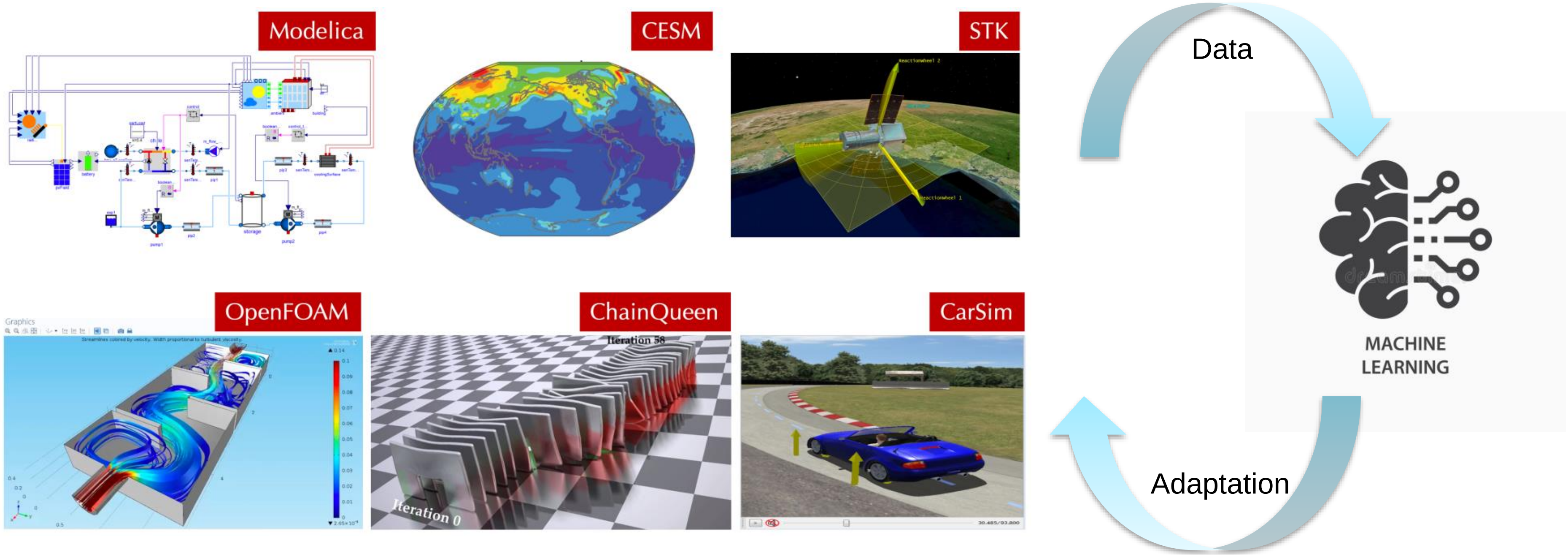
# Physics-Informed Digital Twins

- Digital twins becoming widely available across many different domains
  - *Defn.*: Physics-informed modeling toolkits and simulation environments **containing learning modules capable of adapting to operational data** to maintain high prediction accuracy over product lifetime




- Provides a **rapid**, **scalable**, **safe**, and **repeatable** alternative to field experiments
- May not be entirely transparent
  - (*this part of talk*) Critical to design ML methods that can leverage **physics-informed simulation data** for:
    - Controller design and online performance optimization
    - Improving generalization via multi-source data

# Calibrating (Tuning) Controllers from Simulations

- Control algorithms (e.g. MPC) have a few tunable parameters, say $x \in X$:
  - Model: unknown parameters, scenario tree
  - Constraints: backoff values, terminal constraints
  - Objective: weights in stage cost, horizon, terminal cost
  - Solvers: tolerance, discretization scheme
  - References: set-points, path trajectories

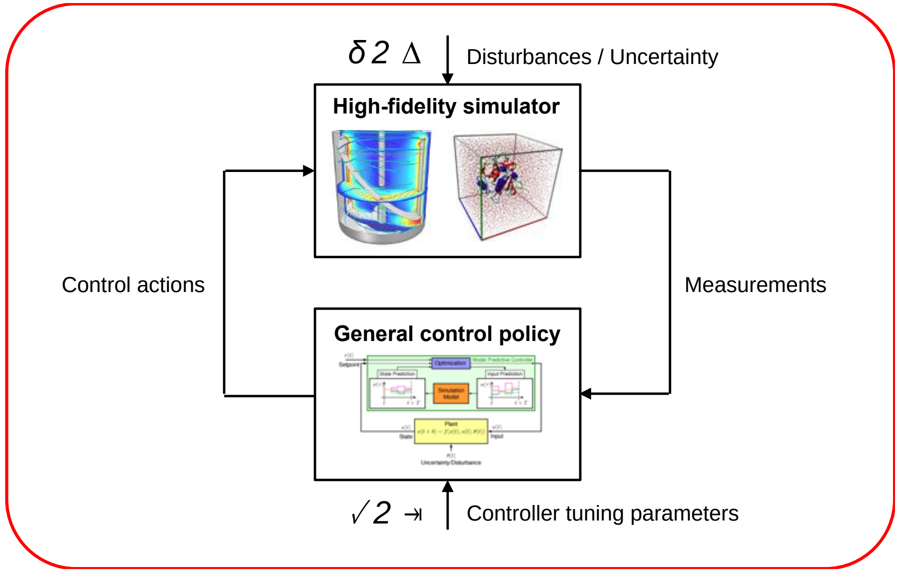
- Must define performance index $f(x)$ over closed-loop simulation: $f$ may be nasty (= expensive, no gradients)

- Auto-tuning = global optimization problem, $\max_{x \in X} f(x)$

- **Bayesian Optimization (BO)** is an effective auto-tuning strategy when working directly with simulation data

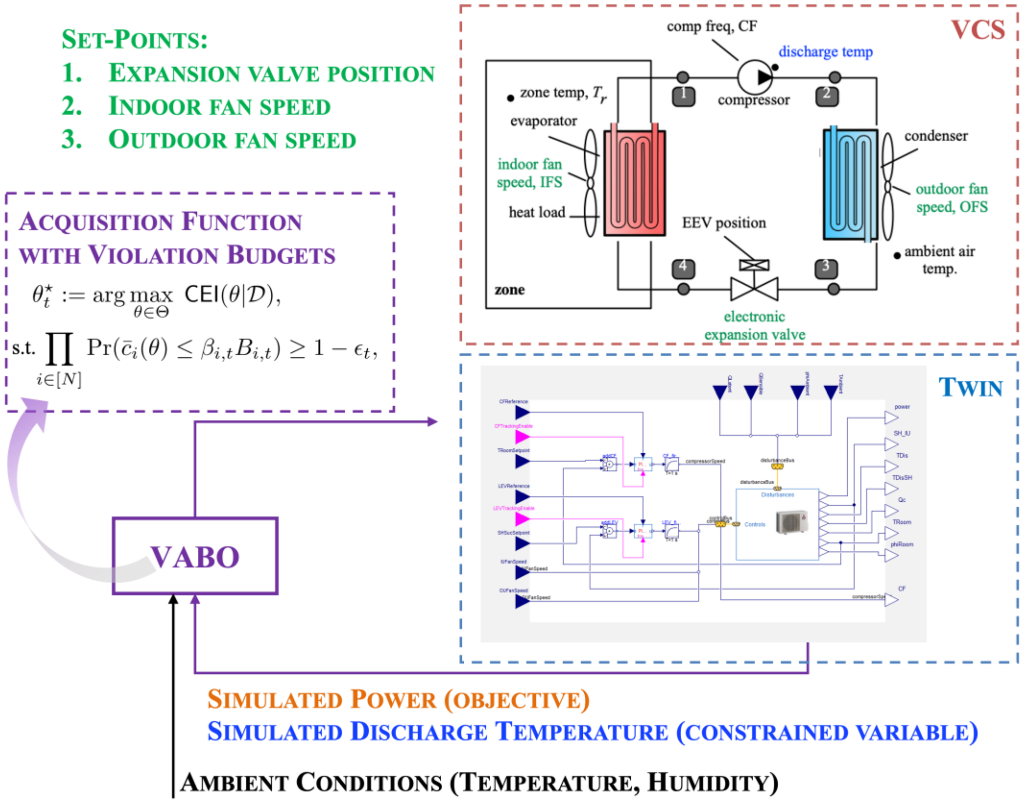
Closed-loop performance measure

$$\min_{\surd 2 \rightarrowtail} \mathrm{E}\{f(\surd, \delta)\}$$



# Bayesian Optimization (BO)

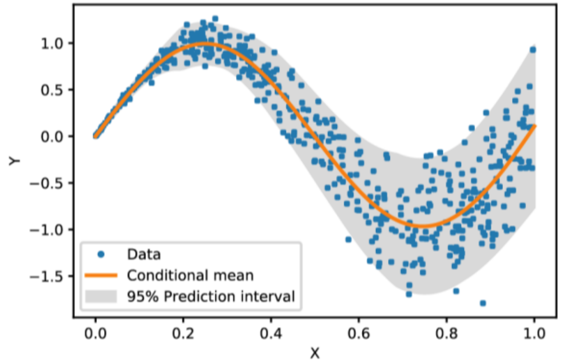
- BO **does not require any knowledge of structure** and/or gradient of $f$ and can tolerate **noisy** observations
  - Flexible
- BO avoids getting “stuck” in local solutions (**global optimization**)
  - Can “teleport” throughout design space
- BO requires **few evaluations** and concurrently **learns a surrogate model** of the CL performance
- Check out the paper for a real-world use-case of BO-based safe controller tuning in vapor compression systems!



# BO is Extremely Flexible & Can Exploit Structure!

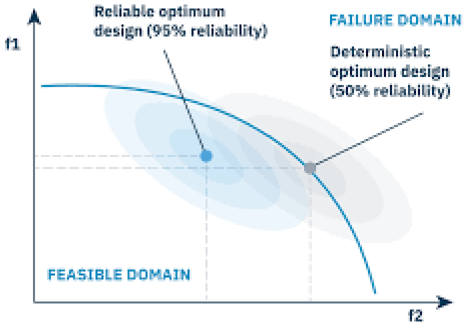
- Input-dependent noise $f(x) = g(x) + \varepsilon(x)$

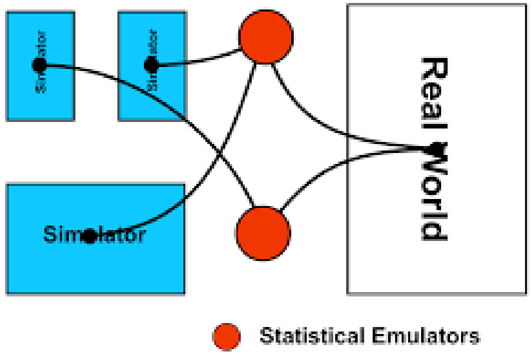
  [Kersting et al., 2007; Lazaro-Gredilla et al., 2011]




- Robust optimization $f(x) = \max_{w \in \mathcal{W}} g(x, w)$

  [Paulson et al., 2021; Kudva et al. 2022]

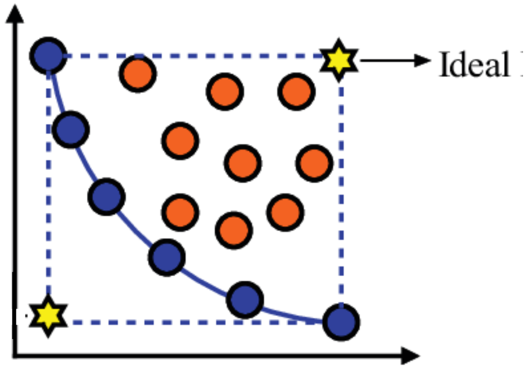



- Multi-objective & constraints

$$f(x) \;\; \text{s.t.} \;\; c(x) \leq 0$$

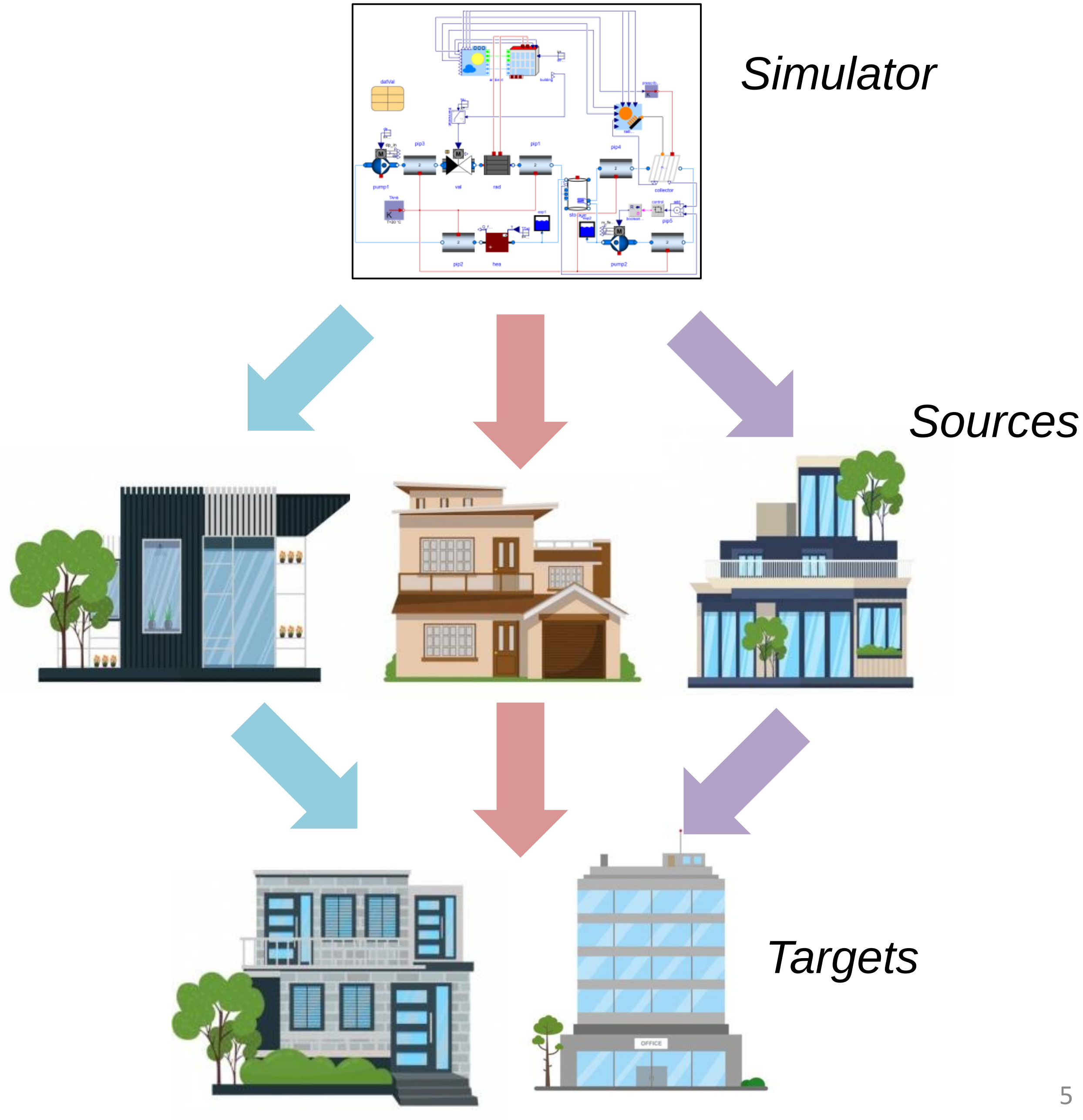
  [Makrygiorgos et al., 2022; Lu et al., 2022, Xu et al. 2022]



# Simulation Environments Enable Generation of Multi-Source Data

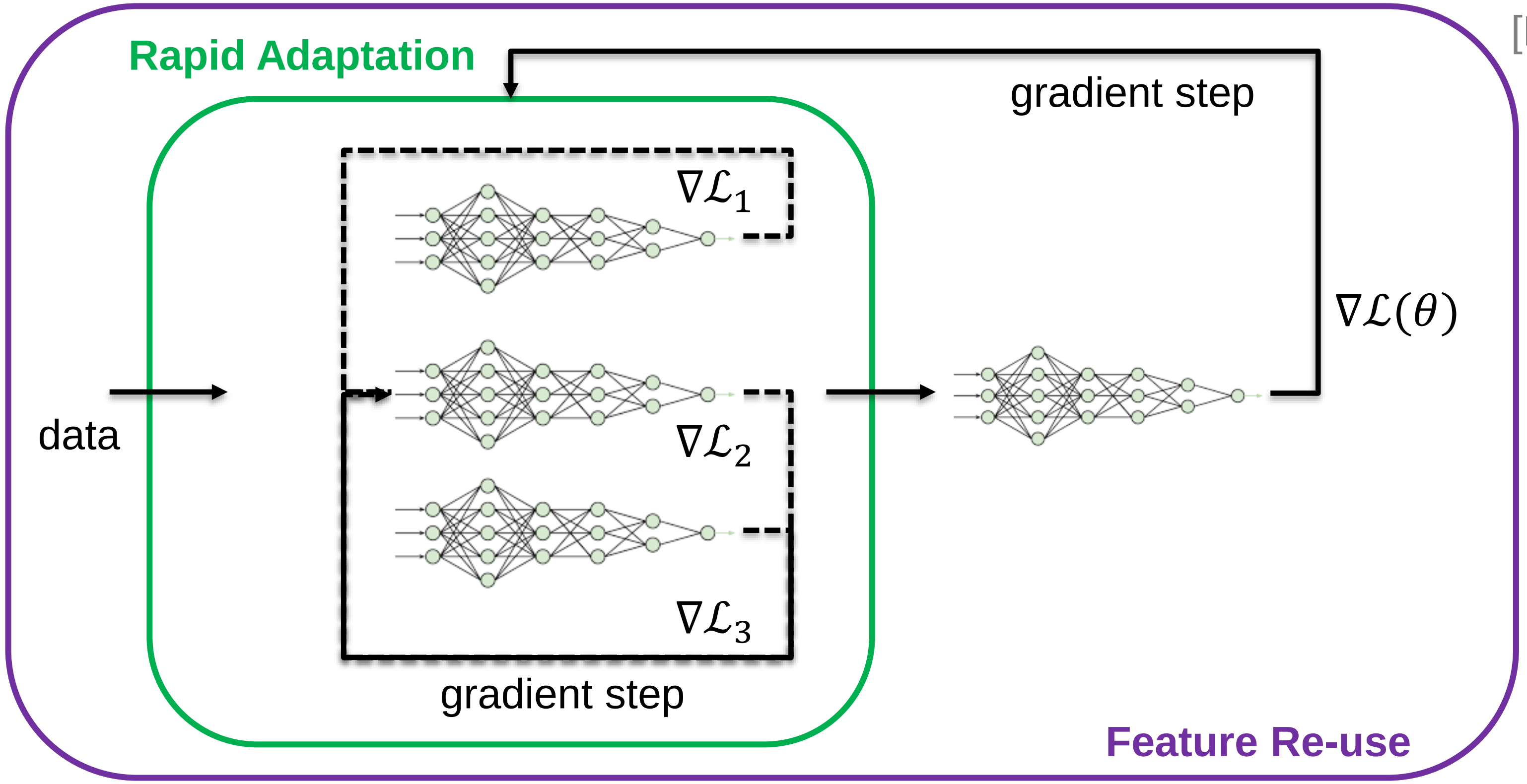
- PI digital twins comprise **parameterized components** that can be modified to generate useful data from multiple source systems similar to the target system
  - This **multi-source dataset**:
    - Reduces data requirements from one system by harnessing data from similar systems

- Specific ML frameworks are suited to learning from multi-source data:
  - **Meta-learning and transfer-learning**



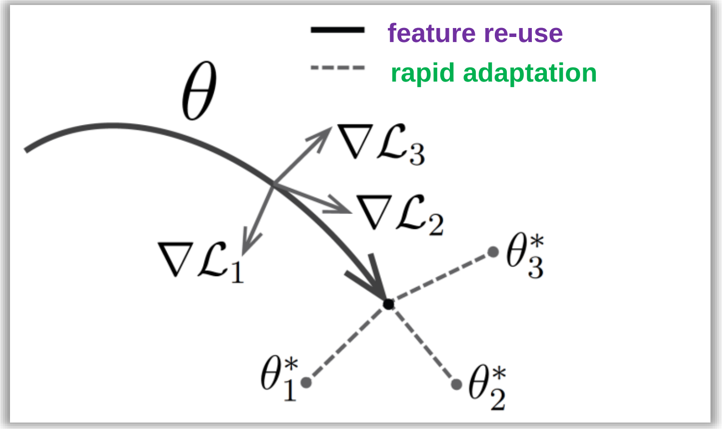
# Meta-Learning

- Classical meta-learners rely on 2 training loops:
  - A feature re-use loop for learning features useful over all source data i.e., task-independent
  - A rapid adaptation loop for network to perform well on single task i.e., task-specific
  - The **adaptation is "baked in"** to the training procedure: this will be exploited at inference



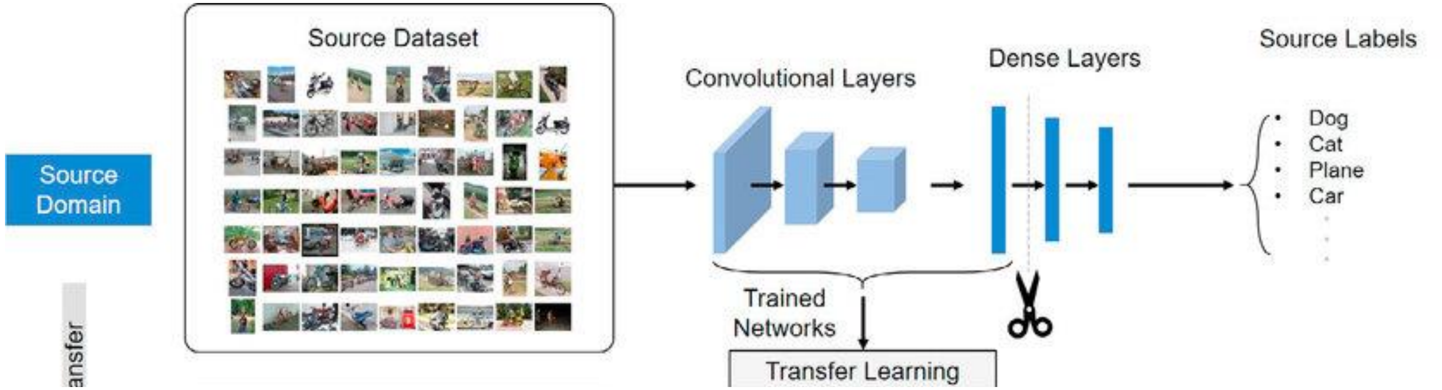

[Finn et al., 2017; Raghu et al., 2019]

*Source*: bair.berkeley.edu



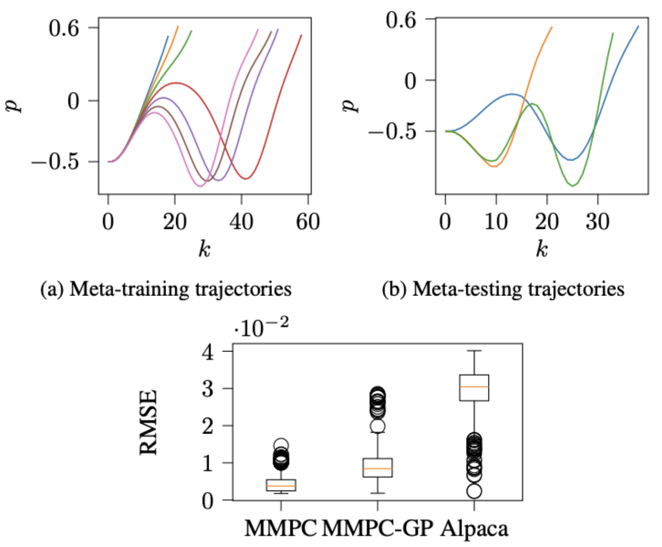
# Transfer-Learning

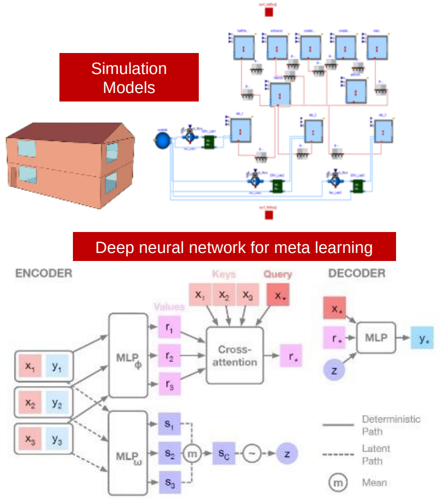
- Transfer learning involves two separate single-loop training procedures
  - 1st training phase: learn task-independent representations
  - 2nd training phase: pre-trained network **modified** and **fine-tuned** for the target task
  - Unlike meta-learning:
    - Adaptation is made problem-specific by 2nd phase, not baked in
    - 2nd-phase training not necessarily few-shot (offline)
    - Same network may not be used at training v. inference

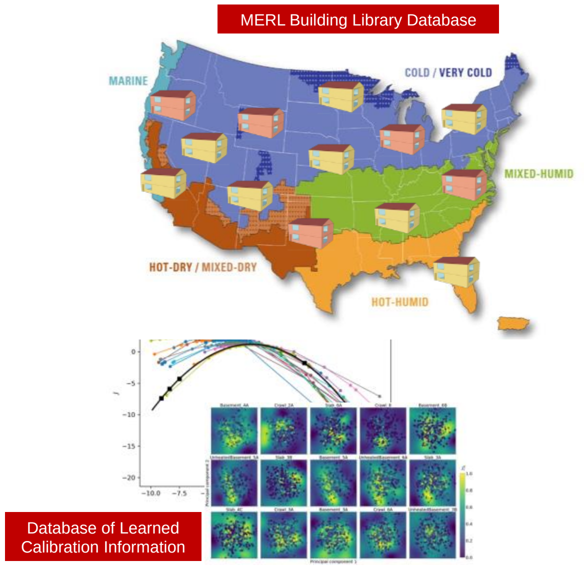


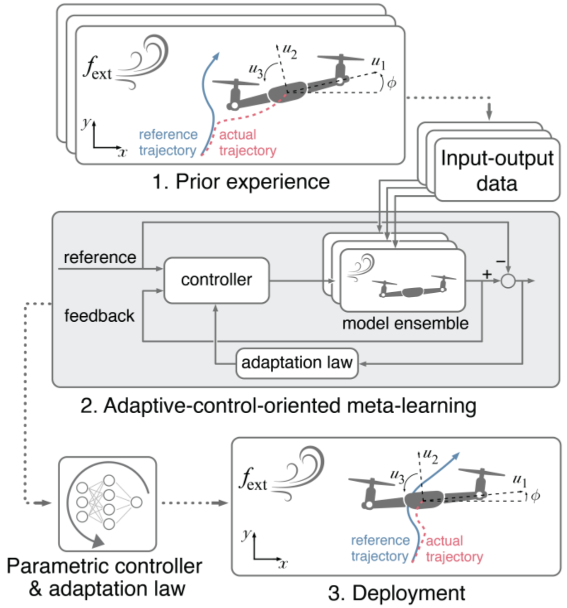
# Snapshot of Multi-Source Learning for Control

- Meta-learning for model identification:
  - Meta-GP for MPC [Arcari et al., 2022]
  - Meta-learned state-space models [Chakrabarty et al., 2023]
- Meta-learning for controller and parameter tuning:
  - Meta-learned BayesOpt [Chakrabarty, 2022; Zhan et al., 2022]
  - Meta-learned adaptive control [Richards et al., 2021; Richards et al., 2022]
- Transfer-learning for modeling and control [Chen et al., 2020; Xu et al., 2020]
- Multi-fidelity auto-tuning [Sorourifar et al., 2021]



# Opportunities for PIML4C

1. Modeling of human(-in-the-control-loop) behaviors
2. Automated adaptation via simulation (closing sim2real gap)
3. Integration of multimodal signals (speech, image/video) into control loops
4. Interpretable and verifiable control policies via physics injection during learning e.g. explicit MPC
5. Providing safety and performance guarantees – new theoretical tools

# Challenges for PIML4C

1. Quantification of ‘minimal’ = ‘useful’ and ‘useful’ data requirements
2. Quantification of uncertainty and modeling errors
3. Quantification of similarity in multi-source learning
4. Formal guarantees of safety/stability for closed loop PIML based controllers
5. Scaling up of verification methods

# Summary & Conclusions

- Classically, ML methods do not integrate known physical constraints

- PIML methods have emerged to systematically combine data-driven ML with physics-based modeling

- PIML methods offer new and exciting opportunities for human-in-the-loop, multi-scale and multi-physics systems

- Open challenges remain: **an exciting time to solve problems in PIML!**