A. Linear Regression Example

Here, we present a simple example of optimizing a collection of quadratic objectives (equivalent to linear regression on fixed set of features), where the solutions to joint training and the meta-learning (MAML) problem are different. The purpose of this example is to primarily illustrate that meta-learning can provide performance gains even in seemingly simple and restrictive settings. Consider a collection of objective functions: $\{f_i : \mathbf{w} \in \mathbb{R}^d \to \mathbb{R}\}_{i=1}^M$ which can be described by quadratic forms. Specifically, each of these functions are of then form

$$f_i(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \boldsymbol{A}_i \mathbf{w} + \mathbf{w}^T \boldsymbol{b}_i.$$

This can represent linear regression problems as follows: let $(\mathbf{x}_{\mathcal{T}_i}, \mathbf{y}_{\mathcal{T}_i})$ represent input-output pairs corresponding to task $\mathcal{T}_i$. Let the predictive model be $\boldsymbol{h}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Here, we assume that a constant scalar (say 1) is concatenated in $\mathbf{x}$ to subsume the constant offset term (as common in practice). Then, the loss function can be written as:

$$f_i(\mathbf{w}) = \frac{1}{2}\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{T}_i}\left[||\boldsymbol{h}(\mathbf{x}) - \mathbf{y}||^2\right]$$

which corresponds to having $\boldsymbol{A}_i = \mathbb{E}_{\mathbf{x}\sim\mathcal{T}_i}[\mathbf{x}\mathbf{x}^T]$ and $\boldsymbol{b}_i = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{T}_i}[\mathbf{x}^T\mathbf{y}]$. For these set of problems, we are interested in studying the difference between joint training and meta-learning.

Joint training The first approach of interest is joint training which corresponds to the optimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}), \text{ where } F(\mathbf{w}) = \frac{1}{M}\sum_{i=1}^M f_i(\mathbf{w}). \tag{6}$$

Using the form of f_i, we have

$$F(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\left(\frac{1}{M}\sum_{i=1}^M \boldsymbol{A}_i\right)\mathbf{w} + \mathbf{w}^T\left(\frac{1}{M}\sum_{i=1}^M \boldsymbol{b}_i\right).$$

Let us define the following:

$$\bar{\boldsymbol{A}} := \frac{1}{M}\sum_{i=1}^M \boldsymbol{A}_i \;\text{ and }\; \bar{\boldsymbol{b}} := \frac{1}{M}\sum_{i=1}^M \boldsymbol{b}_i.$$

The solution to the joint training optimization problem (Eq. 6) is then given by $\mathbf{w}^*_{\text{joint}} = -\bar{\boldsymbol{A}}^{-1}\bar{\boldsymbol{b}}$.

Meta learning (MAML) The second approach of interest is meta-learning, which as mentioned in Section 2.2 corresponds to the optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \tilde{F}(\mathbf{w}), \text{ where } \tilde{F}(\mathbf{w}) = \frac{1}{M}\sum_{i=1}^M f_i(\boldsymbol{U}_i(\mathbf{w})). \tag{7}$$

Here, we specifically concentrate on the 1-step (exact) gradient update procedure: $\boldsymbol{U}_i(\mathbf{w}) = \mathbf{w} - \alpha\nabla f_i(\mathbf{w})$. In the case of the quadratic objectives, this leads to:

$$\begin{aligned} f_i(\boldsymbol{U}_i(\mathbf{w})) = \frac{1}{2}&(\mathbf{w} - \alpha\boldsymbol{A}_i\mathbf{w} - \alpha\boldsymbol{b}_i)^T\boldsymbol{A}_i(\mathbf{w} - \alpha\boldsymbol{A}_i\mathbf{w} - \alpha\boldsymbol{b}_i) \\ &+ (\mathbf{w} - \alpha\boldsymbol{A}_i\mathbf{w} - \alpha\boldsymbol{b}_i)^T\boldsymbol{b}_i \end{aligned}$$

The corresponding gradient can be written as:

$$\begin{aligned} \nabla f_i(\boldsymbol{U}_i(\mathbf{w})) &= \Big(\boldsymbol{I} - \alpha\boldsymbol{A}_i\Big)\Big(\boldsymbol{A}_i\big(\mathbf{w} - \alpha\boldsymbol{A}_i\mathbf{w} - \alpha\boldsymbol{b}_i\big) + \boldsymbol{b}_i\Big) \\ &= \big(\boldsymbol{I} - \alpha\boldsymbol{A}_i\big)\boldsymbol{A}_i\big(\boldsymbol{I} - \alpha\boldsymbol{A}_i\big)\mathbf{w} + \big(\boldsymbol{I} - \alpha\boldsymbol{A}_i\big)^2\boldsymbol{b}_i \end{aligned}$$

For notational convenience, we define:

$$\begin{aligned} \boldsymbol{A}_\dagger &:= \frac{1}{M}\sum_{i=1}^M \big(\boldsymbol{I} - \alpha\boldsymbol{A}_i\big)^2\boldsymbol{A}_i \\ \boldsymbol{b}_\dagger &:= \frac{1}{M}\sum_{i=1}^M \big(\boldsymbol{I} - \alpha\boldsymbol{A}_i\big)^2\boldsymbol{b}_i. \end{aligned}$$

Then, the solution to the MAML optimization problem (Eq. 7) is given by $\mathbf{w}^*_{\text{MAML}} = -\boldsymbol{A}_\dagger^{-1}\boldsymbol{b}_\dagger$.

Remarks In general, $\mathbf{w}^*_{\text{joint}} \neq \mathbf{w}^*_{\text{MAML}}$ based on our analysis. Note that $\boldsymbol{A}_\dagger$ is a weighed average of different $\boldsymbol{A}_i$, but the weights themselves are a function of $\boldsymbol{A}_i$. The reason for the difference between $\mathbf{w}^*_{\text{joint}}$ and $\mathbf{w}^*_{\text{MAML}}$ is the difference in moments of input distributions. The two solutions, $\mathbf{w}^*_{\text{joint}}$ and $\mathbf{w}^*_{\text{MAML}}$, coincide when $\boldsymbol{A}_i = \boldsymbol{A}\;\forall i$. Furthermore, since $\mathbf{w}^*_{\text{MAML}}$ was optimized to explicitly minimize $\tilde{F}(\cdot)$, it would lead to better performance after task-specific adaptation.

This example and analysis reveals that there is a clear separation in performance between joint training and meta-learning even in the case of quadratic loss functions. Improved performance with meta-learning approaches have been noted empirically with non-convex loss landscapes induced by neural networks. Our example illustrates that meta learning can provide non-trivial gains over joint training even in simple convex loss landscapes.

B. Proof of Theorem 1

In this section, we restate Theorem 1 and provide a proof.

Theorem 1. *Suppose f and $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ satisfy assumptions 1 and 2. Let $\tilde{f}$ be the function evaluated after a one step gradient update procedure, i.e.*

$$\tilde{f}(\mathbf{w}) := f\big(\mathbf{w} - \alpha\nabla\hat{f}(\mathbf{w})\big).$$

If the step size is selected as $\alpha \leq \min\{\frac{1}{2\beta}, \frac{\mu}{8\rho G}\}$, then $\tilde{f}$ is convex. Furthermore, it is also $\tilde{\beta} = 9\beta/8$ smooth and $\tilde{\mu} = \mu/8$ strongly convex.

Proof. First, the smoothness and strong convexity of f and $\hat{f}$ implies $\mu \leq ||\nabla^2 \hat{f}(\boldsymbol{\theta})|| \leq \beta \; \forall \boldsymbol{\theta}$. Thus,

$$(1 - \alpha\beta) \leq ||\boldsymbol{I} - \alpha\nabla^2 \hat{f}(\boldsymbol{\theta})|| \leq (1 - \alpha\mu) \;\; \forall \boldsymbol{\theta}.$$

Also recall the earlier notation $\tilde{\boldsymbol{\theta}} = \boldsymbol{U}(\boldsymbol{\theta}) = \boldsymbol{\theta} - \alpha\nabla\hat{f}(\boldsymbol{\theta})$. For $\alpha < 1/\beta$, we have the following bounds:

$$\begin{aligned}(1 - \alpha\beta)||\boldsymbol{\theta} - \boldsymbol{\phi}|| &\leq ||\boldsymbol{U}(\boldsymbol{\theta}) - \boldsymbol{U}(\boldsymbol{\phi})|| \qquad \forall(\boldsymbol{\theta}, \boldsymbol{\phi}) \\ ||\boldsymbol{U}(\boldsymbol{\theta}) - \boldsymbol{U}(\boldsymbol{\phi})|| &\leq (1 - \alpha\mu)||\boldsymbol{\theta} - \boldsymbol{\phi}|| \qquad \forall(\boldsymbol{\theta}, \boldsymbol{\phi}),\end{aligned}$$

since we have $\boldsymbol{U}(\boldsymbol{\theta}) - \boldsymbol{U}(\boldsymbol{\phi}) = \big(\boldsymbol{I} - \alpha\nabla^2\hat{f}(\boldsymbol{\psi})\big)(\boldsymbol{\theta} - \boldsymbol{\phi})$ for some $\boldsymbol{\psi}$ that connects $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ due to the mean value theorem on $\nabla\hat{f}$. Using the chain rule and our definitions,

$$\begin{aligned}&\nabla\tilde{f}(\boldsymbol{\theta}) - \nabla\tilde{f}(\boldsymbol{\phi}) = \nabla\boldsymbol{U}(\boldsymbol{\theta})\nabla f(\tilde{\boldsymbol{\theta}}) - \nabla\boldsymbol{U}(\boldsymbol{\phi})\nabla f(\tilde{\boldsymbol{\phi}}) \\ &= \left(\nabla\boldsymbol{U}(\boldsymbol{\theta}) - \nabla\boldsymbol{U}(\boldsymbol{\phi})\right)\nabla f(\tilde{\boldsymbol{\theta}}) + \nabla\boldsymbol{U}(\boldsymbol{\phi})\left(\nabla f(\tilde{\boldsymbol{\theta}}) - \nabla f(\tilde{\boldsymbol{\phi}})\right)\end{aligned}$$

Taking the norm on both sides, for the specified α, we have:

$$\begin{aligned}||\nabla\tilde{f}(\boldsymbol{\theta}) - \nabla\tilde{f}(\boldsymbol{\phi})|| &\leq ||\left(\nabla\boldsymbol{U}(\boldsymbol{\theta}) - \nabla\boldsymbol{U}(\boldsymbol{\phi})\right)\nabla f(\tilde{\boldsymbol{\theta}})|| \\ &\quad + ||\nabla\boldsymbol{U}(\boldsymbol{\phi})\left(\nabla f(\tilde{\boldsymbol{\theta}}) - \nabla f(\tilde{\boldsymbol{\phi}})\right)|| \\ &\leq (\alpha\rho G + (1 - \alpha\mu)^2\beta)||\boldsymbol{\theta} - \boldsymbol{\phi}|| \\ &\leq \left(\frac{\mu}{8} + \beta\right)||\boldsymbol{\theta} - \boldsymbol{\phi}|| \\ &\leq \frac{9\beta}{8}||\boldsymbol{\theta} - \boldsymbol{\phi}||.\end{aligned}$$

Similarly, we obtain the following lower bound

$$\begin{aligned}||\nabla\tilde{f}(\boldsymbol{\theta}) - \nabla\tilde{f}(\boldsymbol{\phi})|| &\geq ||\nabla\boldsymbol{U}(\boldsymbol{\phi})\left(\nabla f(\tilde{\boldsymbol{\theta}}) - \nabla f(\tilde{\boldsymbol{\phi}})\right)|| \\ &\quad - ||\left(\nabla\boldsymbol{U}(\boldsymbol{\theta}) - \nabla\boldsymbol{U}(\boldsymbol{\phi})\right)\nabla f(\tilde{\boldsymbol{\theta}})|| \\ &\geq (1 - \alpha\beta)^2\mu||\boldsymbol{\theta} - \boldsymbol{\phi}|| - \alpha\rho G||\boldsymbol{\theta} - \boldsymbol{\phi}|| \\ &\geq \left(\frac{\mu}{4} - \frac{\mu}{8}\right)||\boldsymbol{\theta} - \boldsymbol{\phi}|| \\ &\geq \frac{\mu}{8}||\boldsymbol{\theta} - \boldsymbol{\phi}||\end{aligned}$$

which completes the proof. □

C. Proof of Corollary 2

In this section, we restate Corollary 2 and provide a proof.

Corollary 2. *(inherited regret bound for FTML) Suppose that for all t, f_t and $\hat{f}_t$ satisfy assumptions 1 and 2. Suppose that the update procedure in FTML (Eq. 4) is chosen as $\boldsymbol{U}_t(\mathbf{w}) = \mathbf{w} - \alpha\nabla\hat{f}_t(\mathbf{w})$ with $\alpha \leq \min\{\frac{1}{2\beta}, \frac{\mu}{8\rho G}\}$. Then, FTML enjoys the following regret guarantee*

$$\sum_{t=1}^{T} f_t\big(\boldsymbol{U}_t(\mathbf{w}_t)\big) - \min_{\mathbf{w}} \sum_{t=1}^{T} f_t\big(\boldsymbol{U}_t(\mathbf{w})\big) = O\left(\frac{32G^2}{\mu}\log T\right)$$

Proof. From Theorem 1, we have that each function $\tilde{f}_t(\mathbf{w}) = f_t(\boldsymbol{U}_t(\mathbf{w}))$ is $\tilde{\mu} = \mu/8$ strongly convex. The FTML algorithm is identical to FTL on the sequence of loss functions $\{\tilde{f}_t\}_{t=1}^T$, which has a $O(\frac{4G^2}{\tilde{\mu}}\log T)$ regret guarantee (see Cesa-Bianchi & Lugosi (2006) Theorem 3.1). Using $\tilde{\mu} = \mu/8$ completes the proof. □

D. Additional Experimental Details

For all experiments, we trained our FTML method with 5 inner batch gradient descent steps with step size $\alpha = 0.1$. We use an inner batch size of 10 examples for MNIST and pose prediction and 25 datapoints for CIFAR. Except for the CIFAR NML experiments, we train the convolutional networks using the Adam optimizer with default hyperparameters (Kingma & Ba, 2015). We found Adam to be unstable on the CIFAR setting for NML and instead used SGD with momentum, with a momentum parameter of 0.9 and a learning rate of 0.01 for the first 5 thousand iterations, followed by a learning rate of 0.001 for the rest of learning.

For the MNIST and CIFAR experiments, we use the cross entropy loss, using label smoothing with $\epsilon = 0.1$ as proposed by Szegedy et al. (2016). We also use this loss for the inner loss in the FTML objective.

In the MNIST and CIFAR experiments, we use a convolutional neural network model with 5 convolution layers with 32 3×3 filters interleaved with batch normalization and ReLU nonlinearities. The output of the convolution layers is flattened and followed by a linear layer and a softmax, feeding into the output. In the pose prediction experiment, all models use a convolutional neural network with 4 convolution layers each with 16 5×5 filters. After the convolution layers, we use a spatial soft-argmax to extract learned feature points, an architecture that has previously been shown to be effective for spatial tasks (Levine et al., 2016; Singh et al., 2017). We then pass the feature points through 2 fully connected layers with 200 hidden units and a linear layer to the output. All layers use batch normalization and ReLU nonlinearities.