Implicit Regularization in Overparameterized Bilevel Optimization

Paul Vicol, Jonathan Lorraine, David Duvenaud, Roger Grosse




ICML 2021 Workshop
Beyond First-Order Methods in ML Systems

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

Bilevel Optimization & Hypergradients

- Bilevel optimization consists of two *nested sub-problems:*

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}^*)$$
$$\mathbf{y}^* \in \mathcal{S}(\mathbf{x}) = \arg\min_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$$

- **Examples:** *hyperparameter optimization, meta-learning, GANs, NAS, etc.*

Bilevel Optimization & Hypergradients

- Bilevel optimization consists of two *nested sub-problems:*

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}} F(\mathbf{x}, \mathbf{y}^*)$$

$$\mathbf{y}^* \in \mathcal{S}(\mathbf{x}) = \arg\min_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$$

- **Examples:** *hyperparameter optimization, meta-learning, GANs, NAS, etc.*

- When the inner or outer problem is *overparameterized*, there are many equally good solutions, so the argmins are *not unique*
  - The optimization dynamics can lead to implicit regularization effects

- We show that behavior depends to a surprising degree on choices such as the algorithm and hypergradient approximation used

Computing the Response Jacobian

$$\frac{dF(\mathbf{x}, \mathbf{y}^*)}{d\mathbf{x}} = \frac{\partial F}{\partial \mathbf{x}} + \frac{\partial F}{\partial \mathbf{y}^*} \boxed{\frac{\partial \mathbf{y}^*}{\partial \mathbf{x}}} \longleftarrow \textit{response Jacobian}$$

Computing the Response Jacobian

$$\frac{dF(\mathbf{x}, \mathbf{y}^*)}{d\mathbf{x}} = \frac{\partial F}{\partial \mathbf{x}} + \frac{\partial F}{\partial \mathbf{y}^*} \boxed{\frac{\partial \mathbf{y}^*}{\partial \mathbf{x}}}$$

response Jacobian

- The two main ways to compute the response Jacobian are:
    1. Differentiation through unrolling (a.k.a. *iterative differentiation*)

$$\frac{d\mathbf{y}^*}{d\mathbf{x}} \approx \frac{d\Phi_k(\mathbf{y}_0, \mathbf{x})}{d\mathbf{x}}$$

    2. *Implicit differentiation*, applicable when we are at the converged solution to the inner problem:

$$\frac{d\mathbf{y}^*}{d\mathbf{x}} = -\underbrace{\left(\frac{\partial^2 f}{\partial \mathbf{y} \partial \mathbf{y}^\top}\right)^{-1}} \frac{\partial^2 f}{\partial \mathbf{y} \partial \mathbf{x}}$$

Can use *truncated Neumann series approximation*

$$\left(\frac{\partial^2 f}{\partial \mathbf{y} \partial \mathbf{y}^\top}\right)^{-1} \approx \sum_{j=0}^{k} \left(I - \frac{f}{\partial \mathbf{y} \partial \mathbf{y}^\top}\right)^j$$

Cold-Start and Warm-Start Bilevel Optimization

- **Cold-start:** re-initialize the inner parameters and run the inner optimization to convergence each time we compute the gradient for the outer parameters
  - *Impractical* due to the computational expense of full inner optimization

```
while True:
	out_params = outer_step()
	in_params = init_inner()
	while not converged:
		in_params = inner_step()
```

- **Warm-start:** jointly optimize the inner and outer parameters in an *online fashion*, e.g., alternating gradient steps with their respective objectives
  - The *optimization dynamics* can lead to an implicit regularization effect

```
while True:
	out_params = outer_step()
	in_params = inner_step()
```

Dataset Distillation

- We focus on *dataset distillation* as the setup for our toy tasks:

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}} \mathcal{L}_V(\mathbf{x}, \mathbf{y}^*) \qquad \text{s.t.} \qquad \overbrace{\mathbf{y}^* \in \arg\min_{\mathbf{y}} \mathcal{L}_T(\mathbf{x}, \mathbf{y}^*)}^{\text{Training a model on the synthetic data}}$$

Learned datapoints

“Validation” loss on the original dataset

Model params

“Training” loss on the synthetic data

Dataset Distillation

- We focus on *dataset distillation* as the setup for our toy tasks:

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}} \mathcal{L}_V(\mathbf{x}, \mathbf{y}^*) \qquad \text{s.t.} \qquad \mathbf{y}^* \in \overbrace{\arg\min_{\mathbf{y}} \mathcal{L}_T(\mathbf{x}, \mathbf{y}^*)}^{\text{Training a model on the synthetic data}}$$

Learned datapoints

"Validation" loss on the original dataset

Model params

"Training" loss on the synthetic data

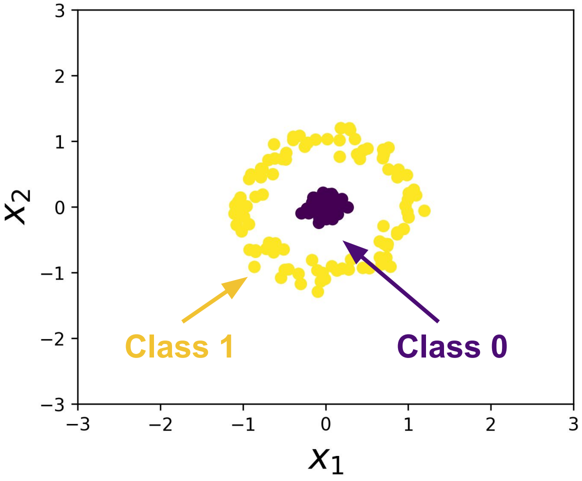
- The *outer objective is only used directly to update the outer variables*

  ➡ It seems intuitive that all of the information about the outer objective is *compressed into the outer variables*.

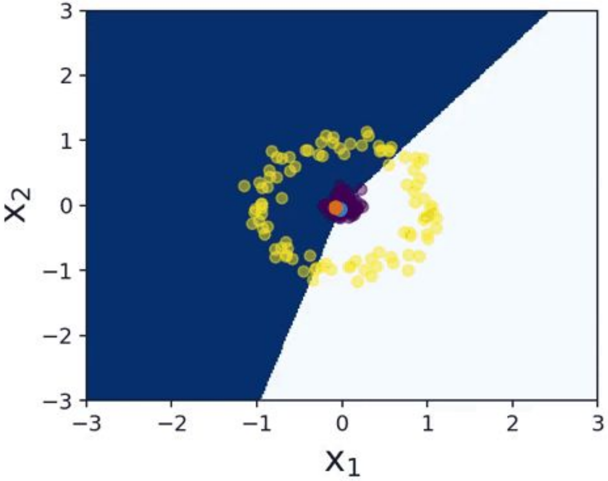
Warm-Start Phenomena

- It seems intuitive that information about the outer objective is *compressed into the outer variables*
  - We show that this is not the case when the inner problem is overparameterized
- Consider *warm-start optimization* to jointly optimize *a model and 2 learned datapoints (1 per class)*

Original dataset

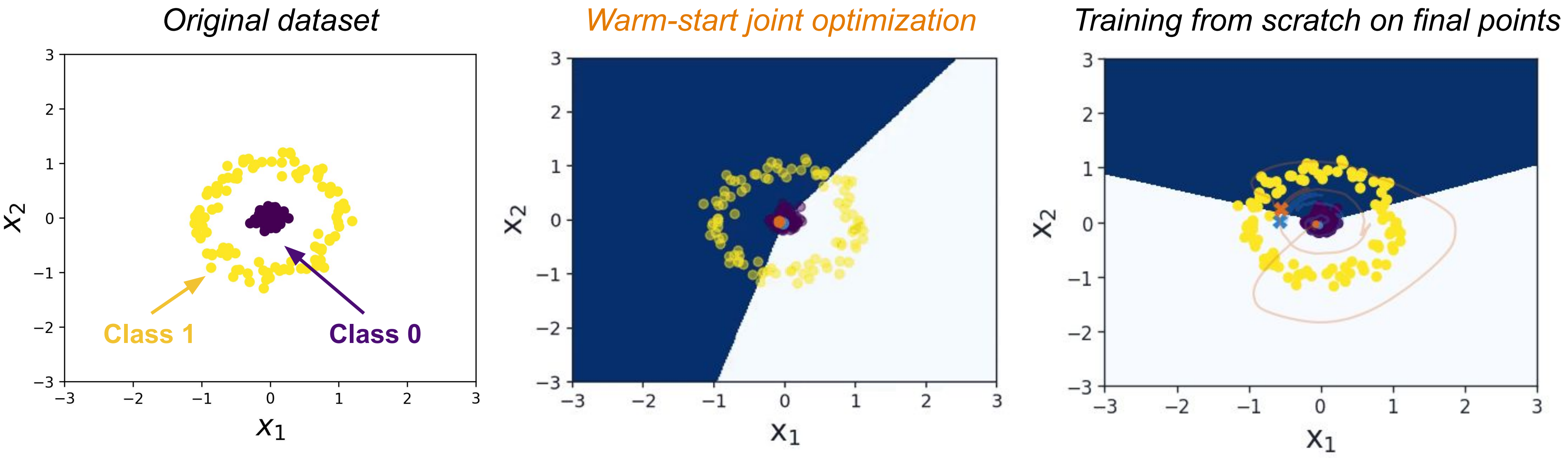


Warm-start joint optimization



Warm-Start Phenomena

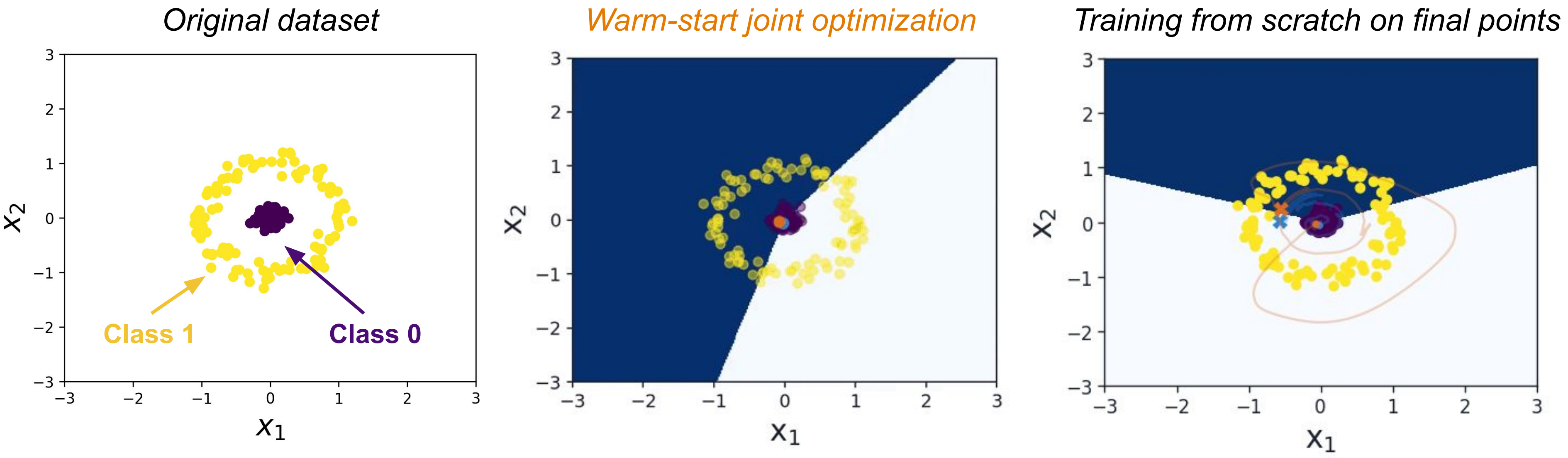
- It seems intuitive that information about the outer objective is *compressed into the outer variables*
  - We show that this is not the case when the inner problem is overparameterized
- Consider *warm-start optimization* to jointly optimize *a model and 2 learned datapoints (1 per class)*



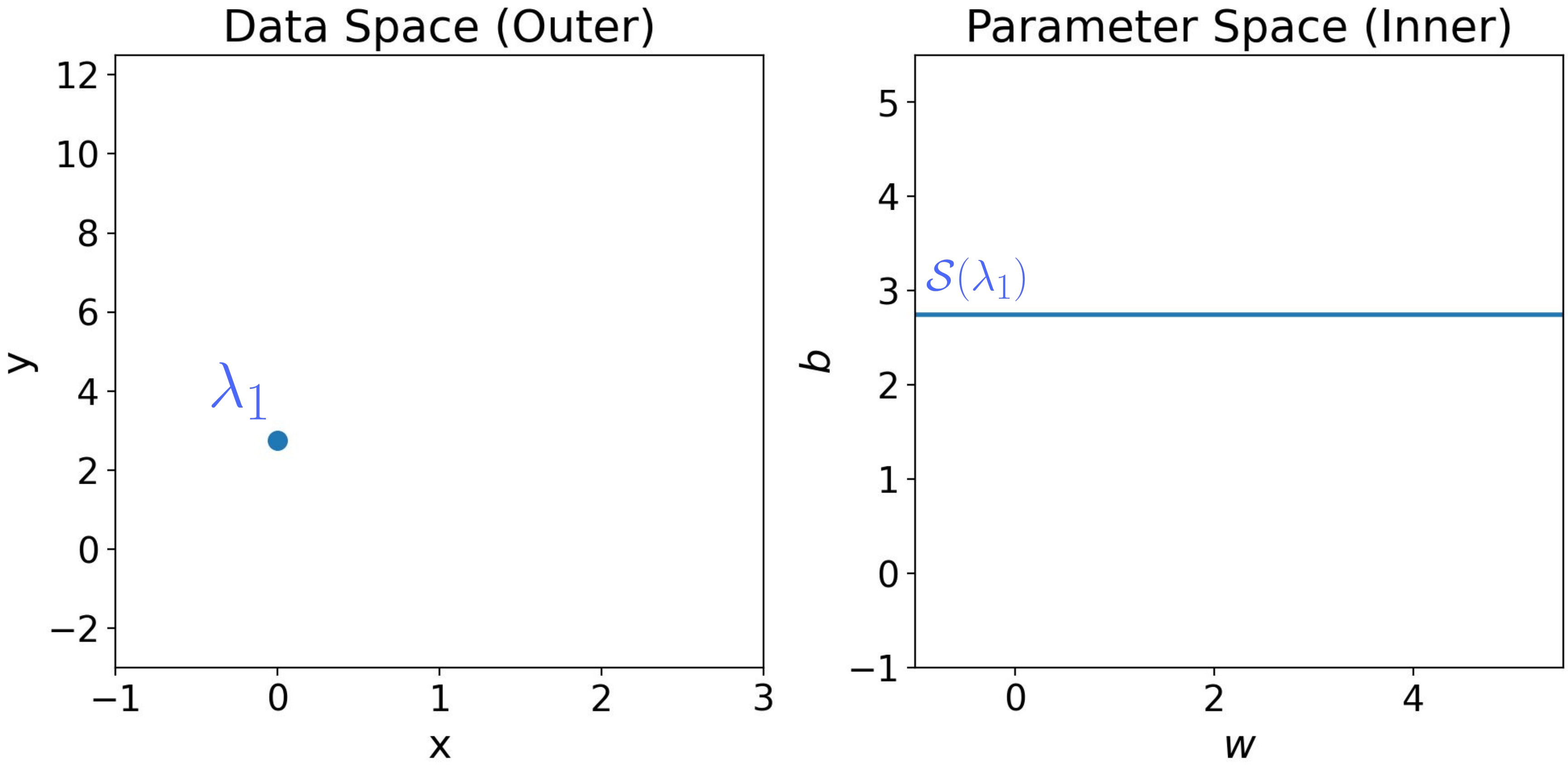
Warm-Start Phenomena

- It seems intuitive that information about the outer objective is *compressed into the outer variables*
  - We show that this is not the case when the inner problem is overparameterized
- Consider *warm-start optimization* to jointly optimize *a model and 2 learned datapoints (1 per class)*

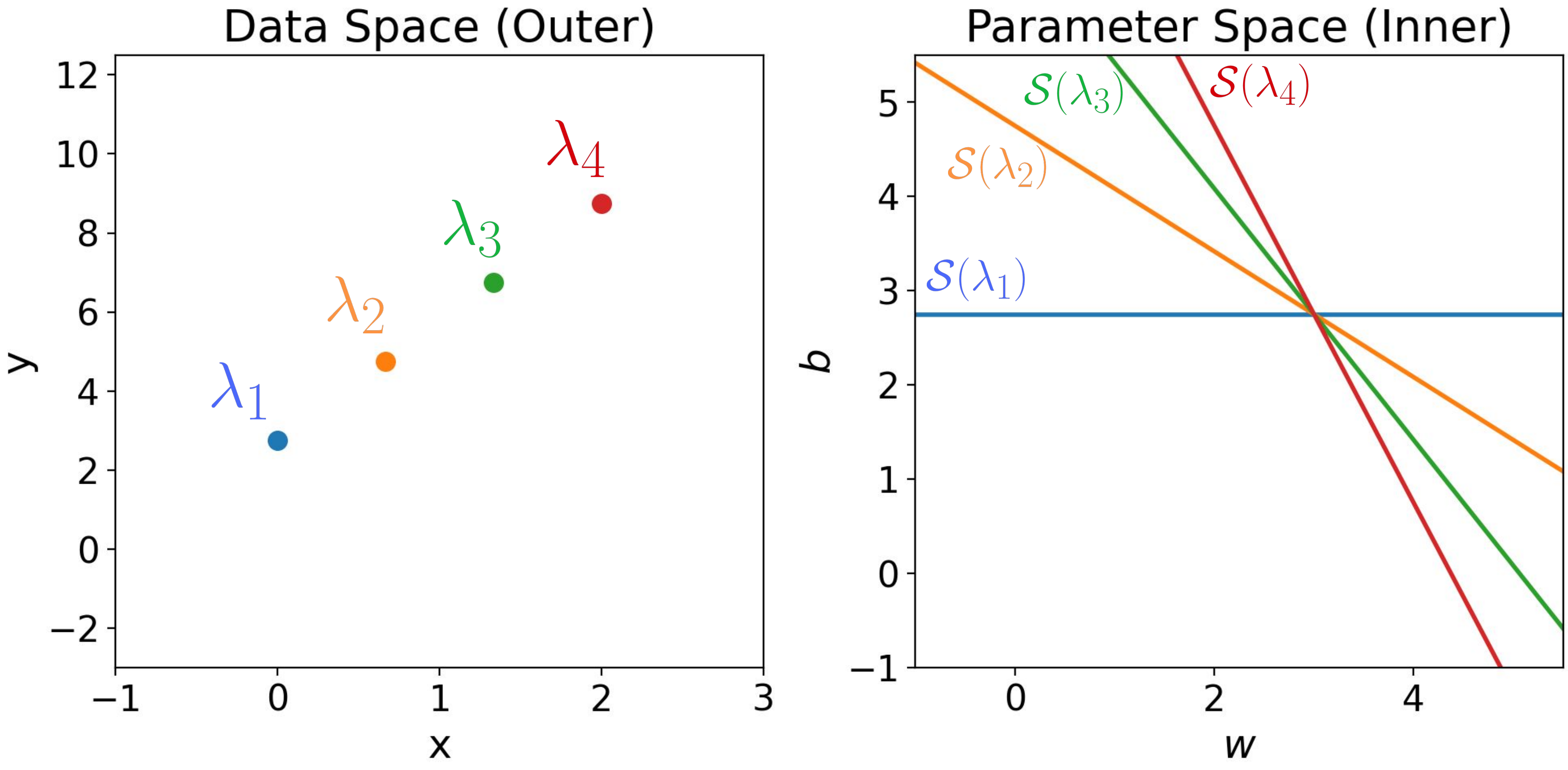
  - **Takeaway:** A surprising amount of *information about the outer objective can leak to the inner parameters*, even when the outer parameters are low-dimensional



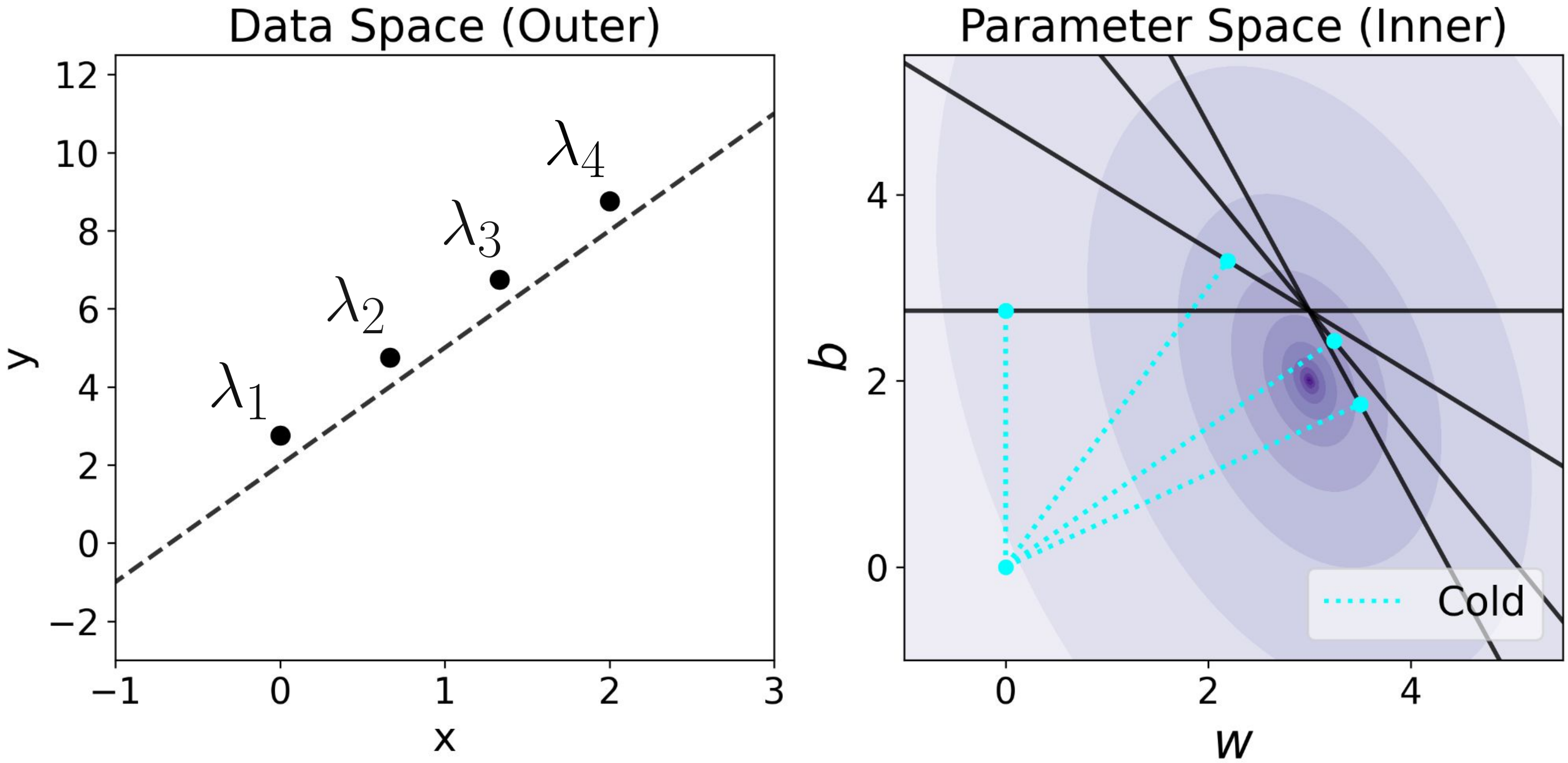
Intuition for Cold-Start and Warm-Start Behavior



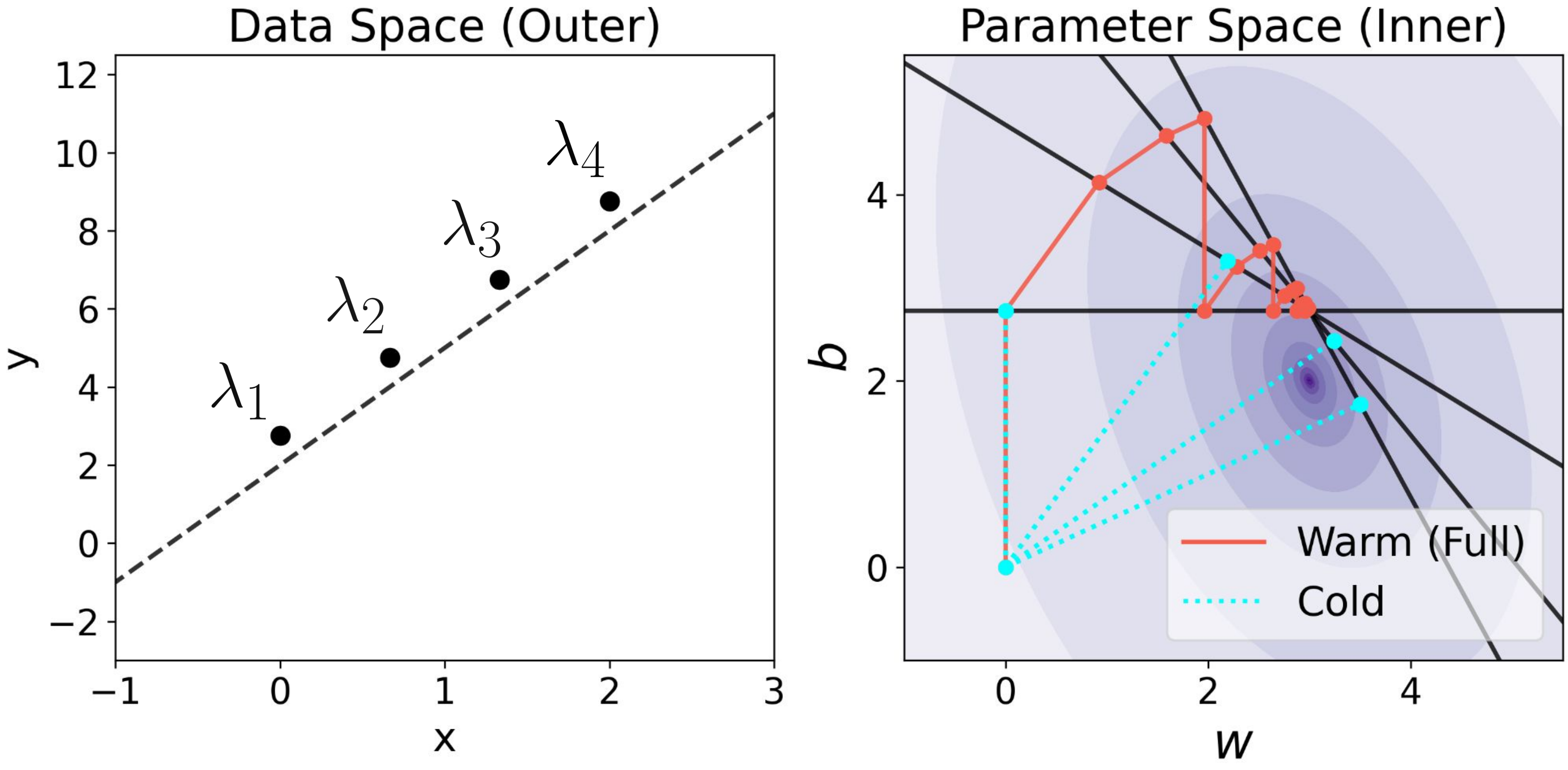
Intuition for Cold-Start and Warm-Start Behavior



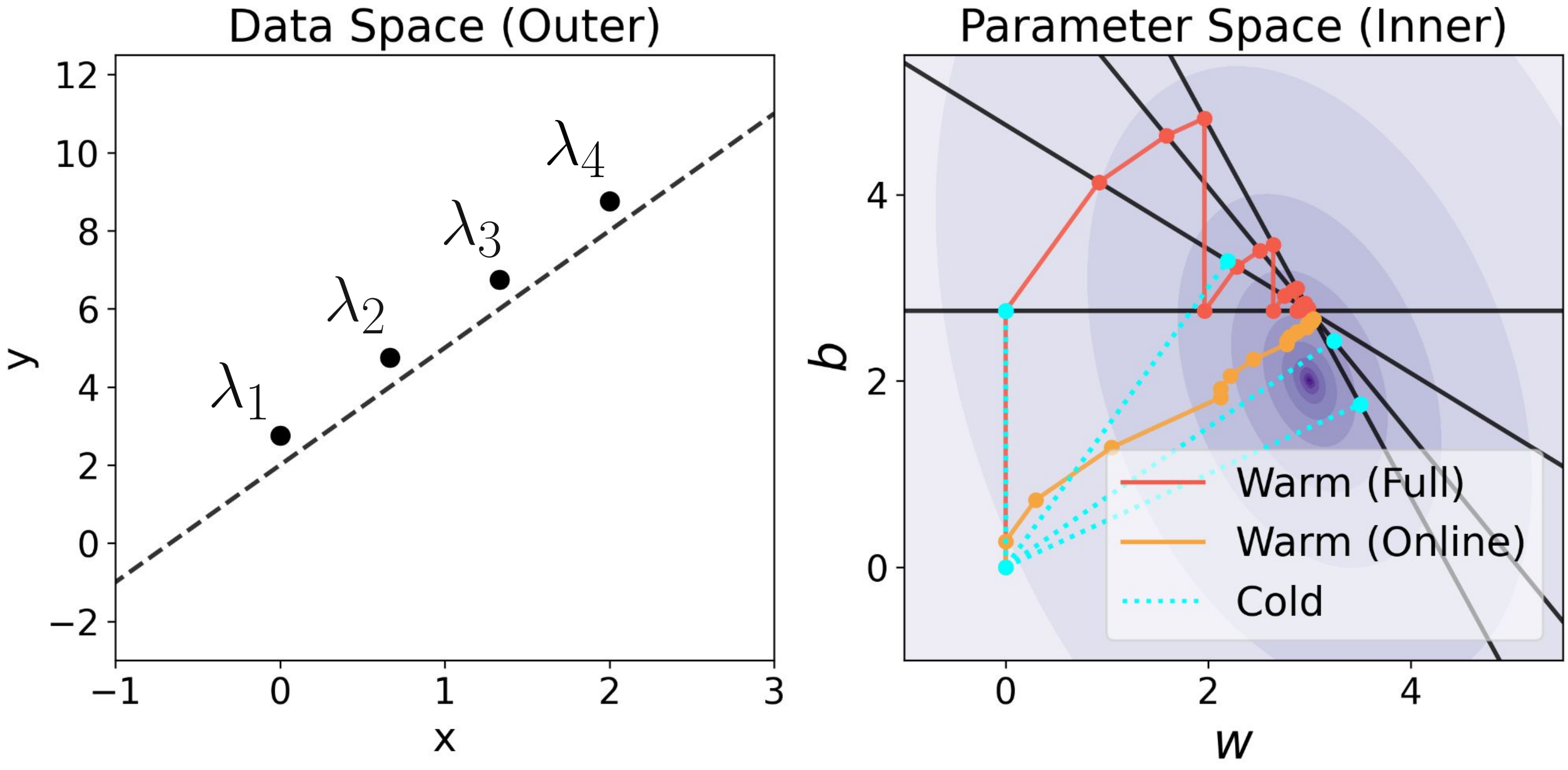
Intuition for Cold-Start and Warm-Start Behavior



Intuition for Cold-Start and Warm-Start Behavior



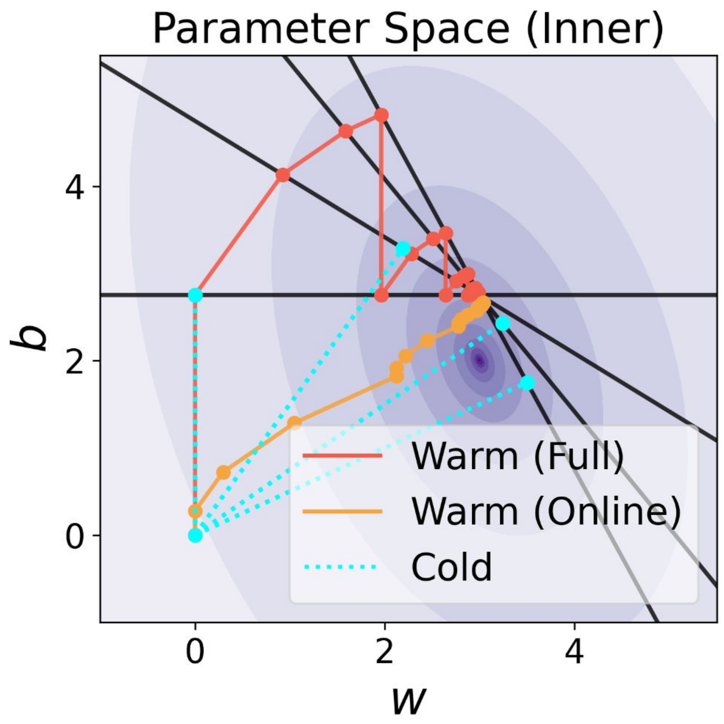
Intuition for Cold-Start and Warm-Start Behavior



Intuition for Cold-Start and Warm-Start Behavior

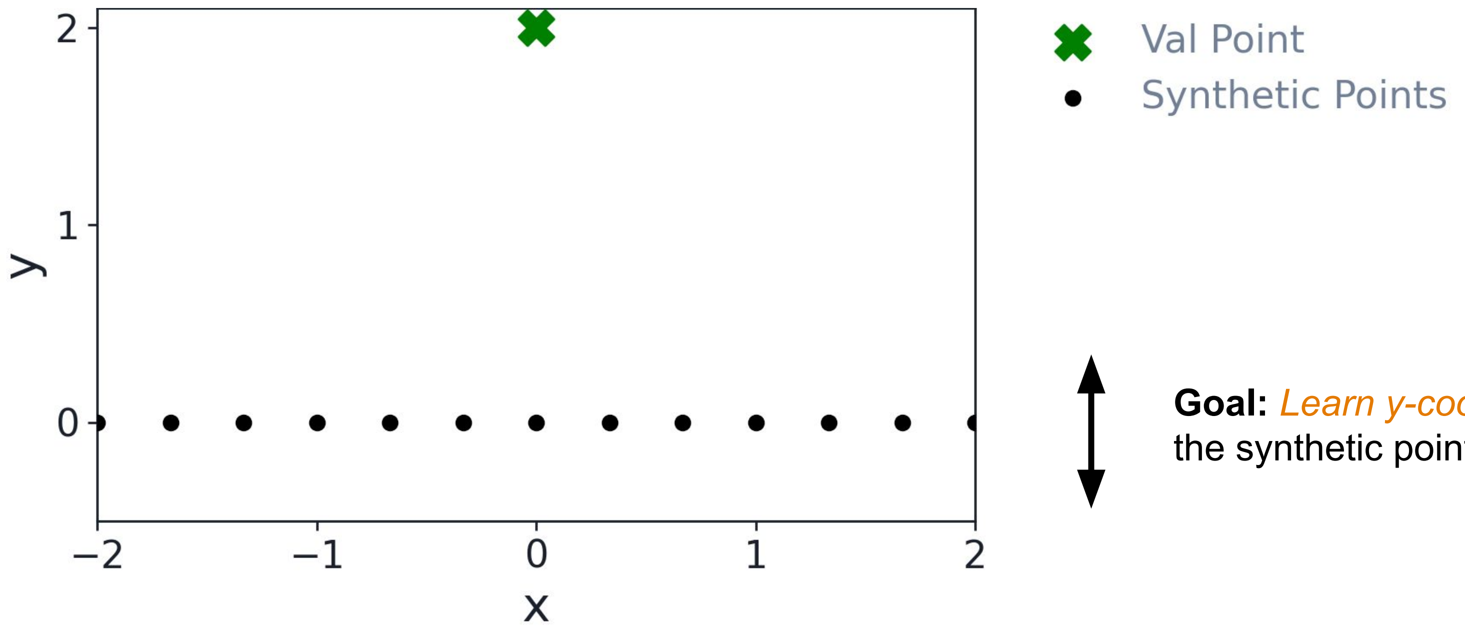
- *Cold-start always projects from the origin onto the solution set for the current datapoint*

- *Warm-start projects from the current weights onto the solution set*
    - By successive projection between solution sets, the inner parameters will *converge to the intersection of the solution sets, yielding inner params that perform well for multiple outer params simultaneously*
    - Note that we do not necessarily converge to the optimal validation loss



Outer Overparameterization: Anti-Distillation

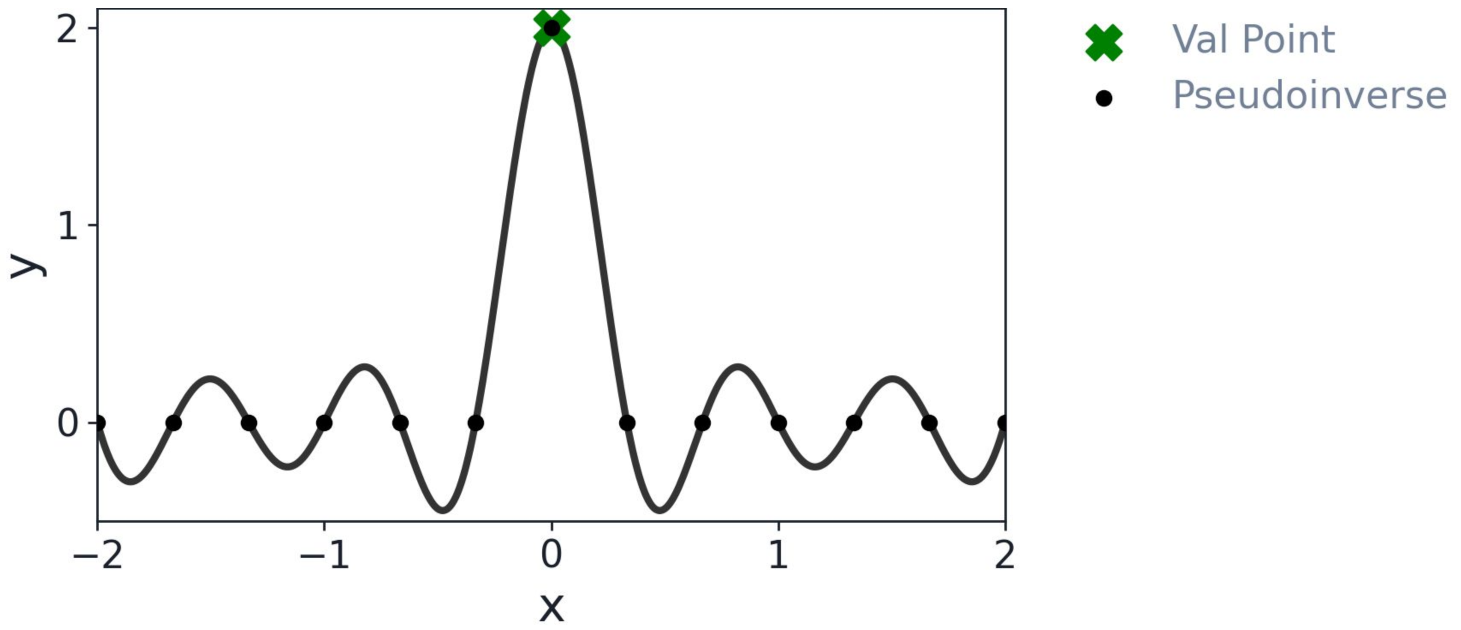
- **Anti-distillation:** *more learned datapoints than original dataset examples*
- The *quality of hypergradient approximations induces a trade-off between the inner and outer parameter norms*—e.g., we can achieve the good performance for the outer objective by either making larger updates to the inner or the outer parameters



Goal: *Learn y-coords* of the synthetic points

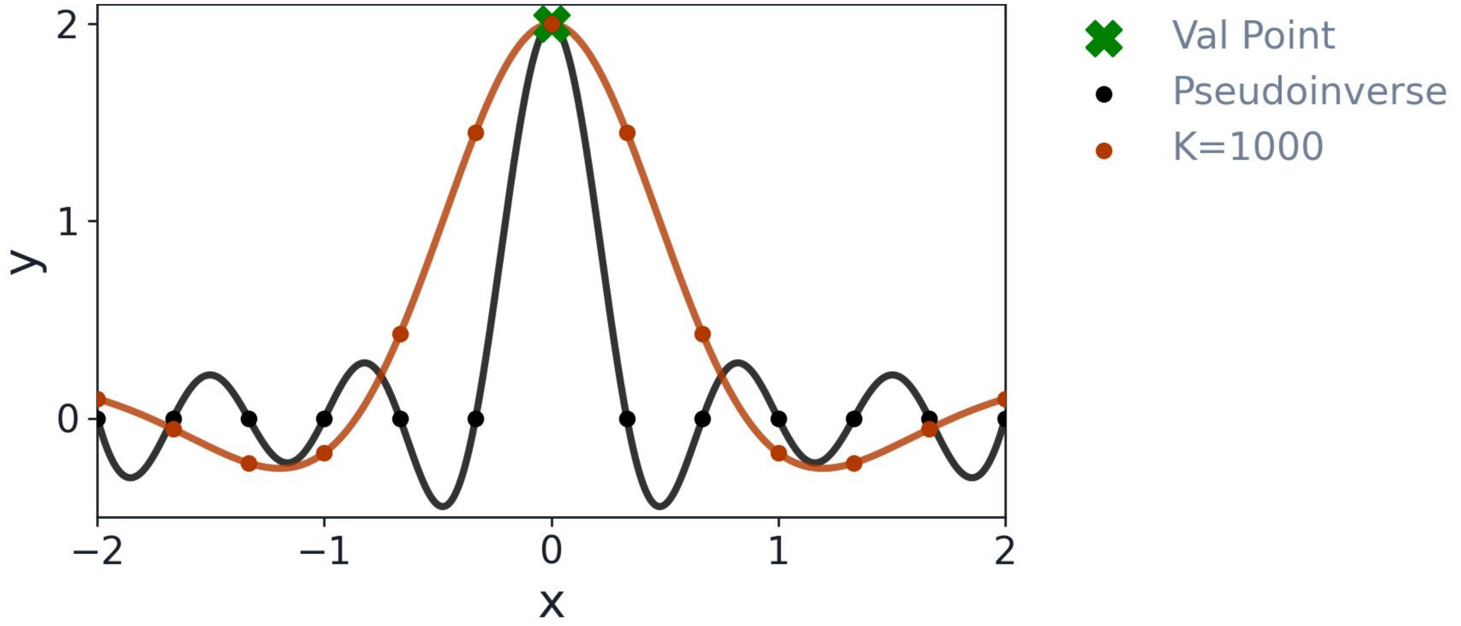
Outer Overparameterization: Anti-Distillation

- **Anti-distillation:** *more learned datapoints than original dataset examples*
- The *quality of hypergradient approximations induces a trade-off between the inner and outer parameter norms*—e.g., we can achieve the good performance for the outer objective by either making larger updates to the inner or the outer parameters



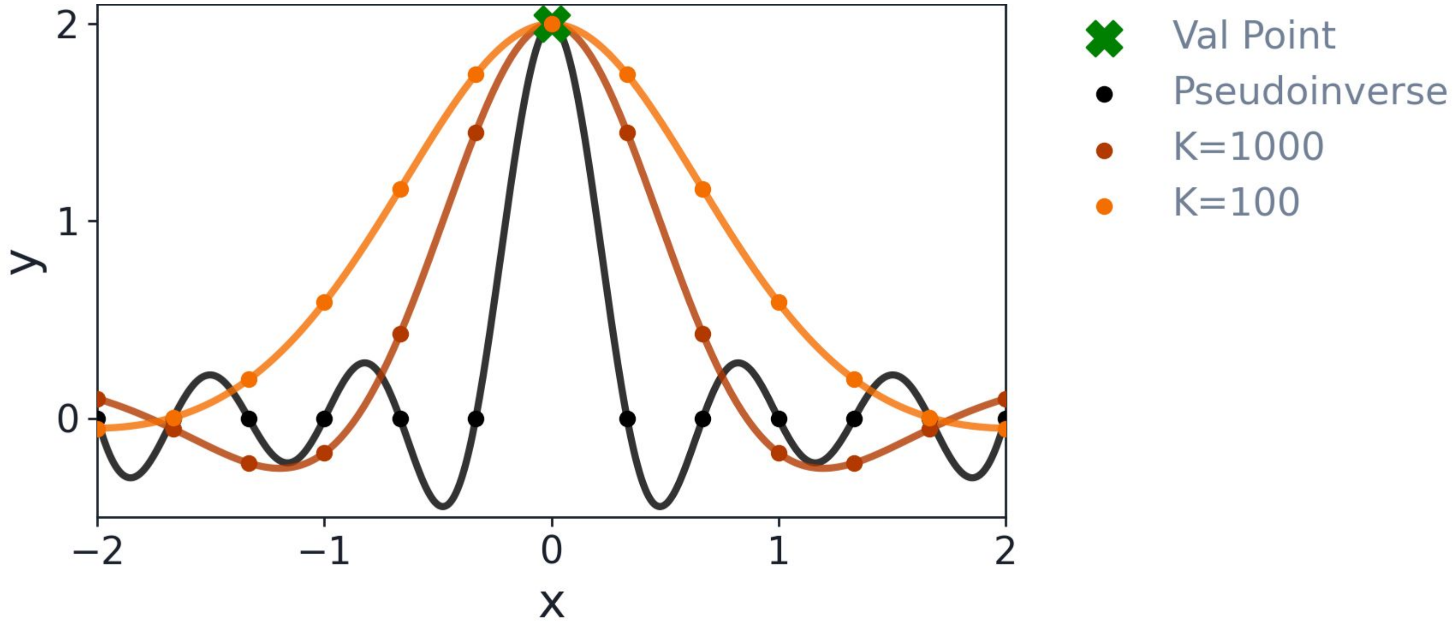
Outer Overparameterization: Anti-Distillation

- **Anti-distillation:** *more learned datapoints than original dataset examples*
- The *quality of hypergradient approximations induces a trade-off between the inner and outer parameter norms*—e.g., we can achieve the good performance for the outer objective by either making larger updates to the inner or the outer parameters



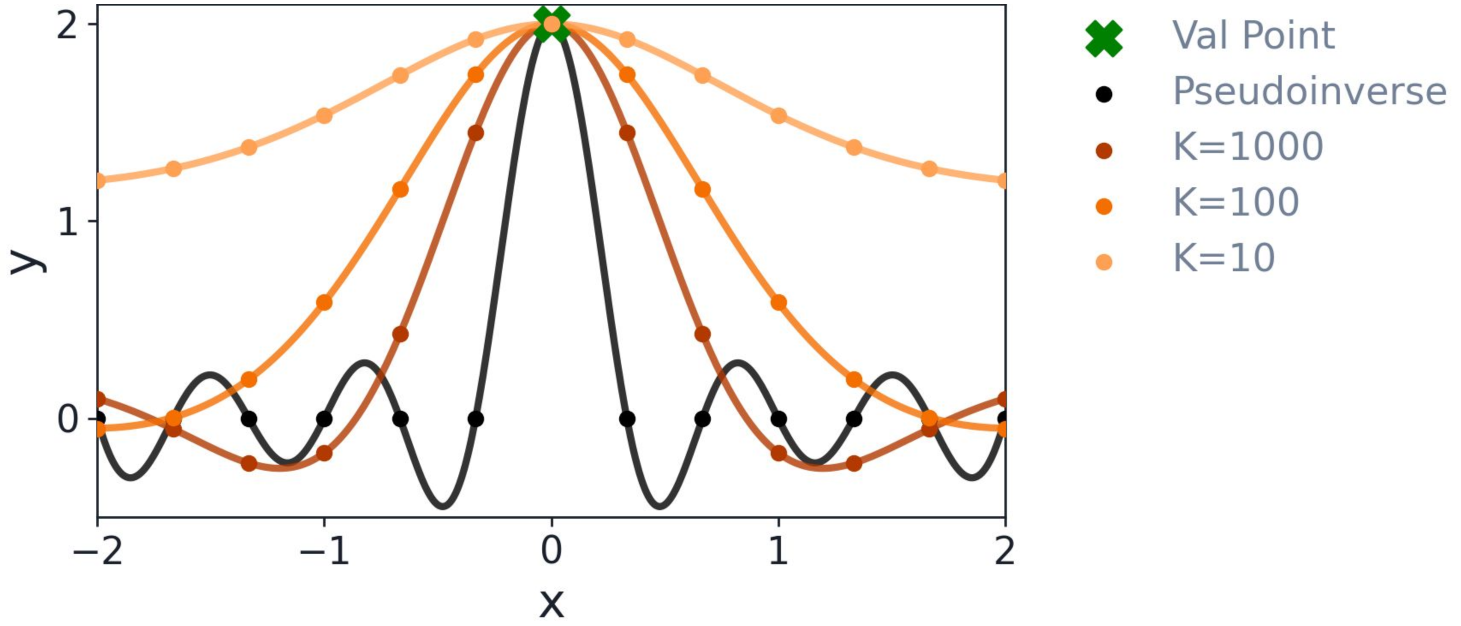
Outer Overparameterization: Anti-Distillation

- **Anti-distillation:** *more learned datapoints than original dataset examples*
- The *quality of hypergradient approximations induces a trade-off between the inner and outer parameter norms*—e.g., we can achieve the good performance for the outer objective by either making larger updates to the inner or the outer parameters



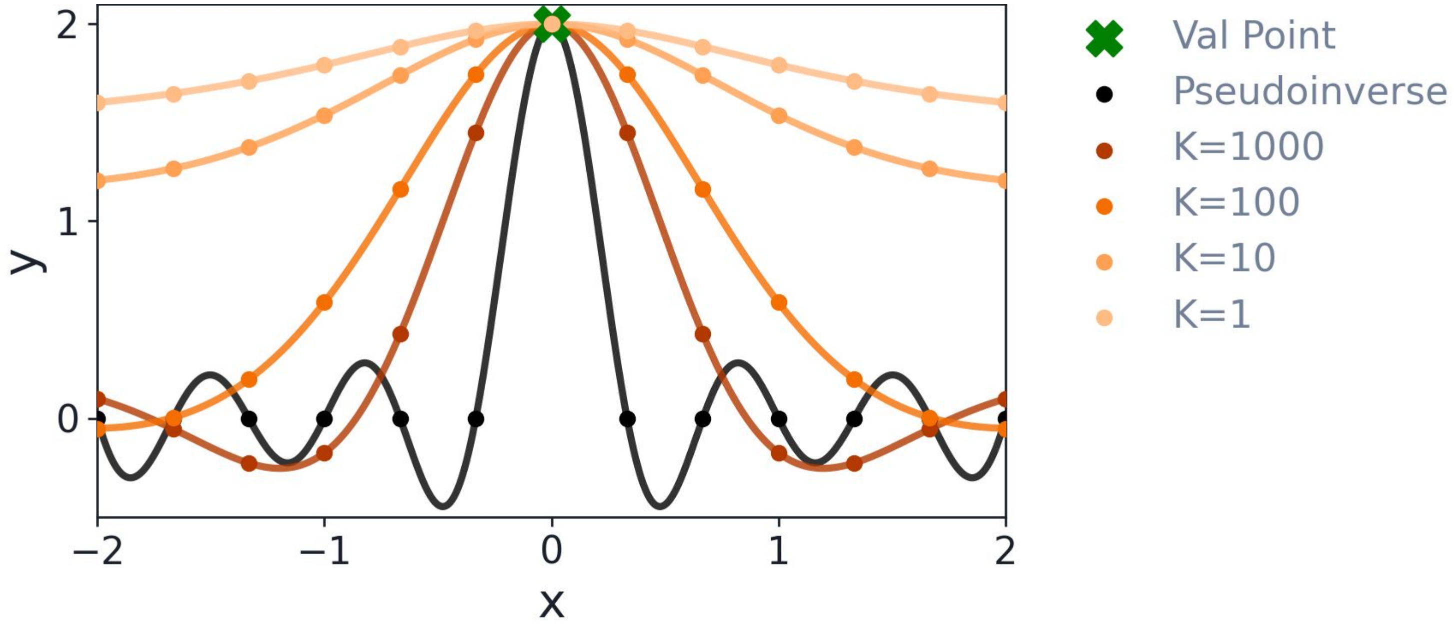
Outer Overparameterization: Anti-Distillation

- **Anti-distillation:** *more learned datapoints than original dataset examples*
- The *quality of hypergradient approximations induces a trade-off between the inner and outer parameter norms*—e.g., we can achieve the good performance for the outer objective by either making larger updates to the inner or the outer parameters



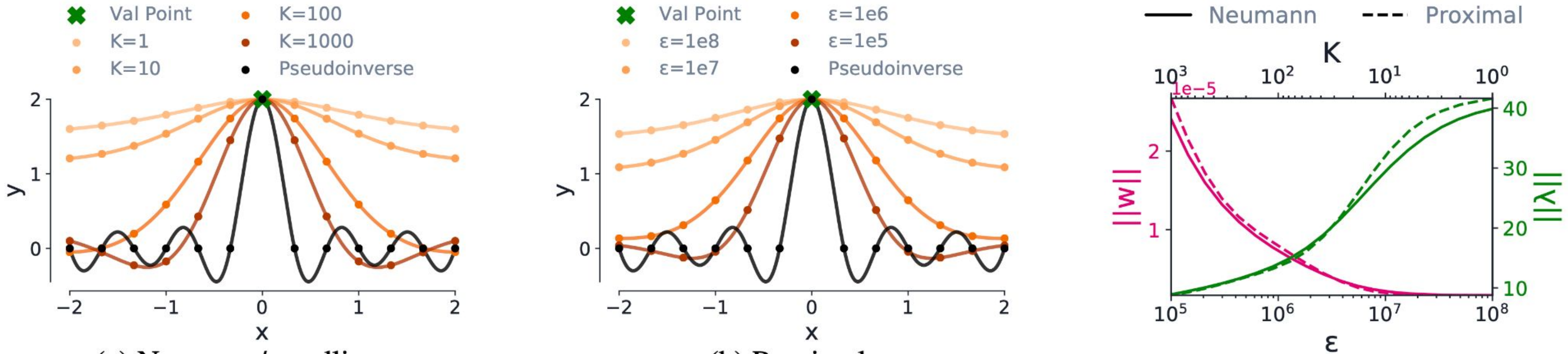
Outer Overparameterization: Anti-Distillation

- **Anti-distillation:** *more learned datapoints than original dataset examples*
- The *quality of hypergradient approximations induces a trade-off between the inner and outer parameter norms*—e.g., we can achieve the good performance for the outer objective by either making larger updates to the inner or the outer parameters



Outer Overparameterization: Anti-Distillation

- **Anti-distillation:** *more learned datapoints than original dataset examples*
- The *quality of hypergradient approximations induces a trade-off between the inner and outer parameter norms*—e.g., we can achieve the good performance for the outer objective by either making larger updates to the inner or the outer parameters



(a) Neumann/unrolling (b) Proximal (c) Parameter norms

Proximal Inner Objective

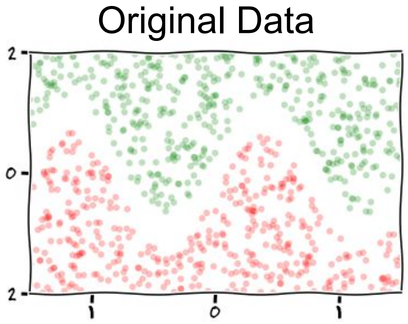
- We can formalize warm-started joint optimization by considering a *proximally regularized inner objective*:

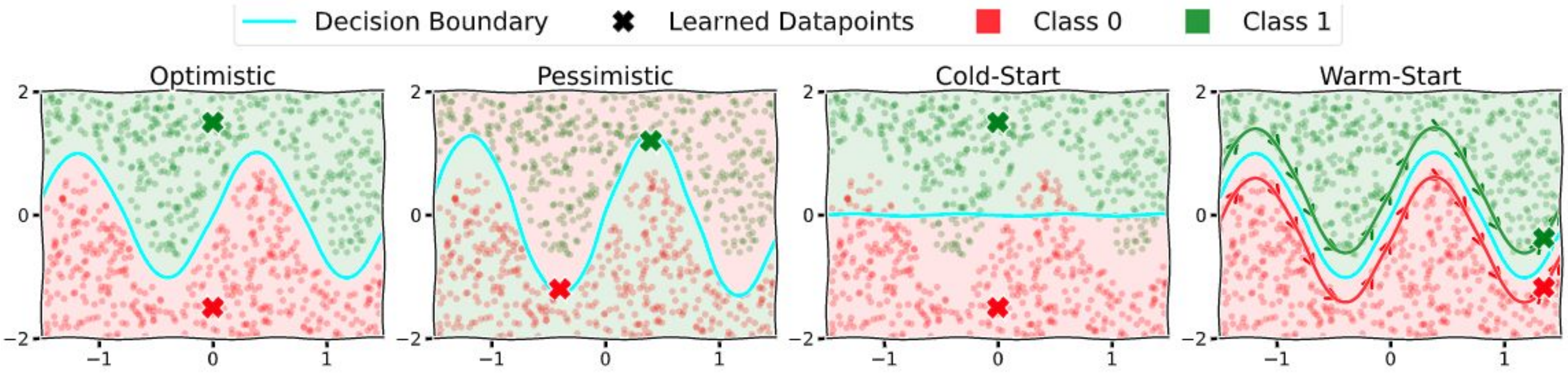
$$\mathbf{y}^* \in \arg\min_{\mathbf{y}} \{ f(\mathbf{x}, \mathbf{y}) + \frac{\epsilon}{2} ||\mathbf{y} - \mathbf{y}_k||^2 \}$$

- We define notions of *cold-start* and *warm-start* equilibria, which correspond to different solutions we obtain with different algorithms

	Cold-Start	**Warm-Start**
Update	$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \frac{\partial F}{\partial \mathbf{y}^*} \frac{\partial \mathbf{y}^*}{\partial \mathbf{x}}$ $\mathbf{y}^*_{t+1} \in \arg\min_{\mathbf{y} \in \mathcal{S}(\mathbf{x}_{t+1})} \|\|\mathbf{y} - \mathbf{y}_0\|\|^2$	$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \frac{\partial F}{\partial \mathbf{y}^*_t} \frac{\partial \mathbf{y}^*_t}{\partial \mathbf{x}}$ $\mathbf{y}^*_{t+1} \in \arg\min_{\mathbf{y}} \{ f(\mathbf{x}_{t+1}, \mathbf{y}) + \frac{\epsilon}{2} \|\|\mathbf{y} - \mathbf{y}_t\|\|^2 \}$
Response Jacobian	$\left(\frac{\partial^2 f}{\partial \mathbf{y} \partial \mathbf{y}^\top} \right)^{-1} \frac{\partial^2 f}{\partial \mathbf{y} \partial \mathbf{x}}$	$\left(\frac{\partial^2 f}{\partial \mathbf{y} \partial \mathbf{y}^\top} + \epsilon I \right)^{-1} \frac{\partial^2 f}{\partial \mathbf{y} \partial \mathbf{x}}$
Neumann Approx.	$H^{-1} \approx \sum_{k=0}^{K} (I - H)^k$	$(H + \epsilon I)^{-1} \approx \sum_{k=0}^{K} ((1 - \epsilon) I - H)^k$

Revisiting Overparam Bilevel Solutions





$$\arg\min_{\mathbf{y}\in\mathcal{S}(\mathbf{x})} F(\mathbf{x},\mathbf{y}) \qquad \arg\max_{\mathbf{y}\in\mathcal{S}(\mathbf{x})} F(\mathbf{x},\mathbf{y}) \qquad \arg\min_{\mathbf{y}\in\mathcal{S}(\mathbf{x})} ||\mathbf{y}-\mathbf{y}_0||_2^2 \qquad \arg\min_{\mathbf{y}\in\mathcal{S}(\mathbf{x})} ||\mathbf{y}-\mathbf{y}_t||_2^2$$

Summary

- In overparameterized bilevel optimization, *the inner and outer problems may admit non-unique solutions*
- We discussed different optimization algorithms: *warm-start* and *cold-start*
- We introduced *synthetic tasks illustrating the effects of hypergradient approximations* and overparameterization in the inner and outer problems
  - Distillation & anti-distillation
- We provided evidence for a *trade-off in the norms of inner and outer parameters*, that depends on the *hypergradient approximation used*

Thank you!