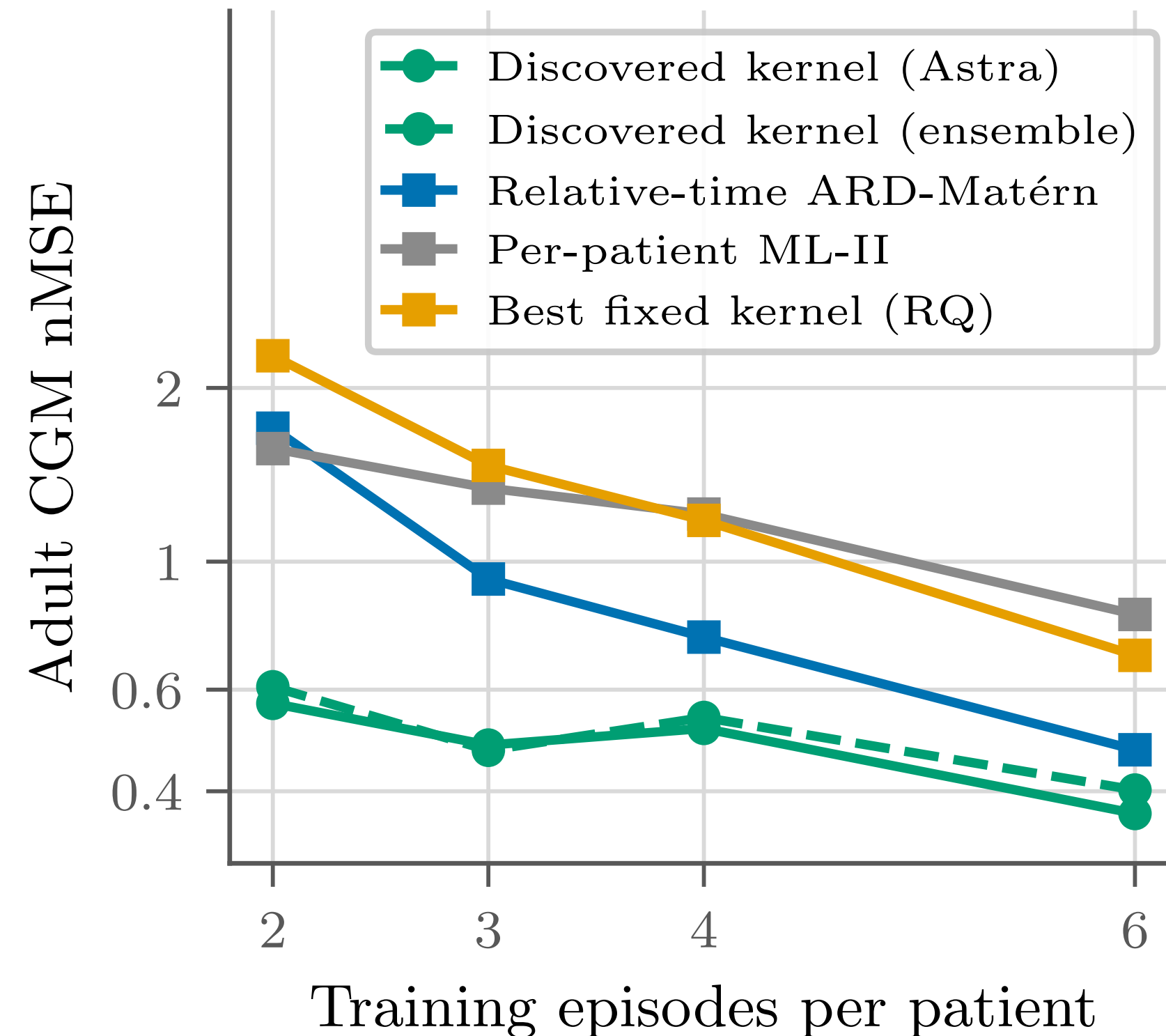
- Kernels discovered on **ten canonical rate laws** predict reaction rates for **five unseen enzyme mechanisms**
- The discovered kernel (GPT-6 Astra) **reduces error by 45%** relative to optimized ARD and beats all six learned baselines, including deep kernel learning (DKL)

Discovered kernel: enzyme-kinetics features



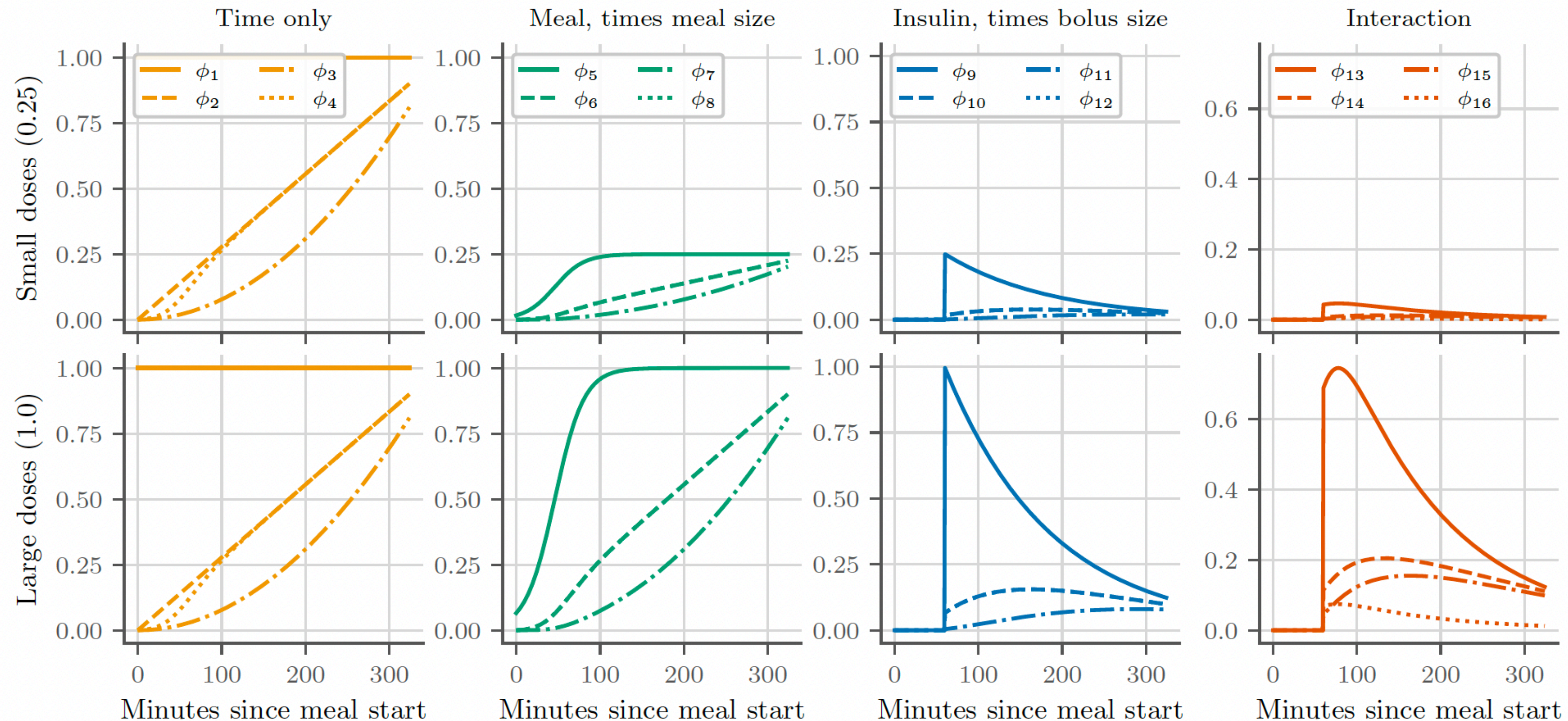
- The discovered kernel builds its features from **textbook enzyme kinetics**, such as substrate saturation, Arrhenius temperature dependence, and a pH window
- It also explains an unseen mechanism as a **combination of the rate-law shapes** seen during search

Transfer to new glucose patients



- Kernels discovered on children and selected on adolescents **forecast blood glucose for unseen adult patients** after a 45-minute observed prefix
- With only two training episodes per patient, the discovered kernels **reduce error by 38–67%** relative to the strongest tuned baseline

Discovered kernel: meal and insulin responses



- The discovered kernel computes the time since the meal and since the insulin bolus, and builds **16 explicit response features** from them
- It encodes a **meal response** that reaches half its effect at 46 minutes and an **insulin effect** that starts at 23 minutes and fades over two hours

Takeaways

- 1 Coding agents can discover **reusable inductive biases** as kernel programs
- 2 We can let the agent propose **kernel components** freely while **construction contracts** guarantee that every accepted kernel is valid
- 3 A **quality-diversity archive** and **novelty screening** can steer the agent toward diverse, functionally new kernels
- 4 We can **amortize the cost of discovery**, because the discovered kernel can be reused on new tasks without any further calls to the agent
- 5 Open-ended search over kernel programs **outperforms hand-designed** kernel grammars and architectures

Ongoing and Future Work

Extend the construction contracts to **new kinds of kernels**

- **Kernels for structured inputs** such as strings and graphs
- **State-space kernels** derived from linear ODEs
- **Semantic kernels** for LLM uncertainty

We welcome **contributions** of new tasks, domains, and construction contracts!



Thank you!



arxiv.org/abs/2610.10394



github.com/richardcsuwandi/kernaut



richardsuwandi@link.cuhk.edu.cn

Scan for the paper

