Auxiliary Learning by Implicit Differentiation

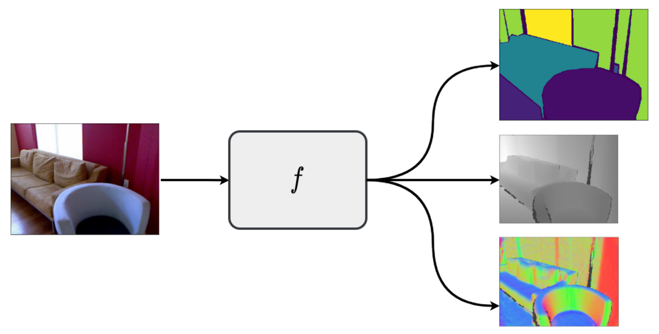
Aviv Navon*[1] Idan Achituve*[1] Haggai Maron[2] Gal Chechik†[1,2] Ethan Fetaya†[1]

[1]Bar-Ilan University, Israel [2]NVIDIA, Israel

Overview

Training neural networks with auxiliary tasks is a common practice for improving the performance on a main task of interest. Two main challenges arise in this multi-task learning setting: (i) designing useful auxiliary tasks; and (ii) combining auxiliary tasks into a single coherent loss. To tackle both challenges we propose *AuxiLearn*, a novel framework based on implicit differentiation.

Auxiliary Learning



- Aim at learning a main task of interest.
- Auxiliary tasks facilitate the learning of the main task.

AuxiLearn

- Optimization of two networks: primary network $f(\cdot\,;W)$, and auxiliary network $g(\cdot\,;\phi)$.
- Bi-level optimization:

$$\phi^* = \arg\min_{\phi} \mathcal{L}_A(W^*(\phi)), \quad \text{s.t.}$$

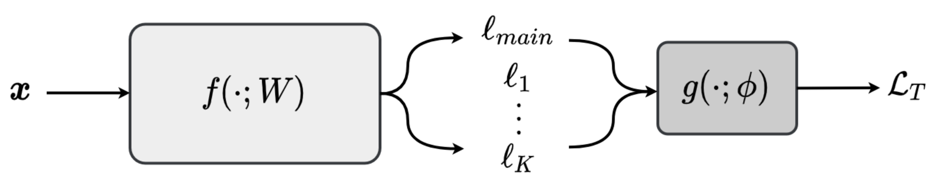
$$W^*(\phi) = \arg\min_{W} \mathcal{L}_T(W, \phi).$$

where, $\mathcal{L}_T = \sum_i \ell_{main}(\mathbf{x}_i, \boldsymbol{y}_i; W) + h(\mathbf{x}_i, \boldsymbol{y}_i, W; \phi)$ and $\mathcal{L}_A = \sum_i \ell_{main}(\mathbf{x}_i, \boldsymbol{y}_i; W)$. Here h is the overall auxiliary loss.

Goal: Find auxiliary parameters, ϕ, such that a network trained using ϕ will generalize well.

Solution: Utilizing IFT with efficient approximations.

Combining Losses



- Auxiliary tasks are given.
- AuxiLearn learns a deep auxiliary network over the losses. Here: $h(\cdot) = g(\boldsymbol{\ell}; \phi)$.
- **Key advantages:** (i) capture complex interactions between tasks; (ii) scales well with the number of tasks.
- **An Illustrative Example:**

(a) main task (b) $t = 0$ (c) $t = T$

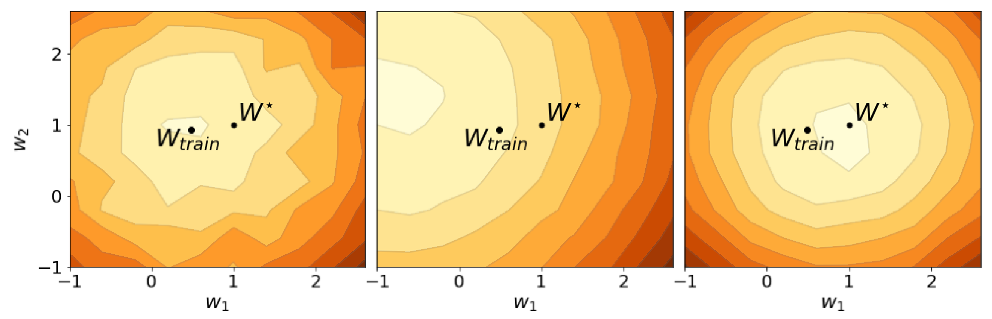
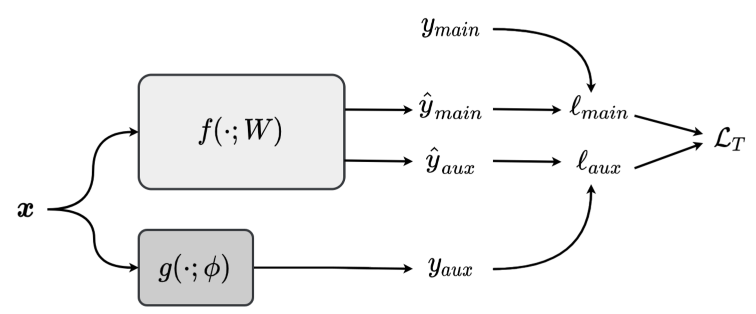


Figure: Loss landscape. Darker is higher.

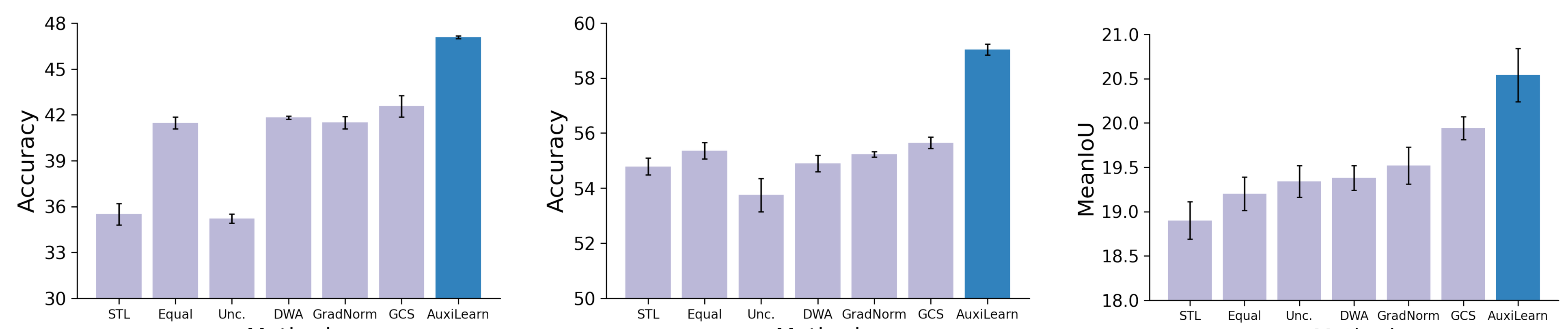
- ▶ A regression task with two auxiliaries: one helpful & one harmful.
- ▶ AuxiLearn learns to ignore the harmful auxiliary and uses the helpful one to find a better solution.

Learning Auxiliary Tasks



- Often auxiliary tasks are not available.
- A teacher network g produces auxiliary labels.
- The primary network f is trained to predict the main and the learned auxiliary tasks.
- Using AuxiLearn we can generate auxiliary tasks. Here: $h(\cdot) = \ell_{aux}(f(\boldsymbol{x}; W), g(\boldsymbol{x}; \phi))$.

Results for Combining Losses



(a) CUB 5-shot (b) CUB 10-shot (c) NYU Segmentation

Figure: Results for CUB dataset with 5/10 labels per main class (left), and NYUv2 dataset (right).

In the CUB experiments:

- The main task: fine-grained classification of 200 bird species.
- Auxiliary tasks: 312 binary visual attributes, such as breast color and bill length.
- Few labels per class for the main task, and auxiliary information is available for the entire dataset.

In the NYUv2 experiments:

- The main task: semantic segmentation
- Auxiliary tasks: depth estimation and surface-normal prediction

Results for Learning Auxiliaries

	CIFAR10 (5%)	CIFAR100 (5%)	SVHN (5%)	CUB (30-shot)	Pet (30-shot)	Cars (30-shot)
STL	50.8 ± 0.8	19.8 ± 0.7	72.9 ± 0.3	37.2 ± 0.8	26.1 ± 0.5	59.2 ± 0.4
MAXL-F	56.1 ± 0.1	20.4 ± 0.6	75.4 ± 0.3	39.6 ± 1.3	26.2 ± 0.3	59.6 ± 1.1
MAXL	58.2 ± 0.3	21.0 ± 0.4	75.5 ± 0.4	40.7 ± 0.6	26.3 ± 0.6	60.4 ± 0.8
AuxiLearn	$\mathbf{60.7 \pm 1.3}$	$\mathbf{21.5 \pm 0.3}$	$\mathbf{76.4 \pm 0.2}$	$\mathbf{44.5 \pm 0.3}$	$\mathbf{37.0 \pm 0.6}$	$\mathbf{64.4 \pm 0.3}$

Table: Learning novel classification auxiliary tasks.

- Learning novel auxiliary tasks from multi-class classification and fine-grained classification datasets.
- Setup: Using a small subset of the labeled data and learning a different multi-class classification auxiliary task for each class of the main task.
- AuxiLearn outperforms all baselines in all setups by a large margin.

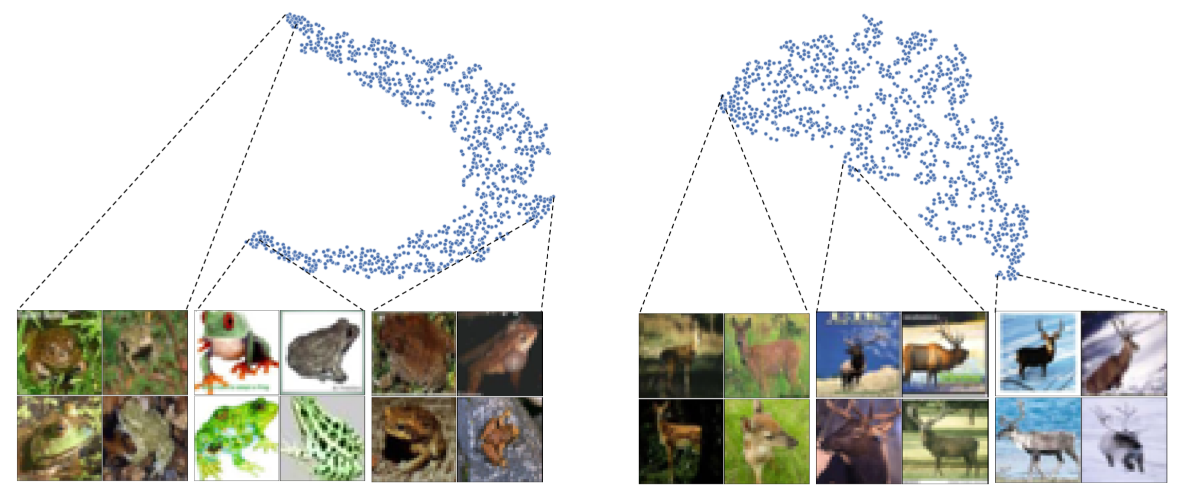


Figure: t-SNE over learned auxiliary labels.

- In the figure: 2D t-SNE projection of the learned labels for the classes *Frog* and *Deer*. AuxiLearn captures semantic features in the learned auxiliary labels.

Paper (arXiv) Project page Code (Github)