Experimental Optimization in Quantitative Trading

For: Systematic Investing / Vasant Dhar

David Sweet

Introductions

- Me
  - Quant trader
  - Market making, stocks
  - Experimental optimization
  - andamooka.org
- You
  - Trading
  - Supervised Learning?
  - Backtesting/simulation?
  - Reinforcement learning?
  - ChatGPT / LLMs?

Quant trading

Execution

- Agency: no position risk
- Big customer order ===> many small trades
- Thousands of trades / day

Quant trading

High-frequency market making (HFMM)

- Principal: Small positions for ~10 seconds
- Continuous quoting
- Thousands of trades / day

Quant trading

Statistical arbitrage

- Principal: Larger positions for hours to days
- Opportunistic trade selection
- Tens - hundreds of trades / day

Experimentation

- Compare performance in **live trading**
- Returns predictions incomplete
  - risk, liquidity, capital, preferences
- Simulation (aka backtest) too hard
  - Market reaction / impact
  - Latencies
  - Complexity
  - Counterfactuals: What *would have* happened?

Experimentation: Complexity

- Example: US stock market
  - 13 lit exchanges
  - “around 50” dark pools [https://www.marketswiki.com/wiki/Dark_pool]
  - continuous book, auctions, blind bidding, block trading, internalization
  - orders: limit, market, IOC, FOK, AON, ISO, hide & slide, hidden, post-only

Experimental methods

- Experimental methods b/c
  - Evaluation is expensive: $$/time/risk
  - Evaluation is uncertain (noisy)
- Experimental methods
  - Minimize expense
  - Minimize uncertainty

Experimental methods

- Without experimental method
  - Deploy new strategy, make money
  - ==> “My new strategy is great!”
- With experimental method
  - Run new and old strategies side-by-side
  - P{new beats old} = .53, deltaPnL = \$700 ± \$2200
  - ==> “My new strategy isn’t *bad*.”

Make better decisions

Experimental methods

- Your good ideas probably won’t work.
  https://ai.stanford.edu/~ronnyk/ExPThinkWeek2009Public.pdf
  - Amazon reports < 50% of their A/B tests improve metrics
  - Microsoft reports only 1/3
  - Netflix reports only 10%
- My informal polling: 1/10

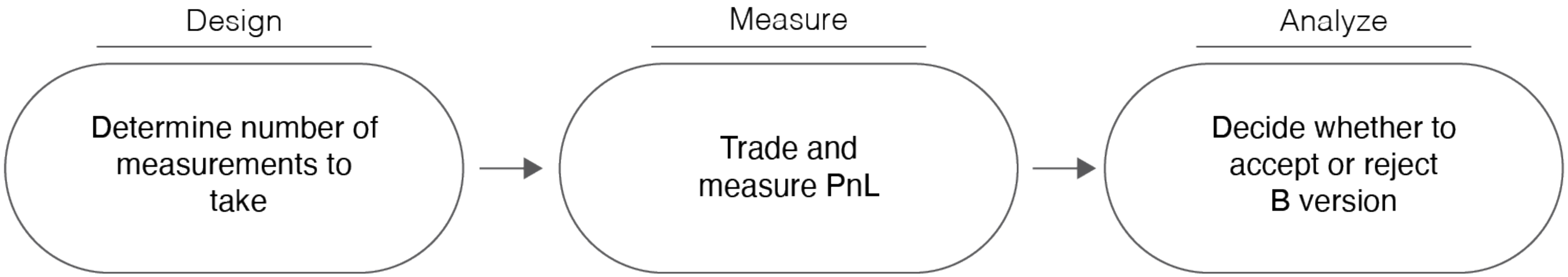
Experimentation pervasive

- Medicine
- Psychology
- Behavioral Economics
- Web search
- Online advertising
- Social media
- Food engineering
- Materials science
- Social science
- Manufacturing
- Consumer product design

Example: HFMM A/B test

- Scenario: HFT strategy
  - A: Existing prediction model, across 100 stocks
  - B: Your new model
- Model B looks better than Model A
  - Lower out-of-sample RMSE
  - Higher PnL in simulations
- Next: A/B test in live trading

Example: HFMM A/B test



Example: HFMM A/B test

- Design:
  - Which stocks will run Model A, Model B?
  - How many days will this run?
- Measure:
  - Trade and record pnl by stock, day: $p_{s,d}$
- Analyze:
  - Which is (probably) better A or B?
  - Significantly better?

Design: Replication

- PnL is noisy
- Replication: Avg. over multiple days

$$\mu = \frac{\Sigma_d p_d}{N} \qquad se = \frac{\sqrt{\Sigma_d (p_d - \mu_d)^2}}{N} = \frac{\sigma}{\sqrt{N}}$$

- Replication decreases “noise” (se) as $1/\sqrt{N}$

Design: Replication

- Before experiment

1. Estimate se ← Data
2. Specify minimum interesting δ_{min} ← Subjective
3. Solve for N:

$$se = \frac{\sigma}{\sqrt{N}} < \delta_{min}/k \quad ==> \quad N > (\frac{k\sigma}{\delta_{min}})^2$$

- (k for safety)

Design: Replication

- Some analysis, $k = 2.8$:
  - Limit false positives (5%)
  - Limit negatives (20%)

$$N >= \left(\frac{2.8\sigma}{\delta_{min}}\right)^2$$

NB: Quadratic

- Typically 1-2 weeks for real experiments

Measure

- Trade!
- Start small for safety
- Stop if *any* metrics look very bad / different
  - PnL terrible — or wonderful!
  - Trading way too little / too much
  - Sending too many / too few orders
- Log everything, record $p_{s,d}$

Measure: Randomization

- Randomly assign 50 stocks to Model B, get $p_{s,d,B}$
- Assign other 50 to Model A, get $p_{s,d,A}$
- Experiment measures ATE = *Average Treatment Effect*

$$\mu_A = \Sigma_{s,d} p_{s,d,A} \quad \mu_B = \Sigma_{s,d} p_{s,d,B}$$

$$ATE = \mu_B - \mu_A$$

Measure: Randomization

- Problem: *Confounders*
- Ex: Assign tech stocks to A, energy stocks to B
- Say you get higher pnl in tech/A
- Do you conclude
  - A is better than B, or
  - Tech is easier than energy?

Measure: Randomization

- Randomly assign stocks to A, B
  - Some tech in A, some in B
  - Some energy in A, some in B
  - Some high liquidity in A, some in B
  - Some high volatility in A, some in B
  - etc.

- Randomization removes confounder bias

Even if you don't know the confounders

Analyze

- Did Model B earn more? Enough to care?

$$ATE = \mu_B - \mu_A > \delta_{min}$$

- Statistically significantly more?

$$t = \frac{ATE}{se_{ATE}} > 1.64$$

Measure: Warning

- Good plan: Wait N days, then ask “Is $t > 1.64$?”
- Bad plan: Stop as soon as $t > 1.64$
  - Much more likely to find $t > 1.64$ than if you wait
  - False Positives
- Analogy
  - Good: Flip a coin N times, ask “More heads than tails?”
  - Bad: Flip a coin up to N times, stop if more heads than tails

Bayesian optimization

- Modern, flexible, efficient method(s)
- A.K.A.
  - Adaptive experimentation
  - Black box optimization
  - Surrogate optimization
  - Model-based optimization

Bayesian optimization

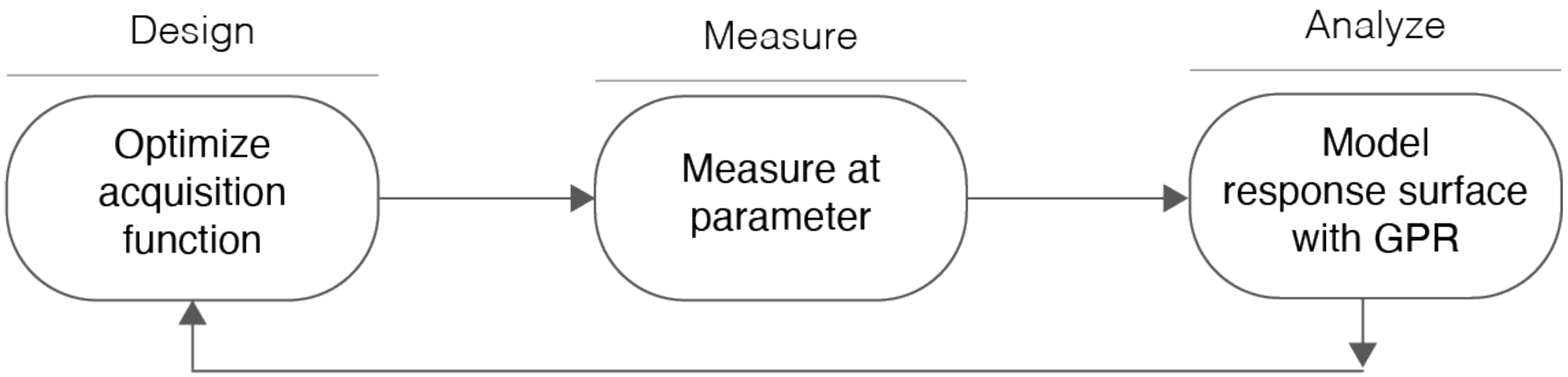
- Can compare A, B
- Also: A, B, C, D, …
- Also: 1, 2, 3, …
- Also: $[0,1]$, $[0,1]^D$
- Also: $\{A, B, C, \ldots\} \otimes \{a, b, c, \ldots\} \otimes \{1,2,3,...\} \otimes [0,1]^D$

- IOW: BO can optimize your strategy's parameters.

Bayesian optimization

- Other uses:
  - Hyperparameter optimization (HPO) for supervised learning models
    - NNs, trees
  - Optimize parameters of strategy in simulation

Bayesian optimization: Overview



Bayesian optimization: Analysis

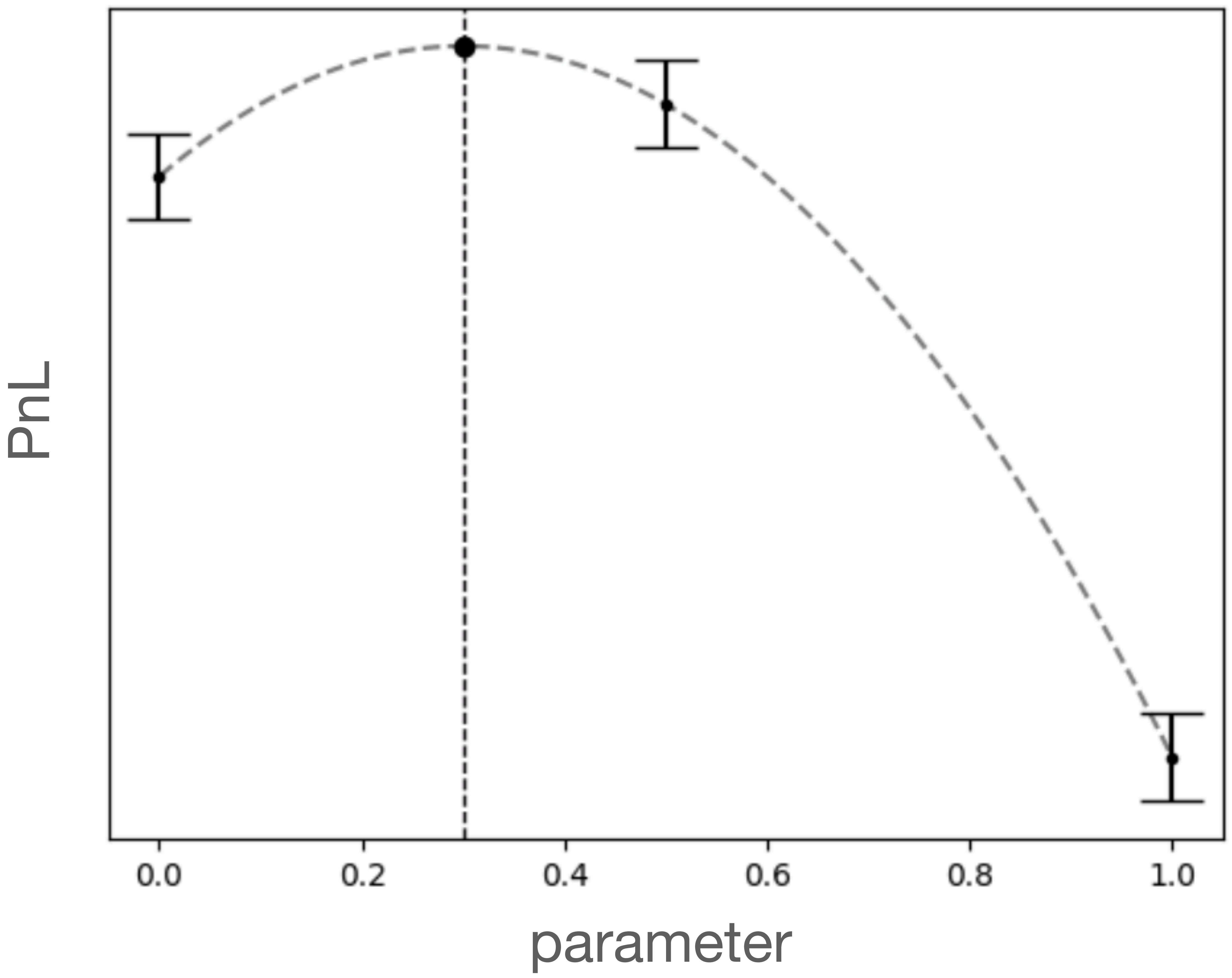
- Say we've already measured something…
- Fit a *surrogate*, a (nonlinear) regression:

$$y(x) = \text{PnL(parameters)}$$

- Then, maximize $y(x)$ over x:

$$x = \underset{x}{\operatorname{argmax}}\, y(x)$$

- **BUT**: Surrogate is poor b/c so few measurements



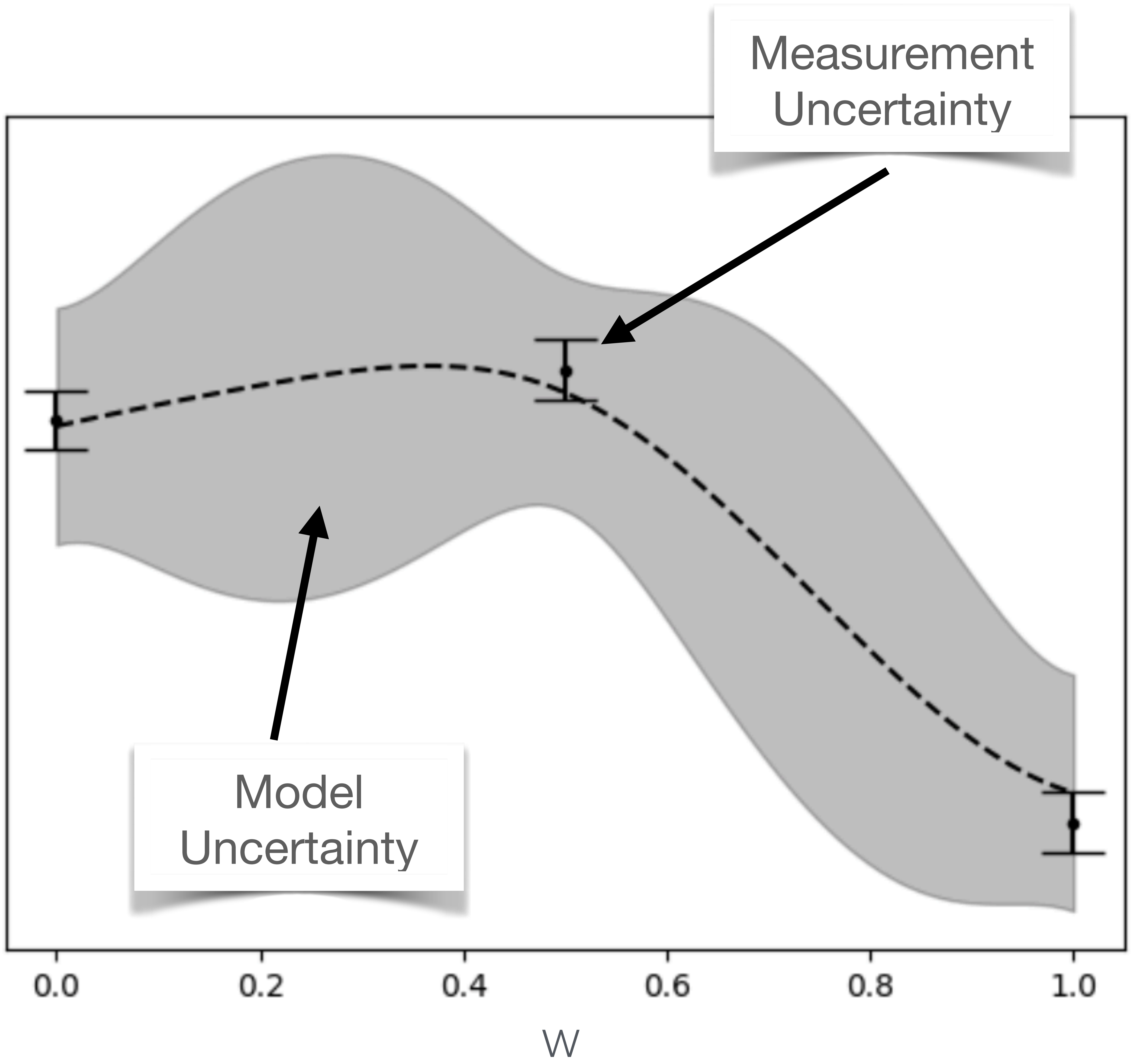
Bayesian optimization: Design

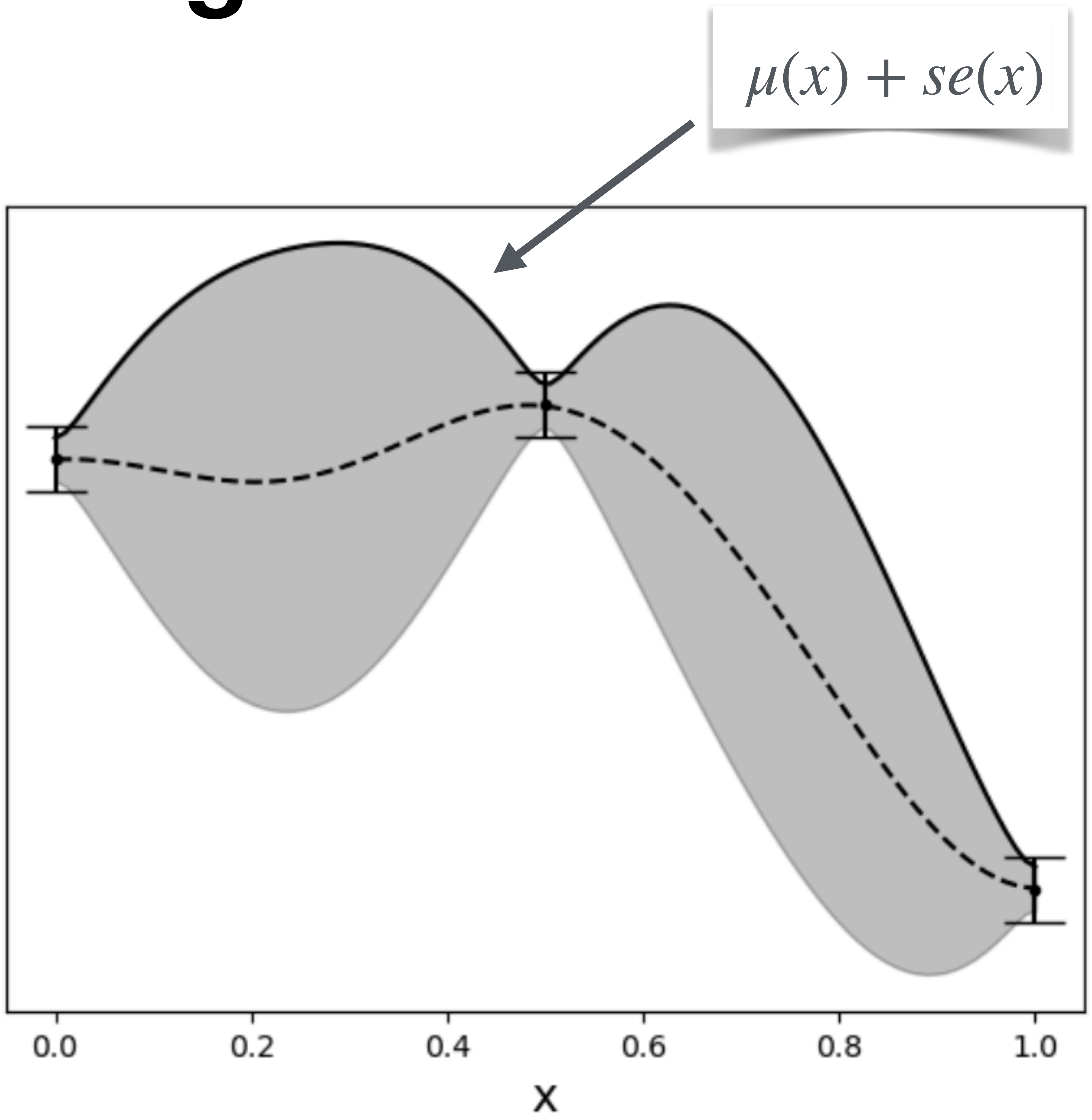
- Instead:

$$x = \underset{x}{\operatorname{argmax}} \Big[\underbrace{\mathrm{E}[y(x)]}^{\text{Exploitation}} + \overbrace{\sqrt{\mathrm{VAR}[y(x)]}}^{\text{Exploration}} \Big]$$

- Special nonlinear regression outputs $\mathrm{E}[y(x)]$ *and* $\mathrm{VAR}[y(x)]$
    - Gaussian process regression (GPR)
- $\mathrm{VAR}[y(x)]$ is *epistemic uncertainty*
    - GPR tells you how confident it is

Bayesian optimization: Design





Bayesian optimization: Design

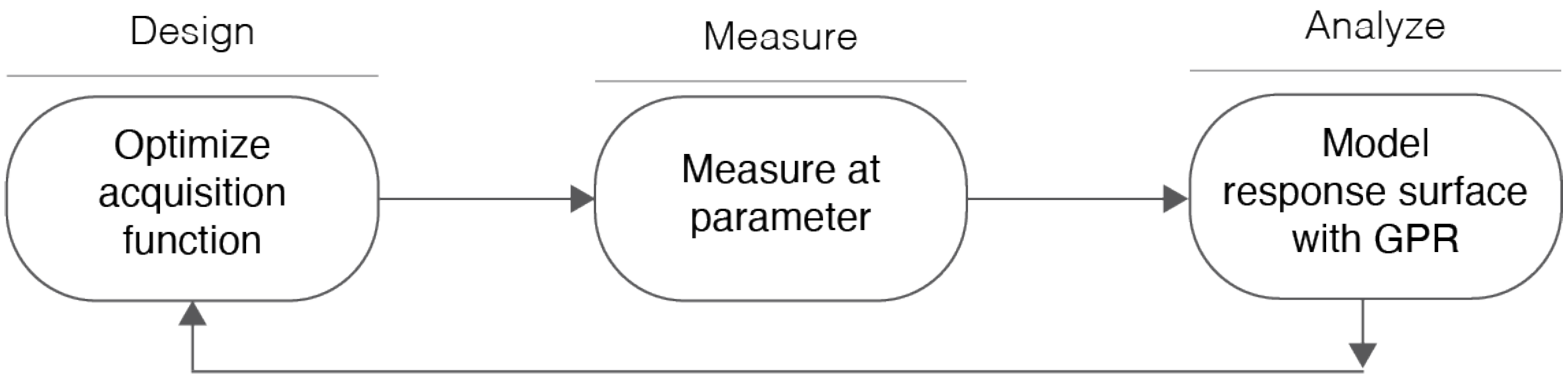
- *Acquisition function*: $\mathrm{E}[y(x)] + \sqrt{\mathrm{VAR}[y(x)]}$
- Exploitation, $\mathrm{E}[y(x)]$
  - Encourages measurements (trading) that increases pnl **now**
- Exploration, $\mathrm{VAR}[y(x)]$
  - Encourages measurements that increase the surrogates confidence
  - …so you'll increases pnl **later**

Bayesian optimization: Measurement

- Same as before
- Go trade

Bayesian optimization: Analysis

- Have you exhausted your budget for experimentation?
- If not, rebuild GPR, design again



Bayesian optimization

- **Mixed variable types**: Continuous, ordinal (integer), and categorical (labels)
- **Multiple metrics**: PnL, risk, volume, order rate, … simultaneously
- **Multiple fidelities**: Combine simulator results w/live results
- **Constraints**: Limit risk, capital, market participation
- **Operations friendly**: Build surrogate from all available measurements

- Build your surrogate as a model of your whole trade.

Bayesian optimization: Tools

- botorch.org
  - Flexible, powerful modeling toolkit

- scikit-optimize.github.io
  - Higher-level, friendlier interface

Bayesian optimization: Frontier

- High-dimensional problems
  - Hundreds of parameters
  - Thousands of measurements
- Multitask optimization
  - Optimizing stock 1 helps you optimize stock 2
  - …which helps even more with stock 3, etc.

Bayesian optimization: Frontier

- Giant, complex spaces
  - Materials, proteins, molecules
- Self-driving labs
  - Loop: BO designs experiment, robots executes it
- LLMs (of course)
  - “Educated” initial guesses
  - Analogizing from related tasks
  - Filtering proposed parameters