# TRANSFERRING KNOWLEDGE ACROSS LEARNING PROCESSES

Sebastian Flennerhag[a], Pablo G. Moreno[b], Neil D. Lawrence[b], Andreas Damianou[b]
[a]The Alan Turing Institute, [b]Amazon

## SUMMARY


- In complex transfer learning scenarios, information contained only in the final parameters of a source model may be insufficient; a higher-level of abstraction is needed.
- We present a framework based on the idea that transfer learning can be achieved by leveraging information across similar learning processes, encoded in the geometry of the loss surface.
- We propose Leap, a lightweight meta-learner that scales beyond few-shot learning. Leap outperforms competing methods from meta-learning and transfer learning across a variety complex transfer learning scenarios.


## SETUP: GRADIENT PATHS ON TASK MANIFOLDS

Given a learning objective $f$ that consumes and input $x$ and a target $y$ and maps a parameterization $\theta$ to a scalar loss value, we define a learning process by the gradient update rule

$$\theta^{i+1} = \theta^i - \alpha \nabla f(\theta^i). \tag{1}$$

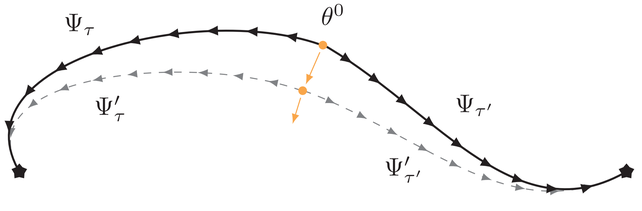
This process defines a *curve* $\gamma$ on a task-specific manifold:

$$\text{Length}(\gamma) = \int_0^1 \sqrt{\langle \dot\gamma(t), \dot\gamma(t) \rangle}\, dt. \tag{2}$$

- The choice of manifold determines how we view length. For simplicity, we define it as the loss surface: $\gamma(t) = (\theta(t), f(\theta(t)))$.
- A discrete approximation can be computed at negligible cost:

$$\text{Length}(\gamma) \approx \sum_{i=0}^{K-1} \|\gamma^{i+1} - \gamma^i\| = d(\theta^0). \tag{3}$$

Because the length of $\gamma$ summarizes a learning process, we transfer knowledge by minimizing the *expected gradient path length* in Eq. 3 across a distribution $p(\tau)$ of tasks with unique paths $\Psi_\tau$:




## META LEARNING ACROSS TASK MANIFOLDS

- We focus on meta-learning a shared initialization $\theta^0$.
- Gradient paths cannot differentiate between good and bad local minima: we need a feasibility constraint.
- Given a second-best initialization $\psi^0$, we aim to solve

$$\begin{aligned} \min_{\theta^0} \quad & F(\theta^0) = \mathbb{E}_{\tau\sim p(\tau)}\left[d_\tau(\theta^0)\right] \\ \text{s.t.} \quad & \theta^0 \in \Theta = \cap_\tau \left\{\theta^0 \mid f_\tau(\theta^\star_\tau) \le f_\tau(\psi^\star_\tau)\right\}. \end{aligned} \tag{4}$$

## LEAP

The feasibility constraint is costly to evaluate. Instead, we can use $\psi^0$ to generate baselines $\bar\gamma^i = (\psi^i, f(\psi^i))$ that guide $\theta^0$:

$$\bar d(\theta^0, \psi^0) = \sum_{i=0}^{K-1} \|\bar\gamma^{i+1} - \gamma\|. \tag{5}$$

*Leap* uses $\bar F(\theta^0, \psi^0) = \mathbb{E}_{\tau\sim p(\tau)}\left[\bar d_\tau(\theta^0, \psi^0)\right]$ to obtain a sequence of incrementally demanding baselines that minimizes Eq. 4:

**Theorem 1** (Pull-forward). *Define a sequence of initializations* $\{\psi^0_s\}_{s\in\mathbb{N}}$ *by*

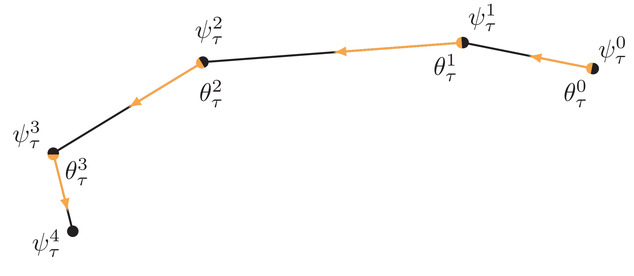
$$\psi^0_{s+1} = \psi^0_s - \beta_s \nabla \bar F(\psi^0_s, \psi^0_s), \qquad \psi^0_0 \in \Theta. \tag{6}$$

*For* $\beta_s > 0$ *sufficiently small, there exist learning rates schedules* $\{\alpha^i_\tau\}_{i=1}^{K_\tau}$ *for all tasks such that* $\psi^0_{k\to\infty}$ *is a limit point in* $\Theta$.

Crucially, the meta gradient can (approximately) be computed on the fly at negligible cost:

$$\nabla \bar F(\theta^0, \psi^0) \approx -\mathbb{E}_{\tau\sim p(\tau)}\left[\sum_{i=0}^{K_\tau - 1} \frac{\Delta f^i_\tau \, \nabla f_\tau(\theta^i_\tau) + \Delta\theta^i_\tau}{\left(\|\bar\gamma^i_\tau - \gamma^i_\tau\|_2^p\right)^{2-p}}\right]. \tag{7}$$

Leap pulls the initialization forward along known gradient paths to find an initialization with minimal expected gradient path length that is guaranteed to perform as well as the baseline:

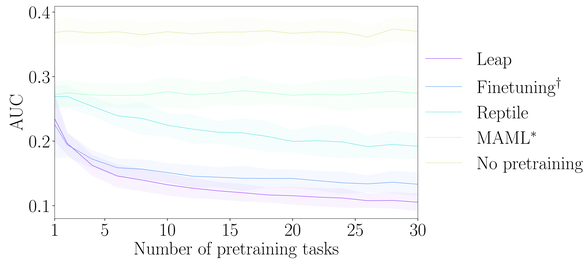



## ALGORITHM 1: LEAP

**Require:** $\beta_s$, $p(\tau)$, $\tau = (f_\tau, u_\tau, p_\tau)$: distribution over tasks
randomly initialize $\theta^0$
**while** not done **do**
  $\nabla \bar F \leftarrow 0$: initialize meta gradient
  sample task batch $\mathcal{B}$ from $p(\tau)$
  **for all** $\tau \in \mathcal{B}$ **do**
    $\psi^0_\tau \leftarrow \theta^0$: initialize task baseline
    **for all** $i \in \{0, \dots, K_\tau - 1\}$ **do**
      $\psi^{i+1}_\tau \leftarrow u_\tau(\psi^i_\tau)$: update baseline
      $\theta^i_\tau \leftarrow \psi^i_\tau$: follow baseline (recall $\psi^0_\tau = \theta^0$)
      increment $\nabla \bar F$ using the pull-forward gradient (Eq. 7)
    **end for**
  **end for**
  $\theta^0 \leftarrow \theta^0 - \frac{\beta}{|\mathcal{B}|}\nabla \bar F$: update initialization
**end while**

## EXPERIMENTS

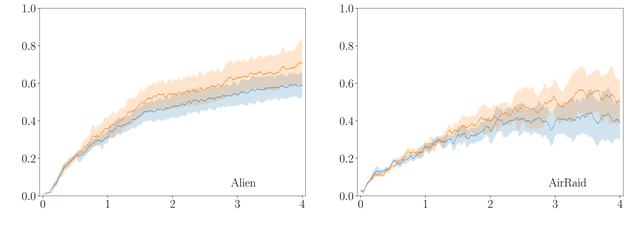
- Omniglot: each alphabet is a task, error AUC on test tasks:




- Multi-CV: a dataset is a task; mean normalized improvement:

| | Leap | Finetuning | Progressive Nets [3] | HAT [4] |
|---|---|---|---|---|
| AUC | **0.74** | 0.90 | 1.06 | 1.09 |
| Test Error | **0.89** | 1.20 | 0.97 | 1.15 |

- Atari: each game is a task; examples of Leap vs. random init:




## References


[1] C. Finn, P. Abbeel, S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. ICML, 2017.
[2] A. Nichol, J. Achiam, J. Schulman. On First-Order Meta-Learning Algorithms. arXiv:1803:02999, 2018.
[3] A. Rusu, N. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell. Progressive neural networks.
[4] J. Serra, D. Suris, M. Miron, A. Kratzoglou. Overcoming catastrophic forgetting with hard attention to the task. ICML, 2018.