Pareto Navigation Gradient Descent: a First-Order Algorithm for Optimization in Pareto Set

Mao Ye[1] **Qiang Liu**[1]

[1]Computer Science Dept., The University of Texas at Austin.

A THEORETICAL ANALYSIS

Theorem 1 [Dual of Equation (7)] *The solution v_t of Equation (7), if it exists, has a form of*

$$v_t = \nabla F(\theta_t) + \sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta_t),$$

with $\{\lambda_{i,t}\}_{i=1}^m$ the solution of the following dual problem

$$\max_{\lambda \in \mathbb{R}_+^m} -\frac{1}{2} \left\| \nabla F(\theta_t) + \sum_{i=1}^m \lambda_t \nabla \ell_i(\theta_t) \right\|^2 + \sum_{i=1}^m \lambda_i \phi_t,$$

where $\mathbb{R}_+^m$ is the set of nonnegative m-dimensional vectors, that is, $\mathbb{R}_+^m = \{\lambda \in \mathbb{R}^m : \lambda_i \geq 0, \ \forall i \in [m]\}$.

Proof. By introducing Lagrange multipliers, the optimization in Equation (7) is equivalent to the following minimax problem:

$$\min_{v \in \mathbb{R}^n} \max_{\lambda \in \mathbb{R}_+^m} \frac{1}{2} \left\| \nabla F(\theta_t) - v \right\|^2 + \sum_{i=1}^m \lambda_i \left(\phi_t - \nabla \ell_i(\theta_t)^\top v \right).$$

With strong duality of convex quadratic programming (assuming the primal problem is feasible), we can exchange the order of min and max, yielding

$$\max_{\lambda \in \mathbb{R}_+^m} \left\{ \Phi(\lambda) := \min_{v \in \mathbb{R}^n} \frac{1}{2} \left\| \nabla F(\theta_t) - v \right\|^2 + \sum_{i=1}^m \lambda_i \left(\phi_t - \nabla \ell_i(\theta_t)^\top v \right) \right\}.$$

It is easy to see that the minimization w.r.t. v is achieved when $v = \nabla F(\theta_t) + \sum_{i=1}^m \lambda_i \nabla \ell_i(\theta_t)$. Correspondingly, the $\Phi(\lambda)$ has the following dual form:

$$\max_{\lambda \in \mathbb{R}_+^m} -\frac{1}{2} \left\| \nabla F(\theta_t) + \sum_{i=1}^m \lambda_i \nabla \ell_i(\theta_t) \right\|^2 + \sum_{i=1}^m \lambda_i \phi_t.$$

This concludes the proof. □

Theorem 2 [Pareto Improvement on ℓ] *Under Assumption 1, assume $\theta_0 \notin \mathcal{P}_e$, and t_e is the first time when $\theta_{t_e} \in \mathcal{P}_e$, then for any time $t < t_e$,*

$$\frac{\mathrm{d}}{\mathrm{d}t} \ell_i(\theta_t) \leq -\alpha_t g(\theta_t), \qquad \min_{s \in [0,t]} g(\theta_t) \leq \frac{\min_{i \in [m]} (\ell_i(\theta_0) - \ell_i^*)}{\int_0^t \alpha_s \mathrm{d}s}.$$

Therefore, the update yields Pareto improvement on $\boldsymbol{\ell}$ when $\theta_t \notin \mathcal{P}_e$ and $\alpha_t g(\theta_t) > 0$.
Further, if $\int_0^t \alpha_s \mathrm{d}s = +\infty$, then for any $\epsilon > e$, there exists a finite time $t_\epsilon \in \mathbb{R}_+$ on which the solution enters $\mathcal{P}_\epsilon$ and stays within $\overline{\mathcal{P}}_\epsilon$ afterwards, that is, we have $\theta_{t_\epsilon} \in \mathcal{P}_\epsilon$ and $\theta_t \in \overline{\mathcal{P}}_\epsilon$ for any $t \geq t_\epsilon$.

Proof. i) When $t < t_e$, we have $g(\theta_t) > e$ and hence

$$\frac{\mathrm{d}}{\mathrm{d}t}\ell_i(\theta_t) = -\nabla \ell_i(\theta_t)^\top v_t \leq -\phi_t = -\alpha_t g(\theta_t), \tag{1}$$

where we used the constraint of $\nabla \ell_i(\theta_t)^\top v_t \geq \phi_t$ in Equation (7). Therefore, we yield strict decent on all the losses $\{\ell_i\}$ when $\alpha_t g(\theta_t) > 0$.

ii) Integrating both sides of Equation (1):

$$\min_{s\in[0,t]} g(\theta_s) \leq \frac{\int_0^t \alpha_s g(\theta_s)\mathrm{d}s}{\int_0^t \alpha_s \mathrm{d}s} \leq \frac{\ell_i(\theta_0) - \ell_i(\theta_t)}{\int_0^t \alpha_s \mathrm{d}s} \leq \frac{\ell_i(\theta_0) - \ell^*}{\int_0^t \alpha_s \mathrm{d}s}.$$

This yields the result since it holds for every $i \in [m]$.

If $\int_0^\infty \alpha_t \mathrm{d}t = +\infty$, then we have $\min_{s\in[0,t]} g(\theta_s) \to 0$ when $t \to +\infty$. Assume there exists an $\epsilon > e$, such that θ_t never enters $\mathcal{P}_\epsilon$ at finite t. Then we have $g(\theta_t) \geq \epsilon$ for $t \in \mathbb{R}_+$, which contradicts with $\min_{s\in[0,t]} g(\theta_s) \to 0$.

iii) Assume there exists a finite time $t' \in (t_\epsilon, +\infty)$ such that $\theta_{t'} \notin \overline{\mathcal{P}}_\epsilon$. Because $\epsilon > e$ and g is continuous, $\mathcal{P}_e$ is in the interior of $\mathcal{P}_\epsilon \subseteq \overline{\mathcal{P}}_\epsilon$. Therefore, the trajectory leading to $\theta_{t'} \notin \overline{\mathcal{P}}_\epsilon$ must pass through $\overline{\mathcal{P}}_\epsilon \setminus \mathcal{P}_e$ at some point, that is, there exists a point $t'' \in [t_\epsilon, t')$, such that $\{\theta_t \colon t \in [t'', t']\} \notin \mathcal{P}_e$. But because the algorithm can not increase any objective ℓ_i outside of $\mathcal{P}_e$, we must have $\boldsymbol{\ell}(\theta_{t'}) \preceq \boldsymbol{\ell}(\theta_{t''})$, yielding that $\theta_{t'} \in \overline{\{\theta_{t''}\}} \subseteq \overline{\mathcal{P}}_\epsilon$, where $\overline{\{\theta_{t''}\}}$ is the Pareto closure of $\{\theta_{t''}\}$; this contradicts with the assumption. □

Theorem 3 *Under Assumption 1, assume $\theta_t \notin \mathcal{P}_e$ is a fixed point of the algorithm, that is, $\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -v_t = 0$, and F, $\boldsymbol{\ell}$ are convex in a neighborhood θ_t, then θ_t is a local minimum of F in the Pareto closure $\overline{\{\theta_t\}}$, that is, there exists a neighborhood of θ_t in which there exists no point θ' such that $F(\theta') < F(\theta_t)$ and $\boldsymbol{\ell}(\theta') \preceq \boldsymbol{\ell}(\theta_t)$.*

Proof. Note that minimizing F in $\overline{\{\theta_t\}}$ can be framed into a constrained optimization problem:

$$\min_\theta F(\theta) \quad s.t. \quad \ell_i(\theta) \leq \ell_i(\theta_t), \ \ \forall i \in [m].$$

In addition, by assumption, $\theta = \theta_t$ satisfies $v_t = \nabla F(\theta_t) + \sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta_t) = 0$, which is the KKT stationarity condition of the constrained optimization. It is also obvious to check that $\theta = \theta_t$ satisfies the feasibility and slack condition trivially. Combining this with the local convexity assumption yields the result. □

Theorem 4 [Optimization of F] *Let $\epsilon > e$ and assume $g_\epsilon := \sup_\theta\{g(\theta) \colon \ \theta \in \overline{\mathcal{P}}_\epsilon\} < +\infty$ and $\sup_{t\geq 0} \alpha_t < \infty$. Under Assumption 1, when we initialize from $\theta_0 \in \mathcal{P}_\epsilon$, we have*

$$\min_{s\in[0,t]} \left\| \frac{\mathrm{d}\theta_s}{\mathrm{d}s} \right\|^2 \leq \frac{F(\theta_0) - F^*}{t} + \frac{1}{t}\int_0^t \alpha_s \left(\alpha_s g_\epsilon + c\sqrt{g_\epsilon}\right) \mathrm{d}s.$$

In particular, if we have $\alpha_t = \alpha = const$, then $\min_{s\in[0,t]} \|\mathrm{d}\theta_s / \mathrm{d}s\|^2 = \mathcal{O}\left(1/t + \alpha\sqrt{g_\epsilon}\right)$.
If $\int_0^\infty \alpha_t^\gamma \mathrm{d}t < +\infty$ for some $\gamma \geq 1$, we have $\min_{s\in[0,t]} \|\mathrm{d}\theta_s / \mathrm{d}s\|^2 = \mathcal{O}(1/t + \sqrt{g_\epsilon}/t^{1/\gamma})$.

Proof. i) The slack condition of the constrained optimization in Equation (7) says that

$$\lambda_{i,t}\left(\nabla \ell_i(\theta_t)^\top v_t - \phi_t\right) = 0, \ \forall i \in [m]. \tag{2}$$

This gives that

$$\begin{aligned}\|v_t\|^2 &= \left(\nabla F(\theta_t) + \sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta_t)\right)^\top v_t \\ &= \nabla F(\theta_t)^\top v_t + \sum_{i=1}^m \lambda_{i,t}\phi_t \qquad \text{//plugging Equation (2).} \end{aligned} \tag{3}$$

If $\theta_t \notin \mathcal{P}_e$, we have $\phi_t = \alpha_t g(\theta_t)$ and this gives

$$\frac{\mathrm{d}}{\mathrm{d}t} F(\theta_t) = -\nabla F(\theta_t)^\top v_t = -\|v_t\|^2 + \sum_{i=1}^m \lambda_{i,t}\phi_t = -\left\|\frac{\mathrm{d}\theta_t}{\mathrm{d}t}\right\|^2 + \sum_{i=1}^m \lambda_{i,t}\alpha_t g(\theta_t)$$

If θ_t is in the interior of $\mathcal{P}_e$, then we run typical gradient descent of F and hence has

$$\frac{\mathrm{d}}{\mathrm{d}t} F(\theta_t) = -\|v_t\|^2 = -\left\|\frac{\mathrm{d}\theta_t}{\mathrm{d}t}\right\|^2.$$

If θ_t is on the boundary of $\mathcal{P}_e$, then by the definition of differential inclusion, $\mathrm{d}\theta/\mathrm{d}t$ belongs to the convex hull of the velocities that it receives from either side of the boundary, yielding that

$$\frac{\mathrm{d}}{\mathrm{d}t} F(\theta_t) = -\left\|\frac{\mathrm{d}\theta_t}{\mathrm{d}t}\right\|^2 + \beta \sum_{i=1}^m \lambda_{i,t}\alpha_t g(\theta_t) \le -\left\|\frac{\mathrm{d}\theta_t}{\mathrm{d}t}\right\|^2 + \sum_{i=1}^m \lambda_{i,t}\alpha_t g(\theta_t),$$

where $\beta \in [0,1]$. Combining all the cases gives

$$\frac{\mathrm{d}}{\mathrm{d}t} F(\theta_t) \le -\left\|\frac{\mathrm{d}\theta_t}{\mathrm{d}t}\right\|^2 + \sum_{i=1}^m \lambda_{i,t}\alpha_t g(\theta_t).$$

Integrating this yields

$$\begin{aligned}\min_{s\in[0,t]} \left\|\frac{\mathrm{d}\theta_s}{\mathrm{d}s}\right\|^2 \le \frac{1}{t}\int_0^t \left\|\frac{\mathrm{d}\theta_s}{\mathrm{d}s}\right\|^2 \mathrm{d}s &\le \frac{F(\theta_0) - F^*}{t} + \frac{1}{t}\int_0^t \sum_{i=1}^m \lambda_{i,s}\alpha_s g(\theta_s)\mathrm{d}s \\ &\le \frac{F(\theta_0) - F^*}{t} + \frac{1}{t}\int_0^t \alpha_s\left(\alpha_s g_\epsilon + c\sqrt{g_\epsilon}\right)\mathrm{d}s,\end{aligned}$$

where the last step used Lemma 1 with $\phi_t = \alpha_t g(\theta_t)$:

$$\sum_{i=1}^m \lambda_{i,t}\alpha_t g(\theta_t) \le \alpha_t^2 g(\theta_t) + c\alpha_t\sqrt{g(\theta_t)} \le \alpha_t^2 g_\epsilon + c\alpha_t\sqrt{g_\epsilon},$$

and here we used $g(\theta_t) \le g_\epsilon$ because the trajectory is contained in $\overline{\mathcal{P}}_\epsilon$ following Theorem 2.

The remaining results follow Lemma 3. □

A.0.1 Technical Lemmas

Lemma 1. *Assume Assumption 1 holds. Define $g(\theta) = \min_{\omega\in\mathcal{C}^m} \left\|\sum_{i=1}^m \omega_i \nabla \ell_i(\theta)\right\|^2$, where $\mathcal{C}^m$ is the probability simplex on $[m]$. Then for the v_t and $\lambda_{i,t}$ defined in Equation (7) and Equation (11), we have*

$$\sum_{i=1}^m \lambda_{i,t} g(\theta_t) \le \max\left(\phi_t + c\sqrt{g(\theta_t)},\ 0\right).$$

Proof. The slack condition of the constrained optimization in Equation (7) says that

$$\lambda_{i,t}\left(\nabla \ell_i(\theta)^\top v_t - \phi_t\right) = 0, \quad \forall i \in [m].$$

Sum the equation over $i \in [m]$ and note that $v_t = \nabla F(\theta_t) + \sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta_t)$. We get

$$\left\| \sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta_t) \right\|^2 + \left(\sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta_t) \right)^\top \nabla F(\theta) - \sum_{i=1}^m \lambda_{i,t} \phi_t = 0. \tag{4}$$

Define

$$x_t = \left\| \sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta_t) \right\|^2, \qquad \bar{\lambda}_t = \sum_{i=1}^m \lambda_{i,t}, \qquad g_t = g(\theta_t) = \min_{\omega \in \mathcal{C}^m} \left\| \sum_{i=1}^m \omega_i \nabla \ell_i(\theta_t) \right\|^2.$$

Then it is easy to see that $x_t \geq \bar{\lambda}_t^2 g_t$. Using Cauchy-Schwarz inequality,

$$\left| \left(\sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta) \right)^\top \nabla F(\theta_t) \right| \leq \| \nabla F(\theta_t) \| \left\| \sum_{i=1}^m \lambda_{i,t} \nabla \ell_i(\theta) \right\| \leq c \sqrt{x_t},$$

where we used $\|\nabla F(\theta_t)\| \leq c$ by Assumption 1. Combining this with Equation (4), we have

$$\left| x_t - \bar{\lambda}_t \phi_t \right| \leq c \sqrt{x_t}.$$

Applying Lemma 2 yields the result. □

Lemma 2. *Assume $\phi \in \mathbb{R}$, and $x, \lambda, c, g \in \mathbb{R}_+$ are non-negative real numbers and they satisfy*

$$|x - \lambda \phi| \leq c \sqrt{x}, \qquad x \geq \lambda^2 g.$$

Then we have $\lambda g \leq \max(0, \phi + c\sqrt{g})$.

Proof. Square the first equation, we get

$$f(x) := (x - \lambda\phi)^2 - c^2 x \leq 0,$$

where f is a quadratic function. To ensure that $f(x) \leq 0$ has a solution that satisfies $x \geq \lambda^2 g$, we need to have $f(\lambda^2 g) \leq 0$, that is,

$$f(\lambda^2 g) = (\lambda^2 g - \lambda\phi)^2 - c^2 \lambda^2 g \leq 0.$$

This can hold under two cases:

Case 1: $\lambda = 0$;

Case 2: $|\lambda g - \phi| \leq c\sqrt{g}$, and hence $\phi - c\sqrt{g} \leq \lambda g \leq \phi + c\sqrt{g}$.

Under both case, we have

$$\lambda g \leq \max(0, \phi + c\sqrt{g}).$$

□

Lemma 3. *Let $\{\alpha_t \colon t \in \mathbb{R}_+\} \subseteq \mathbb{R}_+$ be a non-negative sequence with $A := \left(\int_0^\infty \alpha_t^\gamma \mathrm{d}t\right)^{1/\gamma} < \infty$, where $\gamma \geq 1$, and $B = \sup_t \alpha_t < \infty$. Then we have*

$$\frac{1}{t} \int_0^t \left(\alpha_s^2 + \alpha_s\right) \mathrm{d}s \leq (B+1) A t^{-1/\gamma}.$$

Proof. Let $\eta = \frac{\gamma}{\gamma - 1}$, so that $1/\eta + 1/\gamma = 1$. We have by Holder's inequality,

$$\int_0^t \alpha_s \mathrm{d}s \leq \left(\int_0^t \alpha_s^\gamma \mathrm{d}s \right)^{1/\gamma} \left(\int_0^t 1^\eta \mathrm{d}s \right)^{1/\eta} \leq A t^{1/\eta} = A t^{1 - 1/\gamma}.$$

and hence

$$\frac{1}{t} \int_0^t \left(\alpha_s^2 + \alpha_s\right) \mathrm{d}s \leq \frac{B+1}{t} \int_0^t \alpha_s \mathrm{d}s \leq (B+1) A t^{-1/\gamma}.$$

□

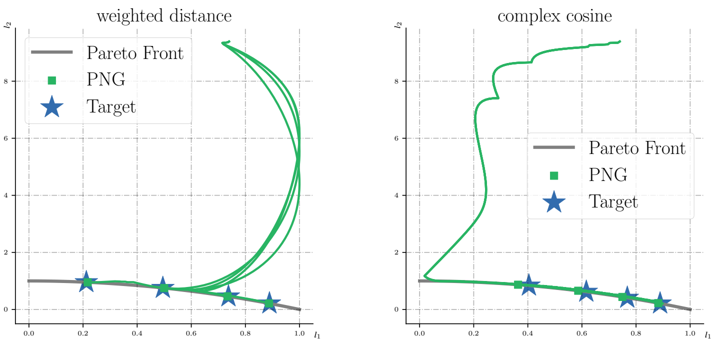

Figure 1: Trajectories of solving OPT-in-Pareto with weighted distance and complex cosine as criterion using PNG. The green dots are the final converged models. PNG is able to successfully locate the correct models in the Pareto set.

B PRACTICAL IMPLEMENTATION

Hyper-parameters Our algorithm introduces two hyperparameters $\{\alpha_t\}$ and e over vanilla gradient descent. We use constant sequence $\alpha_t = \alpha$ and we take $\alpha = 0.5$ unless otherwise specified. We choose e by $e = \gamma e_0$, where e_0 is an exponentially discounted average of $\frac{1}{m}\sum_{i=1}^{m} \|\nabla \ell_i(\theta_t)\|^2$ over the trajectory so that it automatically scales with the magnitude of the gradients of the problem at hand. In the experiments of this paper, we simply fix $\gamma = 0.1$ unless specified.

Solving the Dual Problem Our method requires to calculate $\{\lambda_{i,t}\}_{t=1}^{m}$ with the dual optimization problem in Equation (11), which can be solved with any off-the-shelf convex quadratic programming tool. In this work, we use a very simple projected gradient descent to approximately solve Equation (11). We initialize $\{\lambda_{i,t}\}_{t=1}^{m}$ with a zero vector and terminate when the difference between the last two iterations is smaller than a threshold or the algorithm reaches the maximum number of iterations (we use 100 in all experiments).

C EXPERIMENTS

C.1 FINDING PREFERRED PARETO MODELS

C.1.1 Ratio-based Criterion

The non-uniformity score from [Mahapatra and Rajan, 2020] that we use in Figure 1 is defined as

$$F_{\text{NU}}(\theta) = \sum_{t=1}^{m} \hat{\ell}_t(\theta) \log\left(\frac{\hat{\ell}_t(\theta)}{1/m}\right), \qquad \hat{\ell}_t(\theta) = \frac{r_t \ell_t(\theta)}{\sum_{s\in[m]} r_s \ell_s(\theta)}. \tag{5}$$

We fix the other experiment settings the same as Mahapatra and Rajan [2020] and use $\gamma = 0.01$ and $\alpha = 0.25$ for this experiment reported in the main text. We defer the ablation studies on the hyper-parameter α and γ to Section C.3.

C.1.2 ZDT2-Variant

We consider the ZDT2-Variant example used in Ma et al. [2020] with the same experiment setting, in which the Pareto set is a cylindrical surface, making the problem more challenging. We consider the same criteria, e.g. weighted distance and complex cosine used in the main context with different choices of $r_1 = [0.2, 0.4, 0.6, 0.8]$. We use the default hyper-parameter set up, choosing $\alpha = 0.5$ and $r = 0.1$. For complex cosine, we use MGD updating for the first 150 iterations. Figure 1 shows the trajectories, demonstrating that PNG works pretty well for the more challenging ZDT2-Variant tasks.

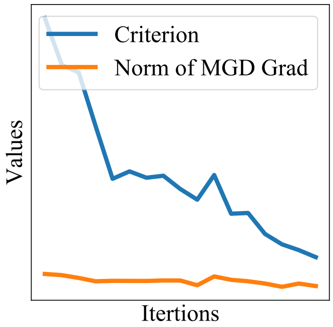

Figure 2: The evolution of Criterion F and the norm of MGD gradient when trained using PNG on NYUv2 dataset with MTAN network. PNG effectively decreases the criterion while ensuring the model is within the Pareto set, since the norm of MGD gradient remains unchanged.

C.1.3 General Criteria: Three-task learning on the NYUv2 Dataset

We show that PNG is able to handle large-scale multitask learning problems by deploying it on a three-task learning problem (segmentation, depth estimation, and surface normal prediction) on NYUv2 dataset [Silberman et al., 2012]. The main goal of this experiment is to show that: 1. PNG is able to handle OPT-in-Pareto in a large-scale neural network; 2. With a proper design of criteria, PNG enables to do targeted fine-tuning that pushes the model to move towards a certain direction. We consider the same training protocol as Liu et al. [2019] and use the MTAN network architecture. Start with a model trained with equally weighted linear scalarization and our goal is to further improve the model's performance on segmentation and surface normal estimation while allowing some sacrifice on depth estimation. This can be achieved by many different choices of criterion and in this experiment, we consider the following design: $F(\theta) = (\ell_{\text{seg}}(\theta) \times \ell_{\text{surface}}(\theta))/(0.001 + \ell_{\text{depth}}(\theta))$. Here ℓ_{seg}, ℓ_{surface} and ℓ_{depth} are the loss functions for segmentation, surface normal prediction and depth estimation, respectively. The constant 0.001 in the denominator is for numeric stability. We point out that our design of criterion is a simple heuristic and might not be an optimal choice and the key question we study here is to verify the functionality of the proposed PNG. As suggested by the open-source repository of Liu et al. [2019], we reproduce the result based on the provided configuration. To show that PNG is able to move the model along the Pareto front, we show the evolution of the criterion function and the norm of the MGD gradient during the training in Figure 2. As we can see, PNG effectively decreases the value of criterion function while the norm of MGD gradient remains the same. This demonstrates that PNG is able to minimize the criterion by searching the model in the Pareto set. Table 1 compares the performances on the three tasks using standard training and PNG, showing that PNG is able to improve the model's performance on segmentation and surface normal prediction tasks while satisfying a bit of the performance in depth estimation based on the criterion.

C.2 FINDING DIVERSE PARETO MODELS

C.2.1 Experiment Details

We train the model for 100 epochs using Adam optimizer with batch size 256 and 0.001 learning rate. To encourage diversity of the models, following the setting in Mahapatra and Rajan [2020], we use equally distributed preference vectors for linear scalarization and EPO. Note that the stochasticity of using mini-batches is able to improve the performance of Pareto approximation for free by also using the intermediate checkpoints to approximate $\mathcal{P}$. To fully exploit this advantage, for all the methods, we collect checkpoints every epoch to approximate $\mathcal{P}$, starting from epoch 60.

C.2.2 Evaluation Metric Details

We introduce the definition of the used metric for evaluation. Given a set $\hat{\mathcal{P}} = \{\theta_1, \ldots, \theta_N\}$ that we use to approximate $\mathcal{P}$, its IGD+ score is defined as:

$$\text{IGD}_+(\hat{\mathcal{P}}) = \int_{\mathcal{P}^*} q(\theta, \hat{\mathcal{P}}) d\mu(\theta), \quad q(\theta, \hat{\mathcal{P}}) = \min_{\hat{\theta} \in \hat{\mathcal{P}}} \left\| \left(\boldsymbol{\ell}(\hat{\theta}) - \boldsymbol{\ell}(\theta) \right)_+ \right\|,$$

Algorithm	Segmentation		Depth		Surface Normal				
	(Higher Better)		(Lower Better)		Angle Distance (Lower Better)		Within t°		
	mIoU	Pix Acc	Abs Err	Rel Err	Mean	Median	11.25	22.5	30
Standard	27.09	56.36	0.6143	0.2618	31.46	27.37	19.51	41.71	54.61
PNG	28.23	56.66	0.6161	0.2632	31.06	26.50	21.06	43.41	55.93

Table 1: Comparing the multitask performance of standard training using linear scalarization with equally weighted losses and the targeted fine-tuning based on PNG.

		Loss		Acc	
		Hv↑ (10^{-2})	IGD↓ (10^{-2})	Hv↑ (10^{-2})	IGD↓ (10^{-2})
$\gamma = 0.1$	$\alpha = 0.25$	7.89 ± 0.11	0.041 ± 0.012	9.39 ± 0.038	0.0056 ± 0.002
	$\alpha = 0.5$	7.86 ± 0.12	0.043 ± 0.012	9.39 ± 0.038	0.0056 ± 0.002
	$\alpha = 0.75$	7.84 ± 0.11	0.045 ± 0.013	9.38 ± 0.037	0.0057 ± 0.002
$\alpha = 0.5$	$\gamma = 0.01$	7.86 ± 0.12	0.042 ± 0.012	9.39 ± 0.038	0.0056 ± 0.002
	$\gamma = 0.1$	7.86 ± 0.12	0.043 ± 0.012	9.39 ± 0.038	0.0056 ± 0.002
	$\gamma = 0.25$	7.85 ± 0.11	0.042 ± 0.012	9.39 ± 0.036	0.0056 ± 0.002

Table 2: Ablation study based on Multi-Mnist dataset with different choice of α and γ.

where μ is some base measure that measures the importance of $\theta \in \mathcal{P}$ and $(t)_+ := \max(t, 0)$, applied on each element of a vector. Intuitively, for each θ, we find a nearest $\hat{\theta} \in \hat{\mathcal{P}}$ that approximates θ best. Here the $(\cdot)_+$ is applied as we only care the tasks that $\hat{\theta}$ is worse than θ. In practice, a common choice of μ can be a uniform counting measure with uniformly sampled (or selected) models from $\mathcal{P}$. In our experiments, since we can not sample models from $\mathcal{P}$, we approximate $\mathcal{P}$ by combining $\hat{\mathcal{P}}$ from all the methods, i.e., $\mathcal{P} \approx \cup_{m \in \{\text{Linear,MGD,EPO,PNG}\}} \hat{\mathcal{P}}_m$, where $\hat{P}_m$ is the approximation set produced by algorithm m.

This approximation might not be accurate but is sufficient to compare the different methods,

The Hypervolume score of $\hat{\mathcal{P}}$, w.r.t. a reference point $\boldsymbol{\ell}^r \in \mathbb{R}_+^m$, is defined as

$$\mathrm{HV}(\hat{\mathcal{P}}) = \mu\left(\left\{\boldsymbol{\ell} = [\ell_1, ..., \ell_m] \in \mathbb{R}^m \mid \exists \theta \in \hat{\mathcal{P}},\ \text{s.t.}\ \ell_t(\theta) \leq \ell_t \leq \ell_t^r\ \forall t \in [m]\right\}\right),$$

where μ is again some measure. We use $\boldsymbol{\ell}^r = [0.6, 0.6]$ for calculating the Hypervolume based on loss and set μ to be the common Lebesgue measure. Here we choose 0.6 as we observe that the losses of the two tasks are higher than 0.6 and 0.6 is roughly the worst case. When calculating Hypervolume based on accuracy, we simply flip the sign.

C.2.3 Ablation Study

We conduct ablation study to understand the effect of α and γ using the Pareto approximation task on Multi-Mnist. We compare PNG with $\alpha = 0.25, 0.5, 0.75$ and $\gamma = 0.01, 0.1, 0.25$. Figure 2 summarizes the result. Overall, we observe that PNG is not sensitive to the choice of hyper-parameter.

C.2.4 Comparing with the Second Order Approach

We give a discussion on comparing our approach with the second order approaches proposed by Ma et al. [2020]. In terms of algorithm, Ma et al. [2020] is a local expansion approach. To apply Ma et al. [2020], in the first stage, we need to start with several well distributed models (i.e., the ones obtained by linear scalarization with different preference weights) and Ma et al. [2020] is only applied in the second stage to find the neighborhood of each model. The performance gain comes from the local neighbor search of each model (i.e. the second stage).

In comparison, PNG with energy distance is a global search approach. It improves the well-distributedness of models in the first stage (i.e. it's a better approach than simply using linear scalarization with different weights). And thus the performance gain comes from the first stage. Notice that we can also apply Ma et al. [2020] to PNG with energy distance to add extra local search to further improve the approximation.

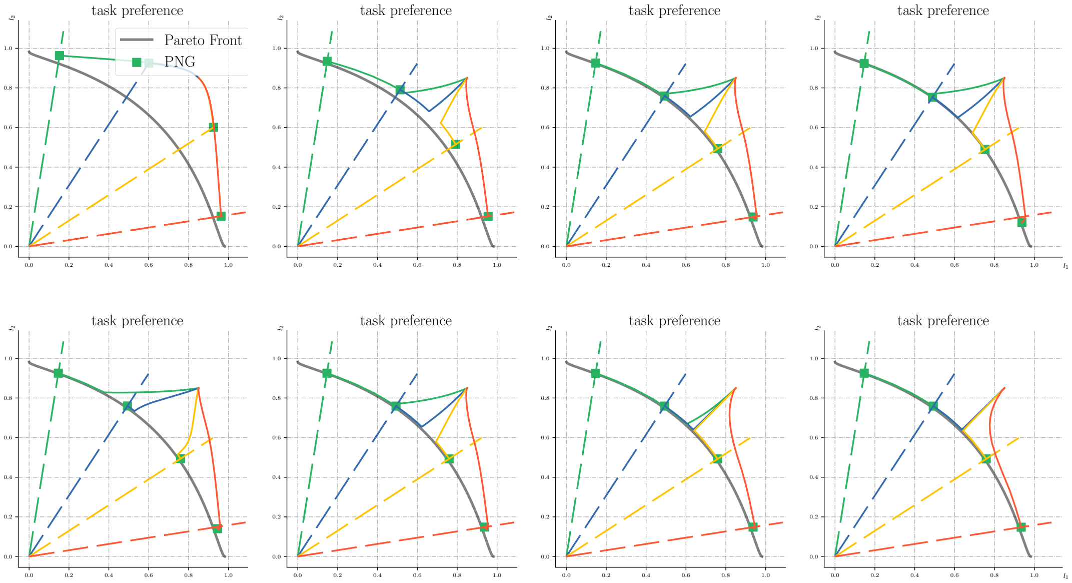

Figure 3: Ablation study on OPT-in-Pareto with different ratio constraint of objectives. Upper row, from left to right: fixing $\alpha = 0.25$, $\gamma = 0.1, 0.05, 0.01, 0.001$; Lower row, from left to right: fixing $\gamma = 0.01$, $\alpha = 0.1, 0.25, 0.5, 0.75$. By comparing the figures in the first row, we find that choosing a too large γ make the final converged model be far away from the Pareto set, which is as expected. By comparing the figures in the second row, we find that changing α make PNG give different priority in making Pareto improvement or descent on F. When α is larger (the right figures), PNG will first move the model to Pareto set and start to decrease F after that.

In terms of run time comparison. We compare the wall clock run time of each step of updating the 5 models using PNG and the second order approach in Ma et al. [2020]. We calculate the run time based on the multi-MNIST dataset using the average of 100 steps. PNG uses 0.3s for each step while Ma et al. [2020] uses 16.8s. PNG is *56x* faster than the second order approach. And we further argue that, based on time complexity theory, the gap will be even larger when the size of the network increases.

C.3 TRAJECTORY VISUALIZATION WITH DIFFERENT HYPER-PARAMETERS

We give visualization on the PNG trajectory when using different hyper-parameters. We reuse synthetic example introduced in Section 7.1 for studying the hyper-parameters α and γ. We fix $\alpha = 0.25$ and vary $\gamma = 0.1, 0.05, 0.01, 0.1$; and fix $\gamma = 0.01$ and vary $\alpha = 0.1, 0.25, 0.5, 0.75$. Figure 3 plots the trajectories. As we can see, when γ is properly chosen, with different α, PNG finds the correct models with different trajectories. Different α determines the algorithm's behavior of balancing the descent of task losses or criterion objectives. On the other hand, with too large γ, the algorithm fails to find a model that is close to $\mathcal{P}^*$, which is expected.

C.4 IMPROVING MULTITASK BASED DOMAIN GENERALIZATION

We argue that many other deep learning problems also have the structure of multitask learning when multiple losses presents and thus optimization techniques in multitask learning can also be applied to those domains. In this paper we consider the JiGen [Carlucci et al., 2019]. JiGen learns a model that can be generalized to unseen domain by minimizing a standard cross-entropy loss ℓ_{class} for classification and an unsupervised loss ℓ_{jig} based on Jigsaw Puzzles:

$$\ell(\theta) = (1 - \omega)\ell_{\text{class}}(\theta) + \omega\ell_{\text{jig}}(\theta).$$

The ratio between two losses, i.e. ω, is important to the final performance of the model and requires a careful grid search. Notice that JiGen is essentially searching for a model on the Pareto front using the linear scalarization. Instead of using a fixed linear scalarization to learn a model, one natural questions is that whether it is possible to design a mechanism that dynamically adjusts the ratio of the losses so that we can achieve to learn a better model.

Method	Art paint	Cartoon	Sketches	Photo	Avg
AlexNet					
TF	0.6268	0.6697	0.5751	0.8950	0.6921
CIDDG	0.6270	0.6973	0.6445	0.7865	0.6888
MLDG	0.6623	0.6688	0.5896	0.8800	0.7001
D-SAM	0.6387	0.7070	0.6466	0.8555	0.7120
DeepAll	0.6668	0.6941	0.6002	0.8998	0.7152
JiGen	0.6855 ± 0.004	$\mathbf{0.6889 \pm 0.002}$	$\mathbf{0.6831 \pm 0.011}$	0.8946 ± 0.008	0.7380 ± 0.002
JiGen + adv	0.6857 ± 0.004	0.6837 ± 0.003	0.6753 ± 0.008	0.8980 ± 0.001	0.7357 ± 0.003
Jigen + PNG	$\mathbf{0.6914 \pm 0.005}$	$\mathbf{0.6903 \pm 0.002}$	$\mathbf{0.6855 \pm 0.007}$	$\mathbf{0.9044 \pm 0.003}$	$\mathbf{0.7429 \pm 0.002}$
ResNet-18					
D-SAM	0.7733	0.7243	0.7783	0.9530	0.8072
DeepAll	0.7785	0.7486	0.6774	0.9573	0.7905
JiGen	0.8009 ± 0.004	0.7363 ± 0.007	0.7046 ± 0.013	$\mathbf{0.9629 \pm 0.002}$	0.8012 ± 0.002
JiGen + adv	0.7923 ± 0.006	0.7402 ± 0.004	0.7188 ± 0.005	0.9617 ± 0.001	0.8033 ± 0.001
JiGen + PNG	$\mathbf{0.8014 \pm 0.005}$	$\mathbf{0.7538 \pm 0.001}$	$\mathbf{0.7222 \pm 0.006}$	$\mathbf{0.9627 \pm 0.002}$	$\mathbf{0.8100 \pm 0.005}$

Table 3: Comparing different algorithms for domain generalization using dataset PACS and two network architectures. The setting is the same to that of Table 2.

We give a case study here. Motivated by the adversarial feature learning [Ganin et al., 2016], we propose to improve JiGen such that the latent feature representations of the two tasks are well aligned. Specifically, suppose that $\Phi_{\text{class}}(\theta) = \{\phi_{\text{class}}(x_i, \theta)\}_{i=1}^n$ and $\Phi_{\text{jig}}(\theta) = \{\phi_{\text{jig}}(x_i, \theta)\}_{i=1}^n$ is the distribution of latent feature representation of the two tasks, where x_i is the i-th training data. We consider F_{PD} as some probability metric that measures the distance between two distributions, we consider the following problem:

$$\min_{\theta \in \mathcal{P}^*} F_{\text{PD}}[\Phi_{\text{class}}(\theta), \Phi_{\text{jig}}(\theta)].$$

With PD as the criterion function, our algorithm automatically reweights the ratio of the two tasks such that their latent space is well aligned.

Setup We fix all the experiment setting the same as Carlucci et al. [2019]. We use the Alexnet and Resnet-18 with multihead pretrained on ImageNet as the multitask network. We evaluate the methods on PACS [Li et al., 2017], which covers 7 object categories and 4 domains (Photo, Art Paintings, Cartoon and Sketches). Same to Carlucci et al. [2019], we trained our model considering three domains as source datasets and the remaining one as target. We implement F_{PD} that measures the discrepancy of the feature space of the two tasks using the idea of Domain Adversarial Neural Networks [Ganin and Lempitsky, 2015] by adding an extra prediction head on the shared feature space to predict the whether the input is for the classification task or Jigsaw task. Specifically, we add an extra linear layer on the shared latent feature representations that is trained to predict the task that the latent space belongs to, i.e.,

$$F_{\text{PD}}(\Phi_{\text{class}}(\theta), \Phi_{\text{jig}}(\theta)) = \min_{w,b} \frac{1}{n} \sum_{i=1}^{n} \log(\sigma(w^\top \phi_{\text{class}}(x_i, \theta))) + \log(1 - \sigma(w^\top \phi_{\text{class}}(x_i, \theta))).$$

Notice that the optimal weight and bias for the linear layer depends on the model parameter θ, during the training, both w, b and θ are jointly updated using stochastic gradient descent. We follow the default training protocol provided by the source code of Carlucci et al. [2019].

Baselines Our main baselines are JiGen [Carlucci et al., 2019]; JiGen + adv, which adds an extra domain adversarial loss on JiGen; and our PNG with domain adversarial loss as criterion function. In order to run statistical test for comparing the methods, we run all the main baselines using 3 random trials. We use the released source code by Carlucci et al. [2019] to obtained the performance of JiGen. For JiGen+adv, we use an extra run to tune the weight for the domain adversarial loss. Besides the main baselines, we also includes TF [Li et al., 2017], CIDDG [Li et al., 2018b], MLDG [Li et al., 2018a] , D-SAM [D'Innocente and Caputo, 2018] and DeepAll [Carlucci et al., 2019] as baselines with the author reported performance for reference.

Result The result is summarized in Table 3 with bolded value indicating the statistical significant best methods with p-value based on matched-pair t-test less than 0.1. Combining Jigen and PNG to dynamically reweight the task weights is able to implicitly regularizes the latent space without adding an actual regularizer which might hurt the performance on the tasks and thus improves the overall result.

References

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Antonio D'Innocente and Barbara Caputo. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*, 2018.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018a.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018b.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning*, 2020.

Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, 2020.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, 2012.