Precision-Weighted Joint Entropy Search for Bayesian Optimisation

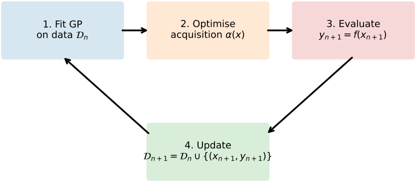
IIT Modelling Seminars

Eduardo César Garrido-Merchán
Universidad Pontificia Comillas – ICADE

April 16, 2026 · 12:30 · Rey Francisco 4, 5th floor

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

Outline



Bayesian Optimisation at a glance

Motivation: a typical energy problem

A typical problem in energy engineering

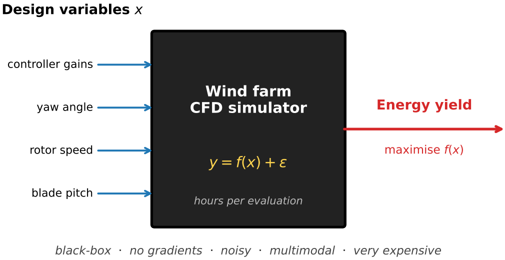
Example. Squeeze the maximum yield out of a wind farm.

Knobs ($x \in \mathbb{R}^d$): blade pitch, rotor speed, yaw, turbine spacing, controller gains.

Objective: maximise annual energy production / minimise LCOE.

Reality check.

- Each evaluation → CFD: **hours of HPC**.
- GA, PSO, CMA-ES → thousands of evals: infeasible.
- Classical optimisation (GD, Newton, SQP) → needs ∇f and no noise: does not apply.

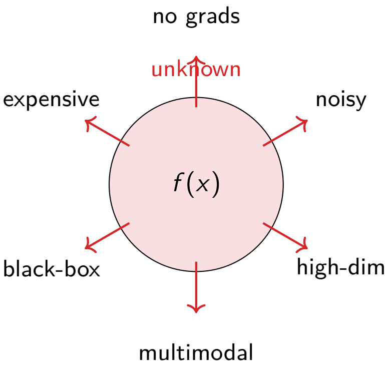


Why can't a human just solve it?

The function is:

- black-box — no analytical expression $f(x)$
- no gradients — CFD does not hand us ∇f
- noisy — stochastic turbulence, seeds
- multimodal — many local optima
- very expensive — each CFD run: hours of CPU
- high-dimensional — $d \gtrsim 10$ in practice

Intuition + grid search do not scale. We need an algorithm that **squeezes every drop of information from each evaluation**.



The problem, formally

Black-box optimisation

$$x^\star = \arg\max_{x \in \mathcal{X} \subset \mathbb{R}^d} f(x), \qquad \text{we observe } y = f(x) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$$

- **No analytical form.** f is accessed only through simulator/experiment queries.
- **No gradients.** First-order methods (GD, Newton, SQP, BFGS) do not apply.
- **Stochastic.** Observations carry noise ε.
- **Very expensive.** Each y costs money, energy, or hours.

Budget. Best x possible in only $n \sim 50$–200 evaluations.

Why not metaheuristics or classical optimisation?

Classical gradient-based optimisation

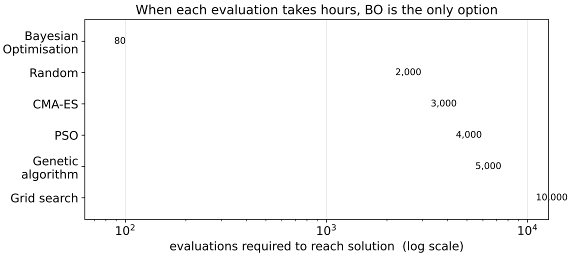
Newton, SQP, L-BFGS…

- Requires ∇f (we do not have it).
- Requires noise-free evaluations.
- Only finds *local* optima.

Metaheuristics

GA, PSO, CMA-ES, simulated annealing…

- Derivative-free ✓
- But memoryless: no probabilistic model of remaining uncertainty.
- Evaluation-hungry: thousands of queries.



With 4 h per evaluation, 5 000 evals ≈ **2 years**.

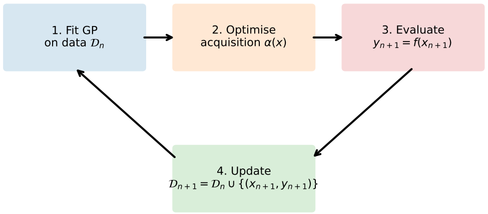
Bayesian Optimisation (BO) is the state of the art when evaluations are the bottleneck.

A crash course on Bayesian Optimisation

The BO recipe, in one diagram

1. Fit a **probabilistic surrogate** (Gaussian process) to the observations collected so far.
2. Build an **acquisition function** $\alpha(x)$ that scores candidates by trading off exploration and exploitation.
3. **Evaluate** f at the maximiser of α.
4. **Update** the dataset; repeat until budget is exhausted.

Two ingredients to design: the **model** (GP) and the **acquisition** (what "information" means).



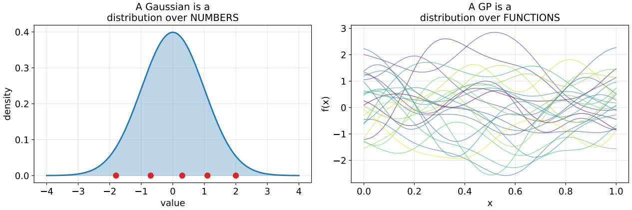
What if we could model the whole problem the Bayesian way?

Bayesian mindset. Put a prior over *the function itself* f, then update with data.

- A **Gaussian distribution** is a prior over *numbers*.
- A **Gaussian Process (GP)** is a prior over *functions*.

Every function drawn from a GP is a plausible guess for f;
the **cloud** of all such functions is our belief about the unknown f.

This is what we visualise on the right.



Step 1 — modelling f with a Gaussian Process

Gaussian process prior

$$f(\cdot) \sim \mathcal{GP}\big(\, m(x),\ k(x, x') \,\big)$$

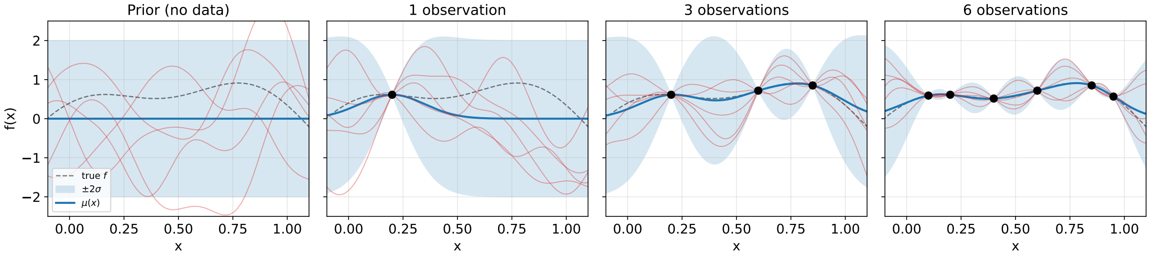
- mean function $m(x)$ — usually zero
- kernel $k(x, x')$ — encodes smoothness, length-scale, amplitude

Conditioning on data $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$ gives a Gaussian posterior at every x:

$$p(f(x) \mid \mathcal{D}_n) \;=\; \mathcal{N}\big(\, \mu_n(x),\ \sigma_n^2(x) \,\big)$$

Notes. $\mu_n(x)$ = best guess for $f(x)$. $\sigma_n^2(x)$ = uncertainty: small near data, large in unexplored regions.

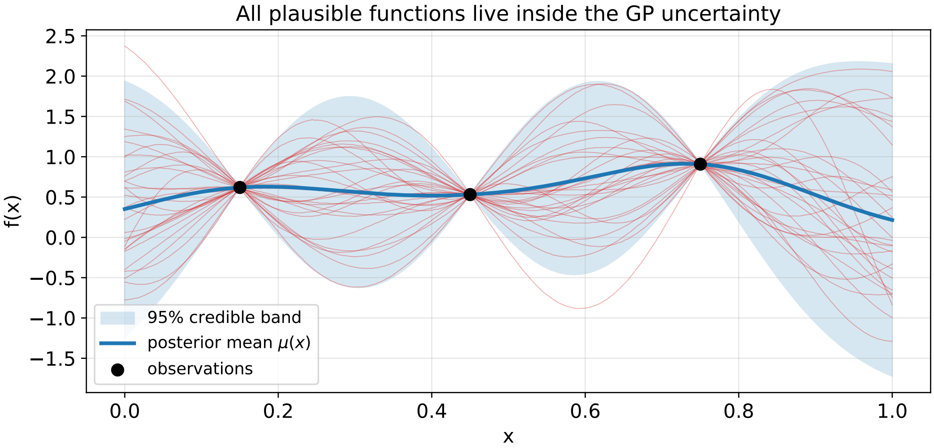
GPs learn as data comes in



Blue line = posterior mean $\mu_n(x)$ Blue band = $\pm 2\sigma_n(x)$ credible region

Red curves = sample functions drawn from the posterior

All plausible f live inside the uncertainty band



A posterior sample is one plausible realisation of f consistent with the data.

Step 2 — acquisition functions: where to look next?

Rule. Given the GP posterior, pick $x_{n+1} = \arg\max_x \alpha(x)$.

Two classic choices:

Upper Confidence Bound (UCB)

$$\alpha_{\mathrm{UCB}}(x) = \mu_n(x) + \kappa\,\sigma_n(x)$$

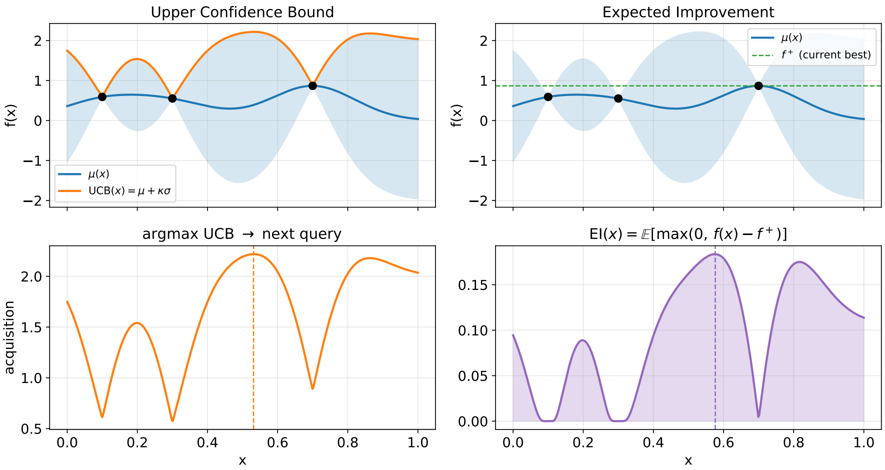
exploit the mean + explore where variance is high. Trade-off controlled by $\kappa > 0$.

Expected Improvement (EI)

$$\alpha_{\mathrm{EI}}(x) = \mathbb{E}_{f(x)\sim p(\cdot|\mathcal{D}_n)}\left[\max(0,\ f(x) - f^+)\right]$$

Expected *improvement* over the current best f^+. Closed-form for a Gaussian posterior.

UCB and EI in pictures



Top: GP posterior. Bottom: acquisition; dashed = argmax → next evaluation.

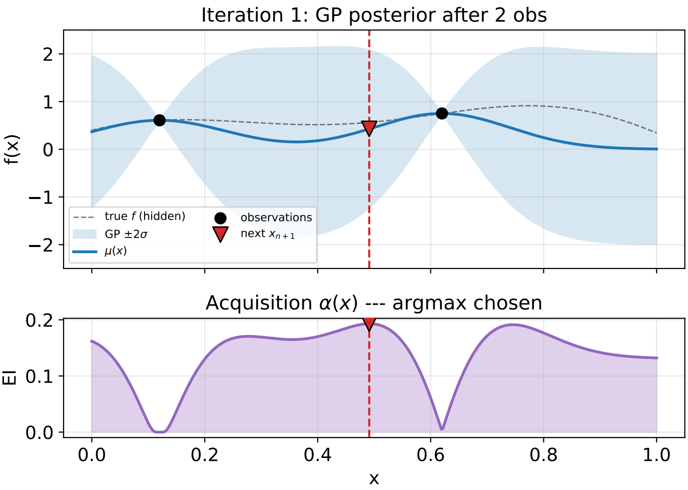
BO in action — iteration 1

Algorithm step:

1. Fit GP to current data $\mathcal{D}_n$.
2. Compute acquisition $\alpha(x)$ from the GP.
3. Pick $x_{n+1} = \arg\max_x \alpha(x)$.
4. Evaluate $y_{n+1} = f(x_{n+1})$; append to $\mathcal{D}$.

Now: $n = 2$ observations.
GP very uncertain in the middle.

Acquisition peaks where mean is high and variance moderate — we query there.



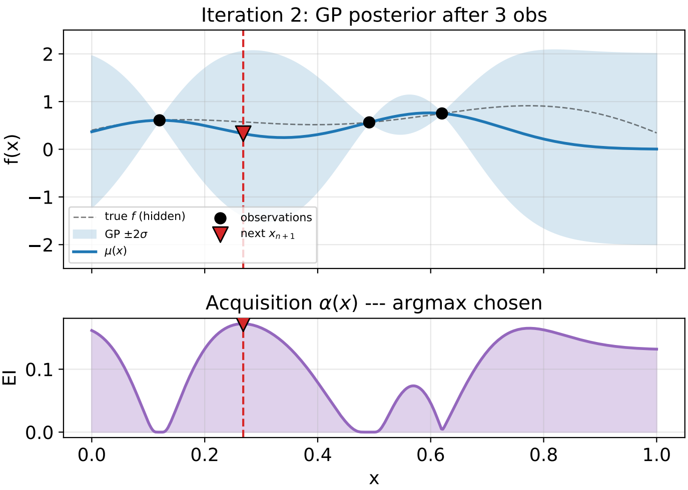
BO in action — iteration 2

Algorithm step:

1. Fit GP to current data $\mathcal{D}_n$.
2. Compute acquisition $\alpha(x)$ from the GP.
3. Pick $x_{n+1} = \arg\max_x \alpha(x)$.
4. Evaluate $y_{n+1} = f(x_{n+1})$; append to $\mathcal{D}$.

Now: Added 1 observation. Uncertainty shrinks locally.

GP now trusts the central region; acquisition re-points elsewhere.



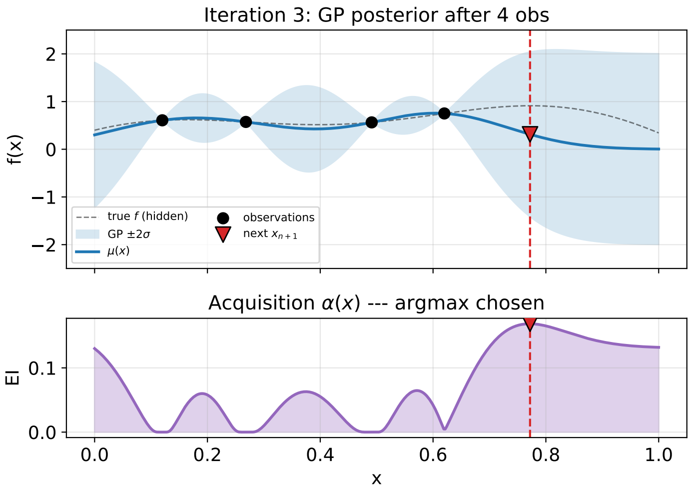
BO in action — iteration 3

Algorithm step:

1. Fit GP to current data $\mathcal{D}_n$.
2. Compute acquisition $\alpha(x)$ from the GP.
3. Pick $x_{n+1} = \arg\max_x \alpha(x)$.
4. Evaluate $y_{n+1} = f(x_{n+1})$; append to $\mathcal{D}$.

Now: Mean updated.
Acquisition balances exploration (edges) vs. exploitation (peak).

The dashed red line marks the next evaluation: where uncertainty still promises improvement.



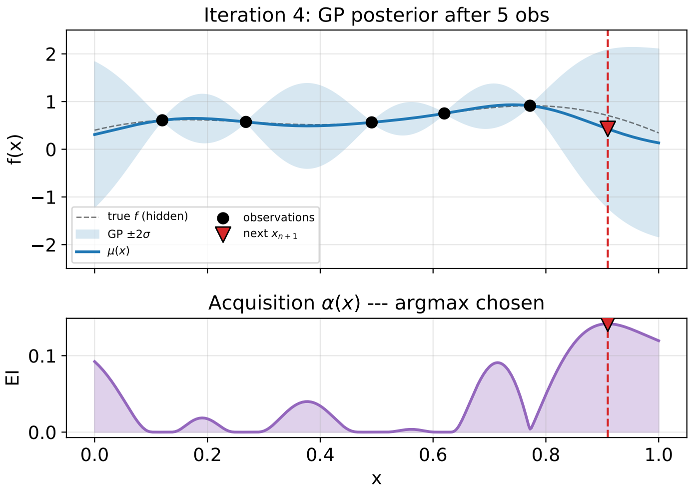
BO in action — iteration 4

Algorithm step:

1. Fit GP to current data $\mathcal{D}_n$.
2. Compute acquisition $\alpha(x)$ from the GP.
3. Pick $x_{n+1} = \arg\max_x \alpha(x)$.
4. Evaluate $y_{n+1} = f(x_{n+1})$; append to $\mathcal{D}$.

Now: GP now tightly constrained near the true maximum.

Acquisition collapses toward a single peak — we are close to convergence.



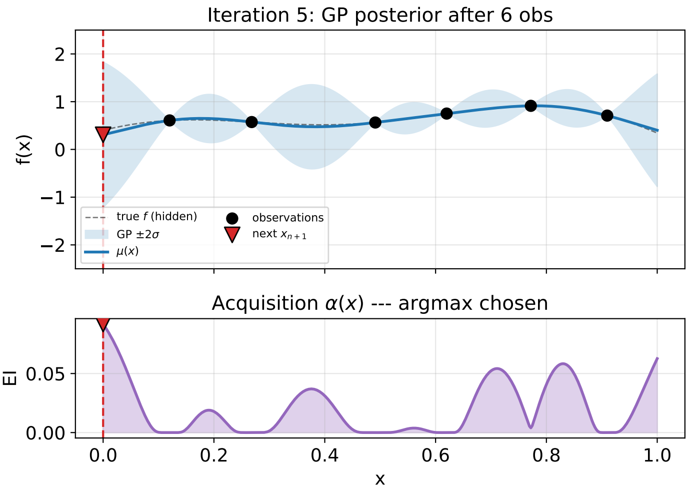
BO in action — iteration 5

Algorithm step:

1. Fit GP to current data $\mathcal{D}_n$.
2. Compute acquisition $\alpha(x)$ from the GP.
3. Pick $x_{n+1} = \arg\max_x \alpha(x)$.
4. Evaluate $y_{n+1} = f(x_{n+1})$; append to $\mathcal{D}$.

Now: After 5 BO iterations.

GP matches the true f near the optimum.
EI has almost flattened $\rightarrow$ we can stop.



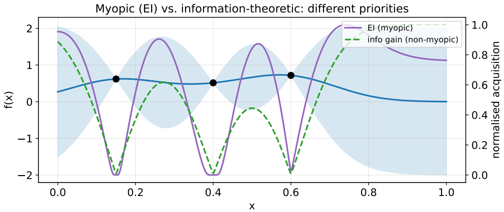
A limitation: UCB and EI are myopic

They reason only about **the function value at the next query**:

$$\text{"will } f(x_{n+1}) \text{ improve on } f^+\text{?"}$$

They do not reason about how much each query *teaches us about the optimum* — which is the real quantity we care about.

Next step. Upgrade from "value at next point" to "information about $x^\star$".



The optimum as a random variable

The key conceptual jump

Idea. Since f is random (a GP), so is its argmax:

$$x^\star = \arg\max_{x \in \mathcal{X}} f(x) \qquad \text{is a random variable.}$$

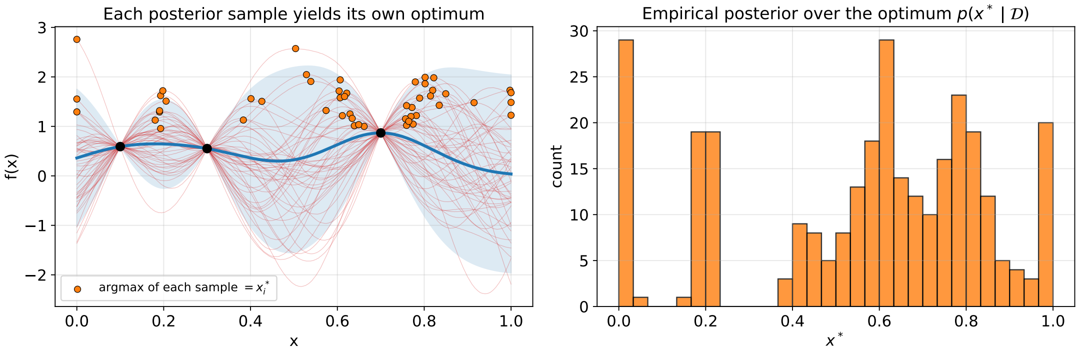
Sampling $x^\star$ from the GP

1. Draw a function $f^{(i)} \sim p(f \mid \mathcal{D}_n)$
2. Optimise it: $x_i^\star = \arg\max_x f^{(i)}(x), \; f_i^\star = f^{(i)}(x_i^\star)$

$\{(x_i^\star, f_i^\star)\}_{i=1}^N$ are samples from the posterior over the optimum.

This is the modelling trick that unlocks information-theoretic acquisition functions.

The posterior over the optimum, visualised



Left: each sampled function $f^{(i)}$ has its own maximiser (orange dots).
Right: histogram over maximisers $\rightarrow$ the empirical posterior $p(x^\star \mid \mathcal{D}_n)$.

The right objective: information gain on $x^\star$

Uncertainty on $x^\star$ is measured by its entropy:

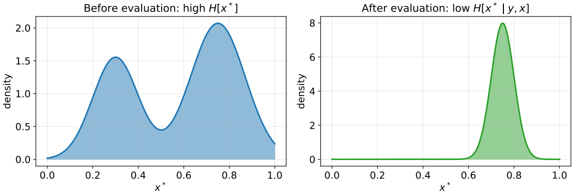
$$\mathrm{H}[\,p(x^\star \mid \mathcal{D}_n)\,] \;=\; -\int p(x^\star) \log p(x^\star)\, dx^\star$$

What should we query? Pick the point that is expected to reduce that entropy the most:

$$\alpha_{\mathrm{ES}}(x) = \mathrm{H}[x^\star] - \mathbb{E}_y\big[\mathrm{H}[x^\star \mid x, y]\big]$$

$$= \mathrm{I}(y; x^\star \mid x, \mathcal{D}_n)$$

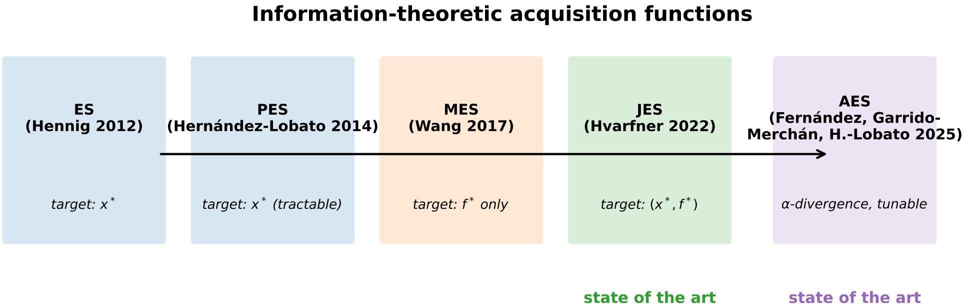
Mutual information between the next observation and the unknown optimum.



Pick x that maximises $H[x^*] - \mathbb{E}_y[H[x^* \mid y, x]]$ (information gain)

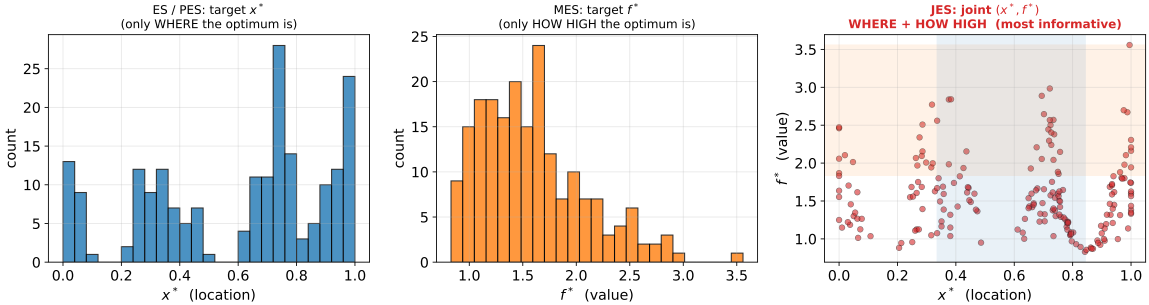
From ES to Joint Entropy Search (JES)

A family of information-theoretic acquisitions



- **ES** (Hennig & Schuler, 2012) — target $x^\star$; intractable.
- **PES** (Hernández-Lobato *et al.*, 2014) — tractable via expectation propagation.
- **MES** (Wang & Jegelka, 2017) — target only $f^\star$; cheap but loses info.
- **JES** (Hvarfner 2022, Tu 2022) — joint $(x^\star, f^\star)$; **state of the art**.
- **AES** (Fernández-Sánchez, Garrido-Merchán & Hernández-Lobato, 2025) — α-divergence generalisation; **also state of the art**.

Why target the joint $(x^\star, f^\star)$?



Knowing *both* where and how high the optimum is leaves no residual ambiguity.
JES uses strictly more information per query than PES (only $x^\star$) or MES (only $f^\star$).

Joint Entropy Search, in equations

JES acquisition

$$\alpha_{\text{JES}}(x) = \mathrm{I}\big(y; (x^\star, f^\star) \,\big|\, x, \mathcal{D}\big) = \mathrm{H}[p(y \mid x, \mathcal{D})] - \mathbb{E}_{(x^\star, f^\star)}\big[\mathrm{H}[p(y \mid x, \mathcal{D}, x^\star, f^\star)]\big]$$

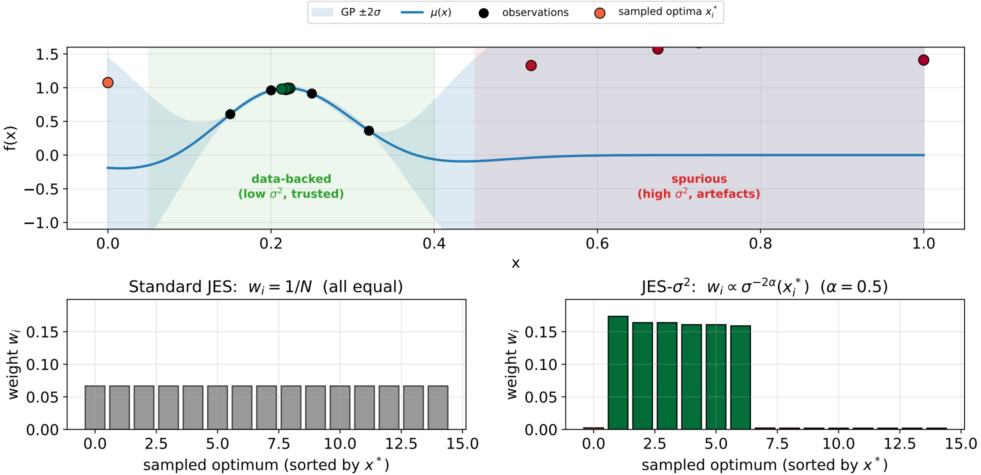
- First term: predictive entropy of y — **analytic** (Gaussian).
- Second term: conditional entropy given the optimum — not analytic!

Fix: Monte Carlo over sampled optima. Draw N optima $\{(x_i^\star, f_i^\star)\}$ and average:

$$\alpha_{\text{JES}}(x) \approx \mathrm{H}[p(y \mid x, \mathcal{D})] - \sum_{i=1}^{N} \frac{1}{N} \mathrm{H}[p(y \mid x, \mathcal{D}, x_i^\star, f_i^\star)]$$

Every sampled optimum contributes $1/N$ — all treated as equally credible.

Why uniform weighting is suspicious — visually



Top: optima coloured by σ^2 (data-backed / spurious). **Bottom:** JES uniform vs. JES-σ^2.

Our method: JES-σ^2

JES-σ^2 — precision-weighted JES

Core idea. Weight each sampled optimum by the model's confidence at that location.

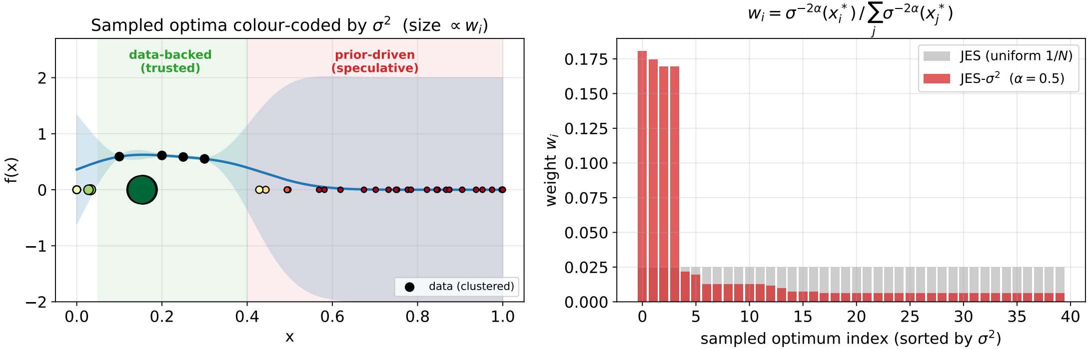
JES-σ^2 acquisition

$$\alpha_{\text{JES-}\sigma^2}(x) = \mathrm{H}[p(y \mid x, \mathcal{D})] - \sum_{i=1}^{N} w_i \cdot \mathrm{H}[p(y \mid x, \mathcal{D}, x_i^\star, f_i^\star)]$$

$$w_i = \frac{\sigma^{-2\alpha}(x_i^\star)}{\sum_{j=1}^{N} \sigma^{-2\alpha}(x_j^\star)}, \qquad \alpha \in [0, 1]$$

- $\alpha = 0 \Rightarrow$ recovers standard JES (uniform weights)
- $\alpha = 0.5 \Rightarrow$ inverse standard deviation (our default)
- $\alpha = 1 \Rightarrow$ aggressive inverse-variance weighting

What the weights do, visually



Left: sampled optima coloured by σ^2, size proportional to weight.
Right: weights under standard JES (grey) vs. JES-σ^2 (red).

Three complementary interpretations

1. Importance sampling

JES samples from a proposal q that over-represents high-σ^2 areas.

The correction is $p/q \propto \sigma^{-2\alpha}$.

Standard JES ignores it.

2. Precision weighting

Classical statistics: combine Gaussian estimates weighting by inverse variance.

$$w_i \propto \frac{1}{\sigma_i^2}$$

Same principle here.

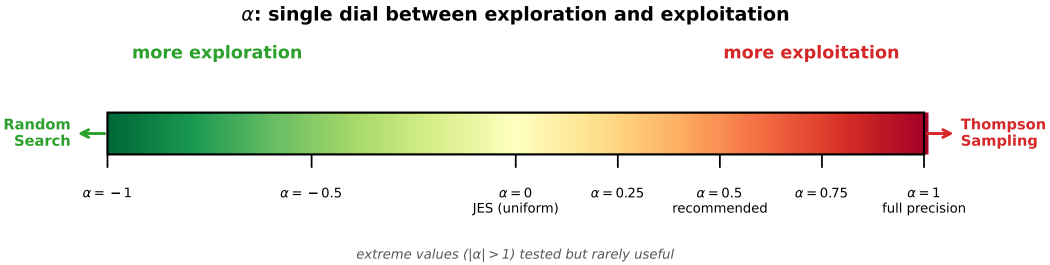
3. Exploration dial

Down-weighting high-σ^2 optima shifts toward exploitation.

α: one knob between exploration ($\alpha \to 0$) and exploitation ($\alpha \to 1$).

One line of code. $O(N)$ overhead. Zero risk at $\alpha = 0$.

α as a slider between exploration and exploitation



- The hyperparameter α gives practitioners fine-grained control over how strongly the model trusts the variance signal.
- No need for different acquisitions in exploration-heavy vs. exploitation-heavy phases; one dial.

The algorithm in full

JES-σ^2 acquisition evaluation

1. Draw N GP posterior samples; optimise each $\rightarrow \{(x_i^\star, f_i^\star)\}$
2. Query $\sigma_n^2(x_i^\star)$ from the GP (already available)
3. Compute $w_i \leftarrow (\sigma_n^2(x_i^\star) + \varepsilon)^{-\alpha} / \sum_j (\sigma_n^2(x_j^\star) + \varepsilon)^{-\alpha}$
4. $H_0 \leftarrow \mathrm{H}[p(y \mid x, \mathcal{D})]$ (closed-form Gaussian)
5. For $i = 1, \ldots, N$: $H_i \leftarrow \mathrm{H}[p(y \mid x, \mathcal{D}, x_i^\star, f_i^\star)]$ (truncated Gaussian)
6. Return $H_0 - \sum_i w_i H_i$

$\varepsilon = 10^{-6}$ clamps numerical stability.

Experimental evaluation

What we measure

Three complementary metrics:

1. **Performance.** Simple regret: $f(x^\star) - \max_i f(x_i)$.
2. **Exploitation.** Average query distance to the best-known optimum.
3. **Exploration.** Spread of queries across $\mathcal{X}$.

Claim. α behaves as a smooth slider across the exploration–exploitation spectrum, not only as a performance booster.

Three benchmarks: toy 1D curve, 2D synthetic two-peak trap, standard benchmarks (Branin, Hartmann, Griewank).

Toy 1D function — visual sanity check

JES-σ²: Exploración vs Explotación

Concave Rotada Anisotrópica 6D — Efecto de α

12 de febrero de 2026

```
Configuración experimental

  Dimensión:          6D
  Dominio:            [0, 1]⁶
  Función:            f(x) = −(x−c)ᵀ Rᵀ Λ R (x−c)
  Centro (c):         [0.65, 0.35, 0.55, 0.45, 0.70, 0.30]
  Autovalores (Λ):    diag(1, 3, 10, 30, 60, 100)
  Número condición:   100
  Rotación (R):       Matriz ortogonal (QR, seed=98765)
  Valor óptimo:       f* = 0 en x* = c

  Seeds: 5   |   Iter. BO: 50   |   n_init: 7 (Sobol)
  Thompson samples: 8   |   Sobol cand.: 512
  GP: SingleTaskGP + Standardize(m=1)

Métodos comparados

  1. JES-σ² (α=-0.5) — Peso proporcional a varianza (EXPLORACIÓN)
     w_i = (σ²(x*_i))^0.5  →  más peso a optima inciertos

  2. JES (α=0)        — Pesos uniformes (BASELINE)
     w_i = 1            →  JES estándar

  3. JES-σ² (α=0.5)  — Peso inversamente proporcional a varianza
     w_i = 1/√σ²(x*_i) →  más peso a optima fiables (EXPLOTACIÓN)

Métricas

  • Exploración (OTSD): Observation Traveling Salesman Distance
    Papenmeier et al. (2025), UAI 2025
    Longitud del tour TSP sobre los puntos seleccionados.
    Mayor OTSD = puntos más dispersos = más exploración.

  • Explotación: Σ_{t=1}^{T} μ(x_t) − f*
    Suma de la media GP en cada punto menos el óptimo.
    Más cercano a 0 = selecciona puntos cerca del óptimo.

Hipótesis

  α negativo invierte el mecanismo de JES-σ²: da más peso a
  optima muestreados en zonas de alta incertidumbre, forzando
  al optimizador a explorar el espacio. Esto debería producir
  una gradación monótona en exploración/explotación con α.
```

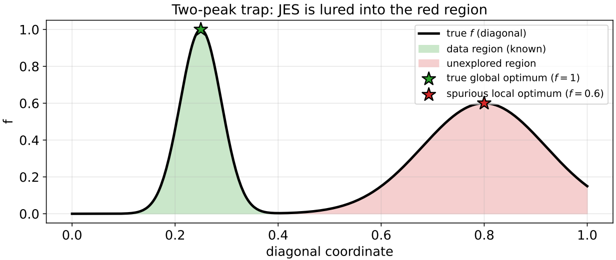
$\alpha = 0.5$ converges faster and more consistently than plain JES.

Synthetic 2D: the two-peak trap

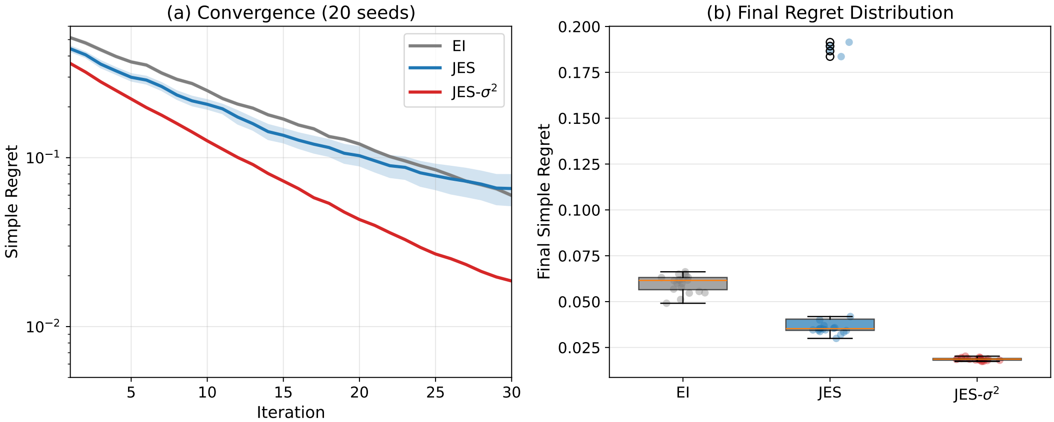
Designed to expose JES's failure mode:

- narrow global optimum at $(0.25, 0.25)$, $f = 1.0$
- broader local optimum at $(0.80, 0.80)$, $f = 0.6$
- initial data near the global peak $\rightarrow$ spurious region unexplored

Result.
JES $\rightarrow$ regret 0.133
JES-σ^2 $(\alpha = 0.5)$ $\rightarrow$ regret **0.032**
-76% regret.



Convergence on the two-peak function



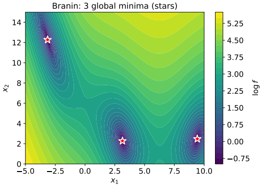
JES occasionally gets "stuck" exploring the spurious peak; JES-σ^2 resists it.

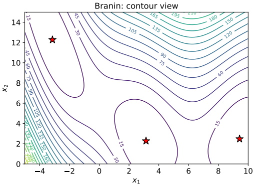
Branin — the landscape

Branin: standard 2D benchmark with **three global minima**.

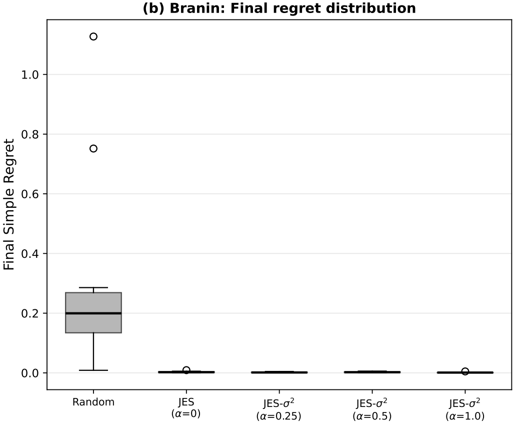
- $\mathcal{X} = [-5, 10] \times [0, 15]$
- Three equivalent global optima (red stars).
- Ridged, curved structure.

A fair test for BO: no pathology, just a real surface.





Branin — final regret distribution



Final regret across 10 seeds. All JES-σ^2 variants beat standard JES with tighter distributions.

Branin — numbers

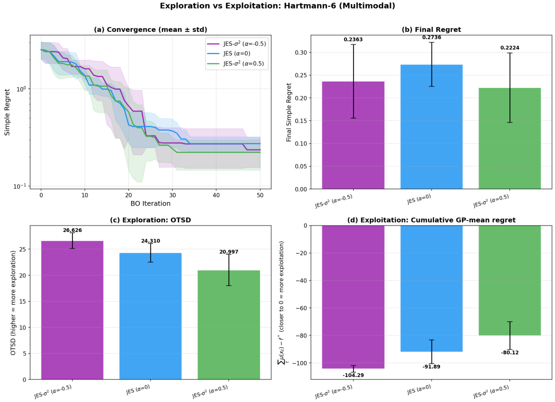
Method	Mean regret	vs. JES
Random search	0.3117	—
JES ($\alpha = 0$)	0.00296	—
JES-σ^2 ($\alpha = 0.25$)	0.00159	-46.1%
JES-σ^2 ($\alpha = 0.50$)	0.00244	-17.6%
JES-σ^2 ($\alpha = 1.00$)	**0.00117**	$-$**60.3**%

10 seeds, 100 iterations, 5 Sobol initial points.

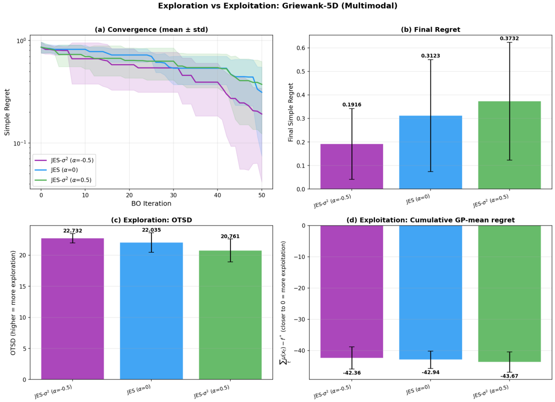
Best $\alpha = 1$ on Branin: well-calibrated GP $\rightarrow$ aggressive precision weighting pays.

Standard benchmarks — Hartmann & Griewank

Hartmann-6

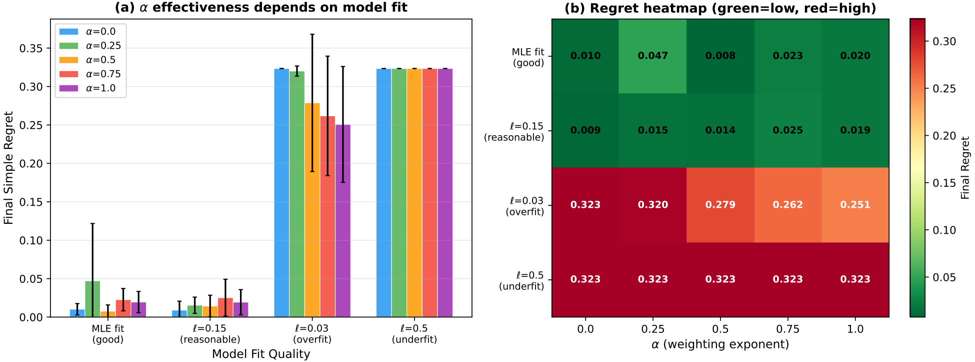


Griewank



Consistent improvement over JES, with tighter distributions across seeds.

When does α help? The calibration requirement



α helps when the GP fits well. A **mis-calibrated model makes the variance signal meaningless**: one cannot exploit structure the model has not learned.

Take-aways from the experiments

- **Synthetic trap:** -76% regret when spurious optima exist.
- **Branin:** -60% regret on a well-behaved benchmark.
- **Hartmann / Griewank:** consistent improvement, tighter seed-to-seed variance.
- **Exploration/exploitation metrics:** α really is a smooth slider.
- **Model calibration matters**: α is useless (but harmless) if the GP is broken.

Practical recommendation. Start with $\alpha = 0.5$ and MLE hyperparameter fitting.

Conclusions & future work

Conclusions

What JES-σ^2 is

- A one-line change to JES that weights sampled optima by GP confidence.
- Motivated by *importance sampling*, *precision weighting*, and *exploration–exploitation control*.
- Recovers standard JES at $\alpha = 0$ — **strictly generalises it**.
- Practically free: $O(N)$ extra cost.

What it gives you

- Up to 76% **regret reduction** on the two-peak trap, 60% on Branin.
- **Tighter convergence** across random seeds.
- A **single tunable dial** for practitioners.

Future work

- **Adaptive** α: multi-signal controller (budget, GP-HP stability, SNR, optima concentration).
- **Formal regret bounds** — currently empirical evidence only.
- **High-dimensional BO**: $d \gtrsim 20$; benefits should grow where the prior dominates.
- **Multi-objective JES-**σ^2: weights factorise over objectives, $w_i \propto \prod_m \sigma_m^{-2\alpha}$.
- Apply precision-weighting principle to **MES, PES, AES**, and beyond.
- Real-world energy deployments: wind farm layouts, storage dispatch, building HVAC.

References

References I

Our related work

D. Fernández-Sánchez, **E. C. Garrido-Merchán**, D. Hernández-Lobato,
Alpha Entropy Search for new information-based Bayesian optimization,
Knowledge-Based Systems, vol. 322, 113612, 2025.
arXiv:2411.16586

E. C. Garrido-Merchán, D. Hernández-Lobato,
Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints (PESMOC),
Neurocomputing, 361: 50–68, 2019.
arXiv:1609.01051

E. C. Garrido-Merchán, D. Hernández-Lobato,
Parallel Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints applied to the tuning of machine learning algorithms,
Expert Systems with Applications, 2022.
arXiv:2004.00601

D. Fernández-Sánchez, **E. C. Garrido-Merchán**, D. Hernández-Lobato,
Improved Max-value Entropy Search for Multi-objective Bayesian Optimization with Constraints (MESMOC+),
Neurocomputing, 2023.
arXiv:2011.01150

References II

E. C. Garrido-Merchán,
Information-theoretic Bayesian Optimization: Survey and Tutorial, 2025.
arXiv:2502.06789

References III

Foundational and related literature

- B. Shahriari *et al.*, *Taking the human out of the loop: a review of Bayesian optimization*, Proc. IEEE, 2016.
- P. I. Frazier, *A tutorial on Bayesian optimization*, arXiv:1807.02811, 2018.
- C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- P. Hennig and C. J. Schuler, *Entropy Search for information-efficient global optimization*, JMLR, 2012.
- J. M. Hernández-Lobato *et al.*, *Predictive Entropy Search for efficient global optimization of black-box functions*, NIPS, 2014.
- Z. Wang and S. Jegelka, *Max-value Entropy Search for efficient Bayesian optimization*, ICML, 2017.
- C. Hvarfner *et al.*, *Joint Entropy Search for maximally-informed Bayesian optimization*, NeurIPS, 2022.
- B. Tu *et al.*, *Joint Entropy Search for multi-objective Bayesian optimization*, NeurIPS, 2022.
- M. Balandat *et al.*, *BoTorch: a framework for efficient Monte-Carlo Bayesian optimization*, NeurIPS, 2020.
- D. R. Jones *et al.*, *Efficient global optimization of expensive black-box functions*, J. Global Optim., 1998.
- N. Srinivas *et al.*, *Gaussian process optimization in the bandit setting: no regret and experimental design*, ICML, 2010.

Thank you!

Questions?

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

ecgarrido@comillas.edu