Learning the Pareto Front with Hypernetworks

Aviv Navon*[1] Aviv Shamsian*[1] Ethan Fetaya†[1] Gal Chechik†[1,2]

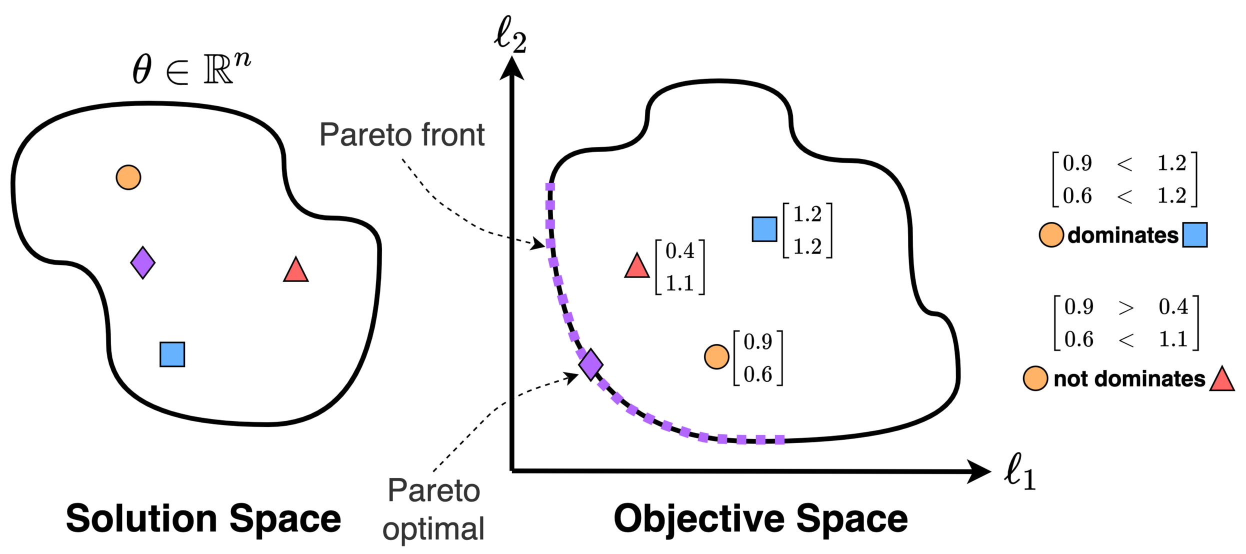
[1]Bar-Ilan University, Israel [2]NVIDIA, Israel

Overview

We put forward a novel multi-objective optimization (MOO) setup which we term *Pareto Front Learning* (PFL): Learning the Pareto front using a single model that can be applied to any objective preference at inference time.

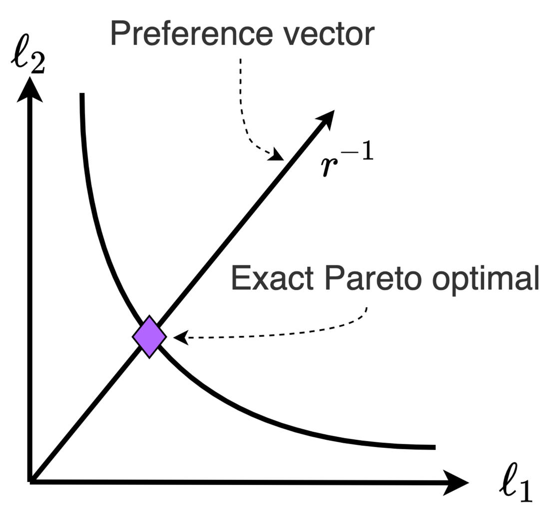
We propose *Pareto Hypernetworks* (PHN), a model for this setup based on hypernetworks.

Multi-objective Optimization



Given losses $\ell_1, ..., \ell_m$, a solution θ_1 *dominates* a solution θ_2 if θ_1 is not worse on any loss, and improves at least one ℓ_i. A solution is called *Pareto optimal* if it is not dominated. The set of all optimal solutions is called the *Pareto front*.

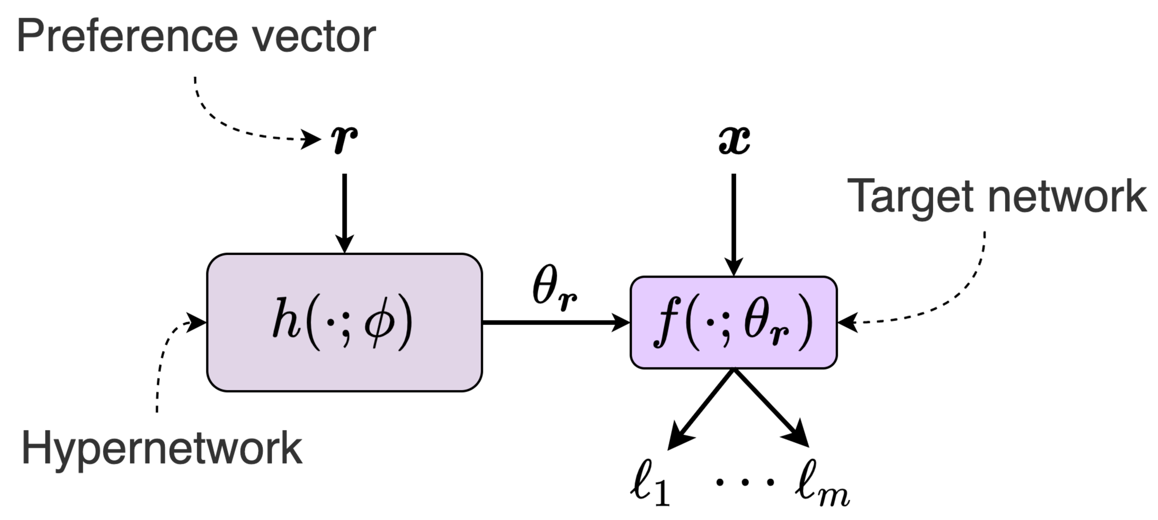
Each optimal solution is an intersection between the front and a *preference vector*. A Pareto optimal solution that lies on the preference vector is called *Exact Pareto Optimal*.



Pareto Hypernetworks

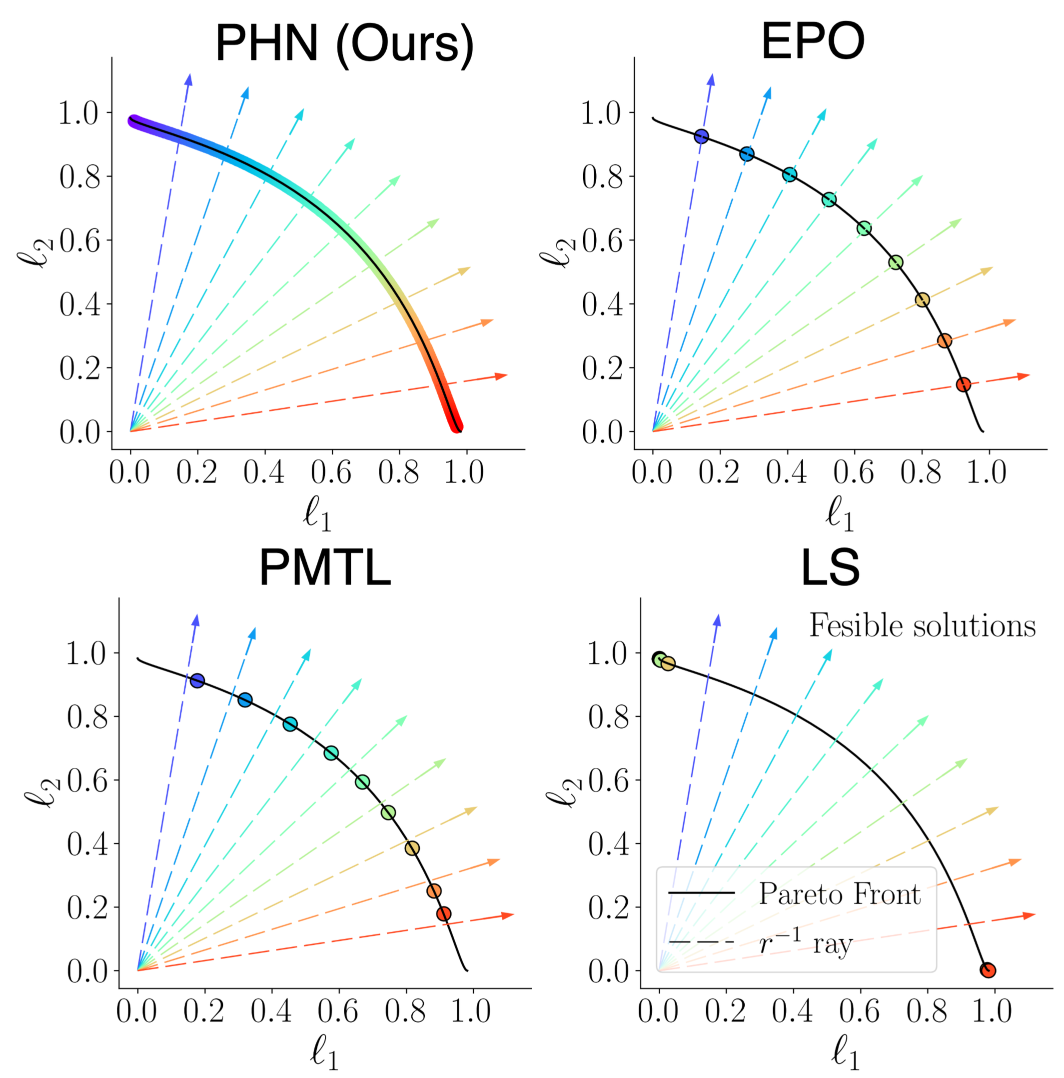
Hypernetworks are deep models that generate the weights of another (target) network.

Our hypernetwork h produces weights θ_r for a given input preference vector $\boldsymbol{r}$. θ_r is trained to be exact Pareto optimal w.r.t. $\boldsymbol{r}$.



Advantages: (i) *Scalability*: A single model covers the front; (ii) *Flexibility*: A user can switch between trade-off points during inference.

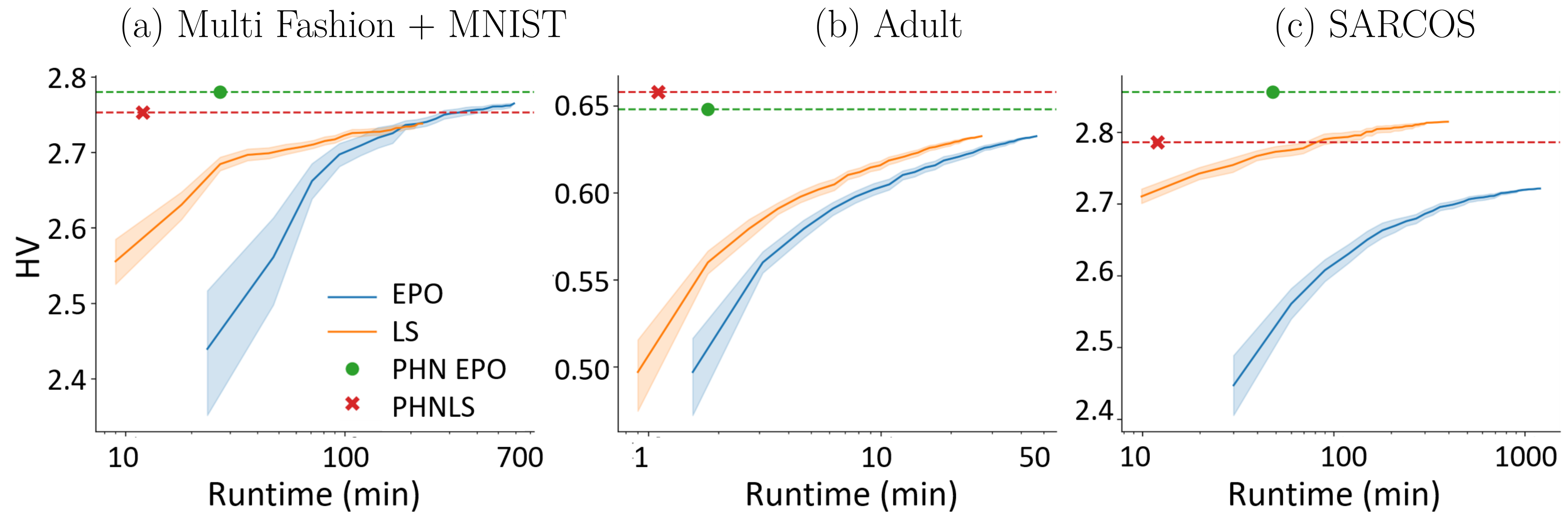
An Illustrative Example: Pareto front (black solid line) for a 2D loss space. Each colored dashed line ("ray") represents a possible preferences.



Top left: A single PHN model learns the entire Pareto front, mapping any given preference ray to its corresponding optimal solution.

Quality-Runtime Trade-off

Baseline models need multiple models to cover the front, yielding a trade-off between solution quality and overall runtime. PHN training takes nearly the same time as a single model, achieves superior or comparable quality (hypervolumne) as 25-40 baseline models, and is also 10 $\sim$ 50 times faster.



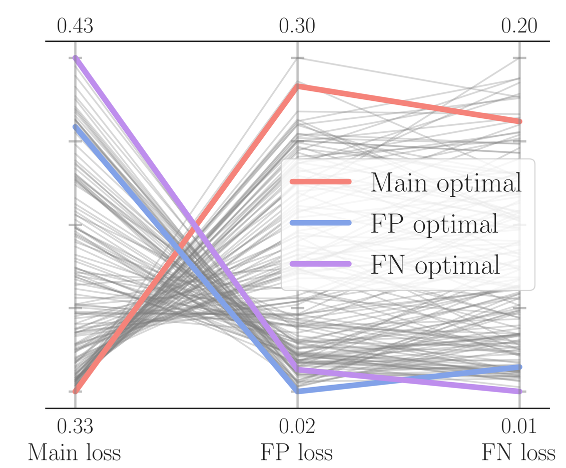
Results

	HV ⇑	Run-time (hours, Tesla V100)
	NYUv2	
LS	3.550	$0.58 \times 5 = 2.92$
PMTL	3.554	$0.96 \times 5 = 4.79$
CPMTL	3.570	$0.71 \times 5 = 3.55$
EPO	3.266	$1.02 \times 5 = 5.11$
PHN-LS (ours)	3.546	**0.67**
PHN-EPO (ours)	**3.589**	1.04

	HV ⇑			Run-time (min., Tesla V100)
	Multi-Fashion+MNIST	Multi-Fashion	Multi-MNIST	
LS	2.70	2.14	2.85	$9.0 \times 5 = 45$
CPMTL	2.76	2.16	2.88	$10.2 \times 5 = 51$
PMTL	2.67	2.13	2.86	$17.0 \times 5 = 85$
EPO	2.67	2.15	2.85	$23.6 \times 5 = 118$
PHN-LS (ours)	2.75	**2.19**	**2.90**	**12**
PHN-EPO (ours)	**2.78**	**2.19**	2.78	27

Modeling Conflicting Objectives

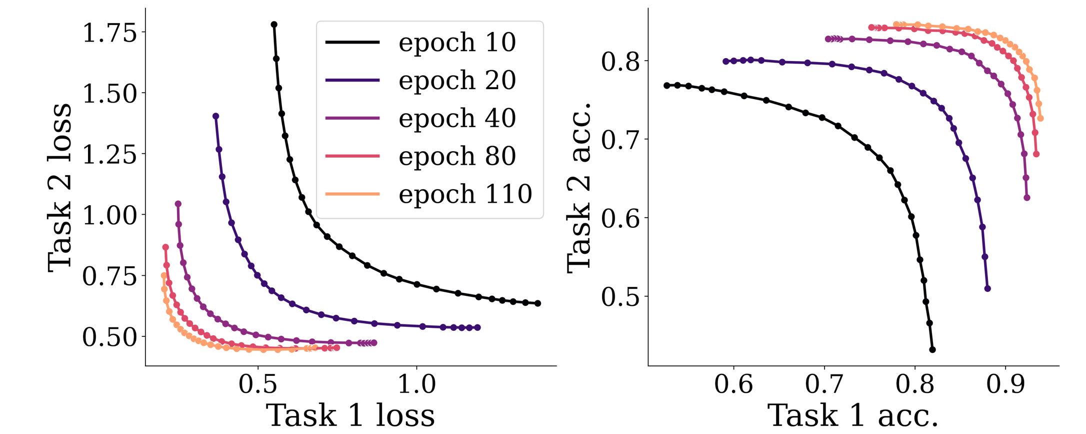
PHN generates models across the entire *accuracy-fairness* trade-off curve for the *Adult* dataset.



Paper (arXiv)

Code (Github)

PHN Training



Evaluation of the learned Pareto front and corresponding accuracies thought PHN training process, over the Multi-Fashion + MNIST test set.