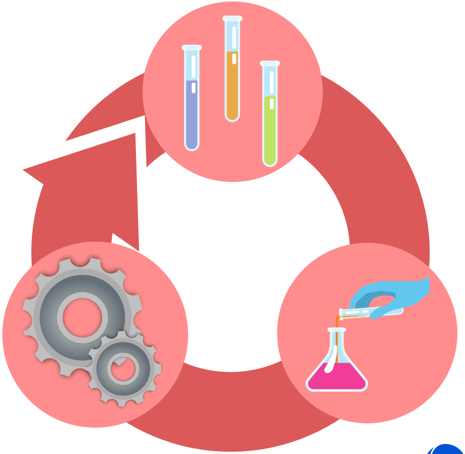
DeepMind

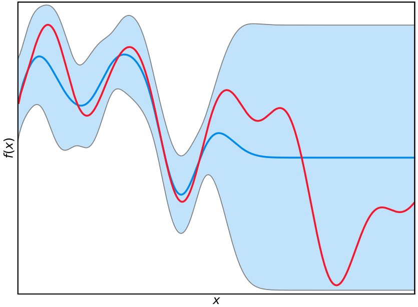
Scaling GP Optimization by Evaluating a Few Unique Candidates Multiple Times

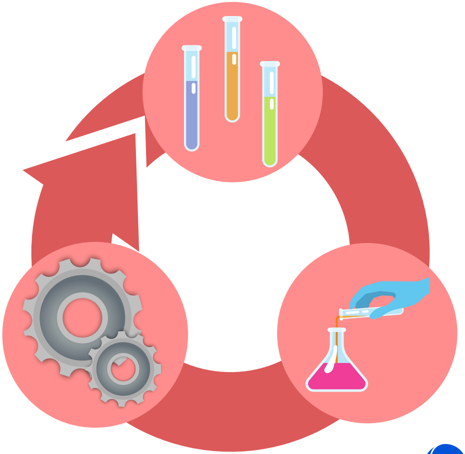
D. Calandriello, L. Carratino, A. Lazaric, M. Valko, L. Rosasco

Running example: optimizing a chemical mixture

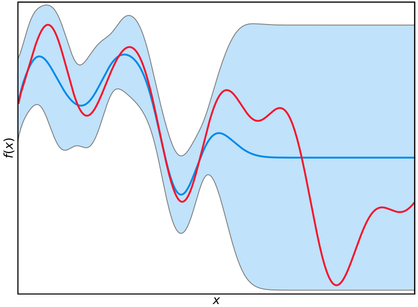


Running example: optimizing a chemical mixture with GP-Opt

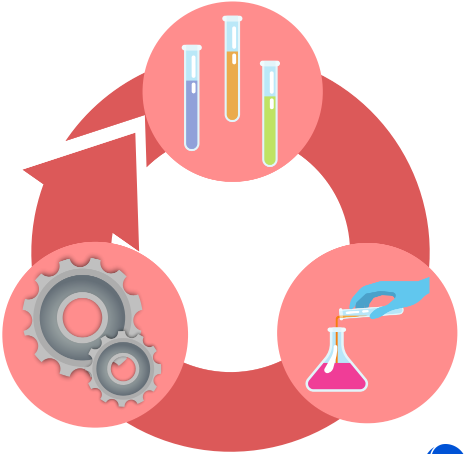




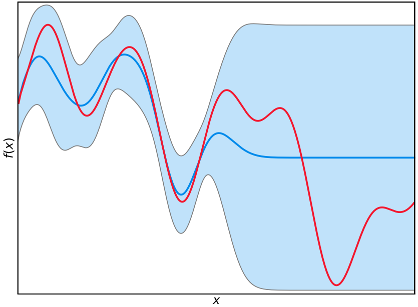
Running example: optimizing a chemical mixture with GP-Opt



GP-Opt can balance exploration/exploitation

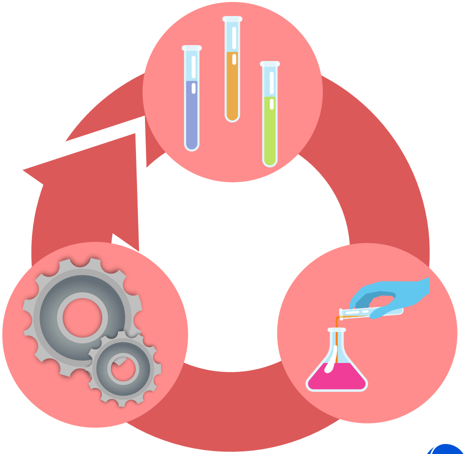


Running example: optimizing a chemical mixture with GP-Opt



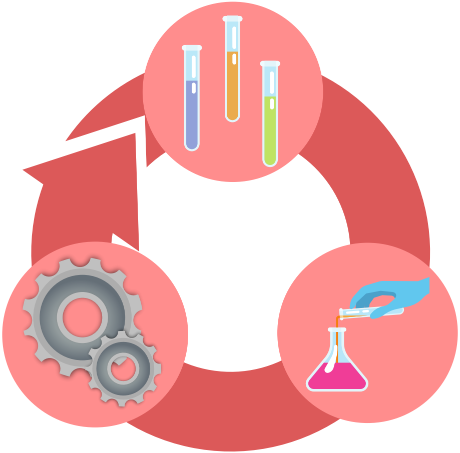
GP-Opt can balance exploration/exploitation but:

- poor experimental scalability (sequential feedback)
- high computational complexity (slow model update)



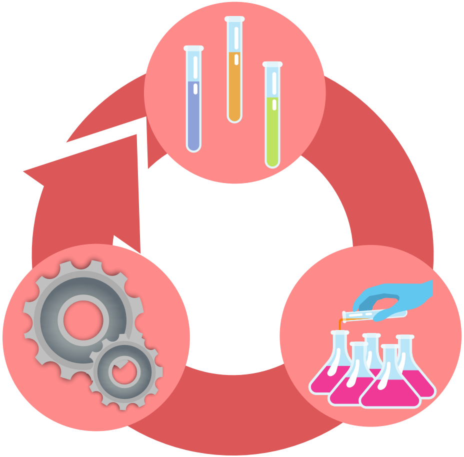
One easy trick to scale

Keep choosing and evaluating the same candidate



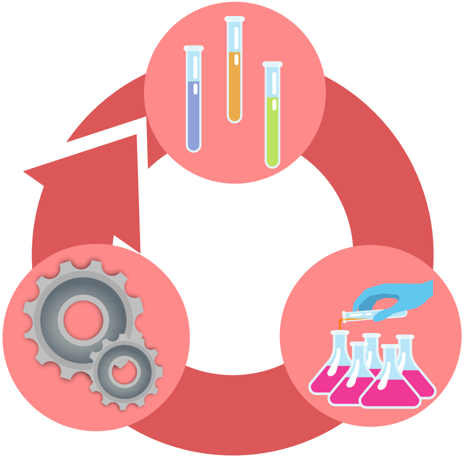
One easy trick to scale

Keep choosing and evaluating the same candidate



One easy trick to scale

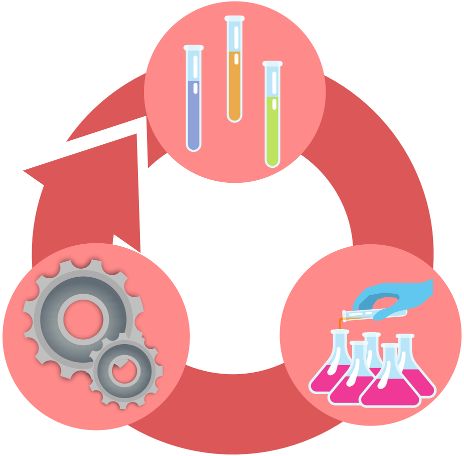
Keep choosing and evaluating the same candidate but **not for too long!**



Too late

One easy trick to scale

Keep choosing and evaluating the same candidate but **not for too long!**

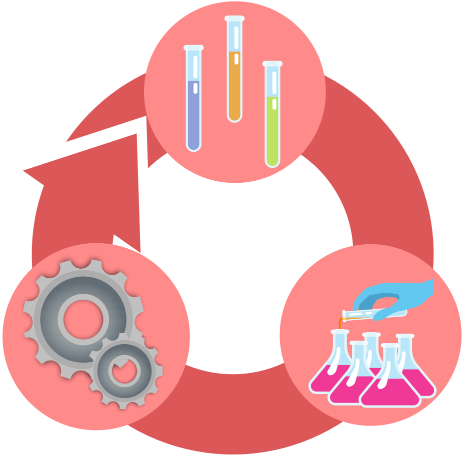


Too early

Too late

One easy trick to scale

Keep choosing and evaluating the same candidate but **not for too long!**



Too early

"Just right"

Too late

Mini-META: making GP-Opt scalable

Contribution: an automatic “just right” detector to choose when to switch candidates

Mini-META: making GP-Opt scalable

Contribution: an automatic "just right" detector to choose when to switch candidates

Strengths:

- Provably preserves convergence rate (not too late)

Mini-META: making GP-Opt scalable

Contribution: an automatic “just right” detector to choose when to switch candidates

Strengths:

- Provably preserves convergence rate (not too late)
- With a provably low number of switches (not too early)
  - Unlocks experimental parallelism
  - Improves computational complexity

Mini-META: making GP-Opt scalable

Contribution: an automatic "just right" detector to choose when to switch candidates

Strengths:

- Provably preserves convergence rate (not too late)
- With a provably low number of switches (not too early)
  - Unlocks experimental parallelism
  - Improves computational complexity
- Easily applicable to popular GP-Opt algorithms (Mini-fied variants)
  - Mini-GP-UCB from GP-UCB [6], Mini-GP-EI from GP-EI [7]

Mini-META: making GP-Opt scalable

Contribution: an automatic "just right" detector to choose when to switch candidates

Strengths:

- Provably preserves convergence rate (not too late)
- With a provably low number of switches (not too early)
  - Unlocks experimental parallelism
  - Improves computational complexity
- Easily applicable to popular GP-Opt algorithms (Mini-fied variants)
  - Mini-GP-UCB from GP-UCB [6], Mini-GP-EI from GP-EI [7]

Weaknesses:

- Cannot be applied when repeated choices are not possible (e.g., personalized news)
- Not very useful for noiseless scenarios

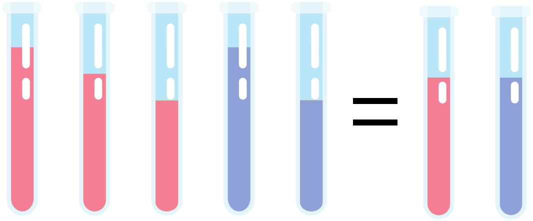
Bonus properties: few unique candidates

Low number of switches + selecting same candidate = few unique candidates in the GP

Bonus properties: few unique candidates

Low number of switches + selecting same candidate = few unique candidates in the GP

Mini-META

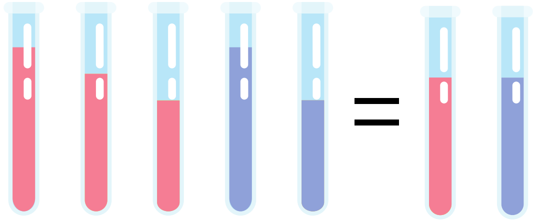


GP reformulation, not GP approximation!

Bonus properties: few unique candidates

Low number of switches + selecting same candidate = few unique candidates in the GP

Mini-META



GP reformulation, not GP approximation!

vs.

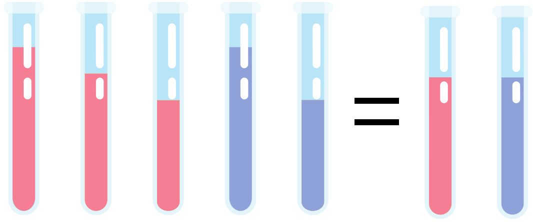
Existing scalable GP-Opt methods [5, 3, 1, 4]

Randomized GP approximation

Bonus properties: few unique candidates

Low number of switches + selecting same candidate = few unique candidates in the GP

Mini-META



vs.

Existing scalable GP-Opt methods [5, 3, 1, 4]

GP reformulation, not GP approximation!

Randomized GP approximation

Special case of stratified sampling [2], inapplicable to GPs with i.i.d. samples

Bonus property: low setup costs

vs.

Wrapping up

We present a simple and practical modification to existing GP-Opt algorithms that drastically improves scalability without sacrificing convergence.

Wrapping up

We present a simple and practical modification to existing GP-Opt algorithms that drastically improves scalability without sacrificing convergence.

See the paper for

- experiments
- rigorous proofs
- more details on continuous domains

Wrapping up

We present a simple and practical modification to existing GP-Opt algorithms that drastically improves scalability without sacrificing convergence.

See the paper for

- experiments
- rigorous proofs
- more details on continuous domains

See us at the poster/virtual session to check if your GP-Opt algorithm can be Mini-fied!

DeepMind

Thanks!

[1] Maximilian Balandat et al. "Botorch: Programmable bayesian optimization in pytorch". In: *arXiv preprint arXiv:1910.06403* (2019).

[2] Mickaël Binois et al. "Replication or exploration? Sequential design for stochastic simulation experiments". In: *Technometrics* 61.1 (2019), pp. 7–23.

[3] Daniele Calandriello et al. "Near-linear Time Gaussian Process Optimization with Adaptive Batching and Resparsification". In: *ICML*. 2020.

[4] Jacob R Gardner et al. "Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration". In: *NeurIPS* (2018).

[5] Mojmir Mutny et al. "Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features". In: *NeurIPS*. 2018.

[6] Niranjan Srinivas et al. "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". In: *ICML*. 2010.

[7] Ziyu Wang et al. "Bayesian multi-scale optimistic optimization". In: *Artificial Intelligence and Statistics*. 2014, pp. 1005–1014.