Efficient Continuous Pareto Exploration in Multi-Task Learning

Pingchuan Ma [*1] **Tao Du** [*1] **Wojciech Matusik** [1]

1. Proofs

1.1. Proof of Proposition 4.1

Proof. The last constraint establishes the connection between $\boldsymbol{c}$ and $\boldsymbol{\alpha}$:

$$\sum_{i=1}^{m} \alpha_i \nabla f_i(\boldsymbol{x}_0^*) = (\sum_{i=1}^{m} \alpha_i)\boldsymbol{c} = \boldsymbol{c} \tag{1}$$

The second equality comes from the sum of α_i being 1. Therefore, the optimal solution $\boldsymbol{\alpha}^*$ and $\boldsymbol{c}^*$ must satisfy $\boldsymbol{c}^* = \nabla \boldsymbol{f}(\boldsymbol{x}_0^*)^\top \boldsymbol{\alpha}^*$. Plugging $\boldsymbol{c}^*$ back to Problem (5) reduces it to Problem (3), showing that both problems share the same optimal $\boldsymbol{\alpha}^*$. □

1.2. Proof of Proposition 4.2

Proof. For simplicity, we let $\boldsymbol{v} = \boldsymbol{x}'(0)$. We first prove $\boldsymbol{c}_{\boldsymbol{v}}(t) = \boldsymbol{f}(\boldsymbol{x}^* + t\boldsymbol{v})$ and $\boldsymbol{c}_{\boldsymbol{v}+u}(t) = \boldsymbol{f}(\boldsymbol{x}^* + t(\boldsymbol{v}+\boldsymbol{u}))$ have the same value and tangent direction at $t = 0$, i.e., $\boldsymbol{c}_{\boldsymbol{v}}(0) = \boldsymbol{c}_{\boldsymbol{v}+\boldsymbol{u}}(0)$ and $\boldsymbol{c}'_{\boldsymbol{v}}(0) = \boldsymbol{c}'_{\boldsymbol{v}+\boldsymbol{u}}(0)$. The first equality is trivial because both equals $\boldsymbol{f}(\boldsymbol{x}^*)$. To show they have the same tangent direction, note that

$$\begin{aligned} \boldsymbol{c}'_{\boldsymbol{v}+\boldsymbol{u}}(t) =& (f_1'(\boldsymbol{x}^* + t(\boldsymbol{v}+\boldsymbol{u})), f_2'(\boldsymbol{x}^* + t(\boldsymbol{v}+\boldsymbol{u}))) \\ =& \nabla \boldsymbol{f}(\boldsymbol{x}^* + t(\boldsymbol{v}+\boldsymbol{u}))(\boldsymbol{v}+\boldsymbol{u}) \end{aligned} \tag{2}$$

Taking $t = 0$ gives

$$\boldsymbol{c}'_{\boldsymbol{v}+\boldsymbol{u}}(0) = \nabla \boldsymbol{f}(\boldsymbol{x}^*)(\boldsymbol{v}+\boldsymbol{u}) \tag{3}$$

Since $\boldsymbol{x}^*$ is Pareto optimal, we have $\boldsymbol{\alpha}^\top \nabla \boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$ (Proposition 3.1). Therefore, the dot product between $\boldsymbol{\alpha}$ and $\boldsymbol{c}'_{\boldsymbol{v}+\boldsymbol{u}}(0)$ is

$$\boldsymbol{\alpha}^\top \boldsymbol{c}'_{\boldsymbol{v}+\boldsymbol{u}}(0) = \boldsymbol{\alpha}^\top \nabla \boldsymbol{f}(\boldsymbol{x}^*)(\boldsymbol{v}+\boldsymbol{u}) = 0 \tag{4}$$

This indicates that $\boldsymbol{c}'_{\boldsymbol{v}+\boldsymbol{u}}(0)$ is orthogonal to $\boldsymbol{\alpha}$. Since $m = 2$, we conclude $\boldsymbol{c}'_{\boldsymbol{v}+\boldsymbol{u}}(0)$ is parallel to $(-\alpha_2, \alpha_1)$ no matter how $\boldsymbol{u}$ is chosen.

We now prove the second part of the proposition. First, $\boldsymbol{v}$ and $\boldsymbol{u}$ are not parallel because $\boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{u} = \boldsymbol{0}$ and $\boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{v} \neq \boldsymbol{0}$. The implication is that adding $\boldsymbol{u}$ to the Pareto set spanned by $\nabla f_1(\boldsymbol{x}^*)$ and $\nabla f_2(\boldsymbol{x}^*)$ indeed augments it by bringing a new dimension.

Second, we show $\boldsymbol{c}_{\boldsymbol{v}}(t)$ and $\boldsymbol{c}_{\boldsymbol{v}+\boldsymbol{u}}(t)$ have the same curvature at $t = 0$. To see this, note that the curvature of $\boldsymbol{c}_{\boldsymbol{d}}(t)$ at $t = 0$ is defined as:

$$\kappa = \frac{f_1' f_2'' - f_2' f_1''}{(f_1'^2 + f_2'^2)^{3/2}} \tag{5}$$

where $f_i' = f_i'(\boldsymbol{x}^* + t\boldsymbol{d})|_{t=0}$ and $f_i'' = f_i''(\boldsymbol{x}^* + t\boldsymbol{d})|_{t=0}$, $i = 1, 2$. It is now sufficient to show the denominators and numerators are the same for $\boldsymbol{d} = \boldsymbol{v}$ and $\boldsymbol{d} = \boldsymbol{v} + \boldsymbol{u}$. We prove the following equality to establish the denominators are the same:

$$f_i'(\boldsymbol{x}^* + t\boldsymbol{v})|_{t=0} = f_i'(\boldsymbol{x}^* + t(\boldsymbol{v}+\boldsymbol{u}))|_{t=0}, \quad i = 1, 2 \tag{6}$$

To see this, we expand the right-hand side:

$$\begin{aligned} f_i'(\boldsymbol{x}^* + t(\boldsymbol{v}+\boldsymbol{u}))|_{t=0} =& (\boldsymbol{v}+\boldsymbol{u})^\top \nabla f_i(\boldsymbol{x}^*) \\ =& \boldsymbol{v}^\top \nabla f_i(\boldsymbol{x}^*) + \boldsymbol{u}^\top \nabla f_i(\boldsymbol{x}^*) \\ =& f_i'(\boldsymbol{x}^* + t\boldsymbol{v})|_{t=0} + \boldsymbol{u}^\top \nabla f_i(\boldsymbol{x}^*) \end{aligned} \tag{7}$$

It remains to show that $\boldsymbol{u}^\top \nabla f_i(\boldsymbol{x}^*) = 0$, or these two vectors are orthogonal. Recall that

$$\alpha_1 \nabla f_1(\boldsymbol{x}^*) + \alpha_2 \nabla f_2(\boldsymbol{x}^*) = \boldsymbol{0} \tag{8}$$

where α_i comes from Proposition 3.1. Since $\alpha_1 + \alpha_2 = 1$, at least one of them is nonzero. Without loss of generality, we assume $\alpha_1 \neq 0$, which gives us

$$\nabla f_1(\boldsymbol{x}^*) = -\frac{\alpha_2}{\alpha_1} \nabla f_2(\boldsymbol{x}^*) \tag{9}$$

If $\nabla f_2(\boldsymbol{x}^*) = \boldsymbol{0}$, $\nabla f_1(\boldsymbol{x}^*)$ has to be $\boldsymbol{0}$ as well, and $\boldsymbol{u}^\top \nabla f_i(\boldsymbol{x}_i) = 0$ is trivial. Below we assume $\nabla f_2(\boldsymbol{x}^*) \neq \boldsymbol{0}$. Therefore, the space spanned by $\{\nabla f_1(\boldsymbol{x}^*), \nabla f_2(\boldsymbol{x}^*)\}$ is effectively a one-dimensional line in the direction of $\nabla f_2(\boldsymbol{x}^*)$. Now consider applying Proposition 3.2 to $\boldsymbol{v}$:

$$\boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{v} = \nabla f_2(\boldsymbol{x}^*)\beta \tag{10}$$

where β is some scalar whose exact value is determined by Proposition 3.2. Note that the right-hand side has been

[*]Equal contribution [1]MIT CSAIL. Correspondence to: Pingchuan Ma <pcma@csail.mit.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

simplified due to the fact that $\nabla f_1(\boldsymbol{x}^*)$ and $\nabla f_2(\boldsymbol{x}^*)$ are parallel. Using the fact that $\boldsymbol{u}$ is a null vector of $\boldsymbol{H}(\boldsymbol{x}^*)$ and $\boldsymbol{H}(\boldsymbol{x}^*)$ is a symmetric matrix, we establish the orthogonality between $\boldsymbol{u}$ and $\nabla f_2(\boldsymbol{x}^*)\beta$ as follows:

$$\boldsymbol{u}^\top \nabla f_2(\boldsymbol{x}^*)\beta = \boldsymbol{u}^\top \boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{v} = (\boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{u})^\top \boldsymbol{v} = 0 \quad (11)$$

Since $\boldsymbol{H}(\boldsymbol{x}^*)\boldsymbol{v} \neq \boldsymbol{0}$, β is nonzero. We then conclude $\boldsymbol{u}^\top \nabla f_2(\boldsymbol{x}^*) = 0$. It follows that $\boldsymbol{u}^\top \nabla f_1(\boldsymbol{x}^*) = -\alpha_2 \boldsymbol{u}^\top \nabla f_2(\boldsymbol{x}^*)/\alpha_1 = 0$.

To show the numerators are the same, we first calculate the second-order derivatives for $\boldsymbol{d} = \boldsymbol{v}$ as follows:

$$\begin{aligned} f_i'(\boldsymbol{x}^* + t\boldsymbol{v}) =& \boldsymbol{v}^\top \nabla f_i(\boldsymbol{x}^* + t\boldsymbol{v}) \\ f_i''(\boldsymbol{x}^* + t\boldsymbol{v}) =& \boldsymbol{v}^\top \nabla^2 f_i(\boldsymbol{x}^* + t\boldsymbol{v})\boldsymbol{v} \\ f_i''(\boldsymbol{x}^* + t\boldsymbol{v})|_{t=0} =& \boldsymbol{v}^\top \nabla^2 f_i(\boldsymbol{x}^*)\boldsymbol{v} \end{aligned} \quad (12)$$

As a result, when $\boldsymbol{d} = \boldsymbol{v}$, the numerator is (we simplified the notation by ignoring $\boldsymbol{x}^*$ in ∇f_i and $\nabla^2 f_i$)

$$\begin{aligned} & f_1' f_2'' - f_2' f_1'' \\ =& \boldsymbol{v}^\top \nabla f_1 \boldsymbol{v}^\top \nabla^2 f_2 \boldsymbol{v} - \boldsymbol{v}^\top \nabla f_2 \boldsymbol{v}^\top \nabla^2 f_1 \boldsymbol{v} \\ =& \boldsymbol{v}^\top (\nabla f_1 \boldsymbol{v}^\top \nabla^2 f_2 - \nabla f_2 \boldsymbol{v}^\top \nabla^2 f_1)\boldsymbol{v} \\ =& \boldsymbol{v}^\top (-\frac{\alpha_2}{\alpha_1} \nabla f_2 \boldsymbol{v}^\top \nabla^2 f_2 - \nabla f_2 \boldsymbol{v}^\top \nabla^2 f_1)\boldsymbol{v} \\ =& -\frac{1}{\alpha_1} \boldsymbol{v}^\top \nabla f_2 \boldsymbol{v}^\top (\alpha_2 \nabla^2 f_2 + \alpha_1 \nabla^2 f_1)\boldsymbol{v} \\ =& -\frac{1}{\alpha_1} \boldsymbol{v}^\top \nabla f_2 \boldsymbol{v}^\top \boldsymbol{H}\boldsymbol{v} \end{aligned} \quad (13)$$

Replacing $\boldsymbol{v}$ with $\boldsymbol{v} + \boldsymbol{u}$ in the last equation gives us the numerator when $\boldsymbol{d} = \boldsymbol{v} + \boldsymbol{u}$:

$$\begin{aligned} & f_1' f_2'' - f_2' f_1'' \\ =& -\frac{1}{\alpha_1} (\boldsymbol{v} + \boldsymbol{u})^\top \nabla f_2 (\boldsymbol{v} + \boldsymbol{u})^\top \boldsymbol{H}(\boldsymbol{v} + \boldsymbol{u}) \\ =& -\frac{1}{\alpha_1} \boldsymbol{v}^\top \nabla f_2 (\boldsymbol{v} + \boldsymbol{u})^\top \boldsymbol{H}(\boldsymbol{v} + \boldsymbol{u}) \\ =& -\frac{1}{\alpha_1} \boldsymbol{v}^\top \nabla f_2 \boldsymbol{v}^\top \boldsymbol{H}(\boldsymbol{v} + \boldsymbol{u}) \\ =& -\frac{1}{\alpha_1} \boldsymbol{v}^\top \nabla f_2 \boldsymbol{v}^\top \boldsymbol{H}\boldsymbol{v} \end{aligned} \quad (14)$$

where the second equality was derived with the fact $\boldsymbol{u}^\top \nabla f_2 = 0$ and the last two equalities used $\boldsymbol{H}\boldsymbol{u} = \boldsymbol{0}$. This shows that the two curves have the same numerators at $t = 0$. Putting it together, we have proven $\boldsymbol{c}_{\boldsymbol{v}}(t)$ and $\boldsymbol{c}_{\boldsymbol{v}+\boldsymbol{u}}(t)$ have the same curvature at $t = 0$ when $\boldsymbol{u}$ is a null vector of $\boldsymbol{H}$. □

2. Experimental Setup

2.1. ZDT2-variant

This example has an analytic $\boldsymbol{f}(\boldsymbol{x}) : (x_1, x_2, x_3) \in \mathbb{R}^3 \to (f_1, f_2) \in \mathbb{R}^2$, defined as follows:

$$\begin{aligned} y_1 =& \frac{\sin(x_1 + x_2^2 + x_3^2) + 1}{2} \\ y_2 =& \frac{\cos(x_2^2 + x_3^2) + 1}{2} \\ y_3 =& y_2 \\ g =& 1 + \frac{9}{2}(y_2 + y_3) \\ f_1(x_1, x_2, x_3) =& y_1 \\ f_2(x_1, x_2, x_3) =& g - \frac{y_1^2}{g} \end{aligned} \quad (15)$$

The Pareto front is given by

$$f_2 = 1 - f_1^2 \quad f_1 \in [0, 1] \quad (16)$$

and the analytic Pareto set is

$$x_2^2 + x_3^2 = (2k + 1)\pi, \quad k = 0, 1, 2, \cdots \quad (17)$$

which is a family of concentric cylinders. In the paper, we analyzed the innermost Pareto set $x_2^2 + x_3^2 = \pi$. The rightmost figure in Figure 2 in the paper was generated by plotting $\boldsymbol{f}(\boldsymbol{x}^* + s\boldsymbol{d}), s \in [-0.1, 0.1]$ with $\boldsymbol{d}$ being a unit vector of $\nabla f_1(\boldsymbol{x}^*)$, $\nabla f_2(\boldsymbol{x}^*)$, and the approximated tangent directions after 2 iteration of MINRES respectively.

The experiments in Figure 3 of the main paper were set up as follows: starting with a randomly chosen Pareto optimal $\boldsymbol{x}^*$, we spawned a new $\boldsymbol{x}$ by computing $\boldsymbol{x} = \boldsymbol{x}^* + 0.1\boldsymbol{d}$ where $\boldsymbol{d}$ is a unit vector calculated from two methods: 1) running MINRES for 2 iteration to get the approximated tangent direction; 2) perturbing α at $\boldsymbol{x}^*$ to get α' and letting $\boldsymbol{d} = \alpha_1'(\boldsymbol{x}^*) + \alpha_2' \nabla f_2(\boldsymbol{x}^*)$. The second method is the WeightedSum baseline introduced in the main paper and can be interpreted as exploring by running one iteration of gradient-descent to minimize $\alpha_1' f_1 + \alpha_2' f_2$. We then used MGDA (Désidéri, 2012) plus line search to push new $\boldsymbol{x}$ back to the Pareto front. The step size in our line search was initially 1 and decayed by 0.9 exponentially.

2.2. MultiMNIST Subset

We first generated the full MultiMNIST dataset (see Section 2.3) and picked a subset of 2048 images, downsampled from 28×28 to 14×14, as our MultiMNIST Subset example. The two objectives are the cross entropy losses of classifying the top-left and bottom-right digits evaluated on all 2048 images. Regarding the classifier, we used a modified LeNet5 (LeCun et al., 1998) network, which has 1500 parameters. Our modified network starts with a convolutional

layer with 10 channels, a 5×5 kernel, and a stride of 2 pixels, followed by a 2×2 max pooling layer. Next, the results are fed into a fully connected layer of size 20×10 and then sent to two fully connected layers, one for each task. We use ReLU as the nonlinear function in the network. Essentially, this synthetic example attempts to use a small network to overfit 2048 images. To generate the Pareto front, we ran BFGS (Nocedal & Wright, 2006) to optimize $w_1 f_1 + w_2 f_2$ with $w_1 = 0, 0.01, 0.02, \cdots, 1$ from the same random initial guess, which generated a list of 101 solutions $\boldsymbol{x}_0^*, \boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \cdots, \boldsymbol{x}_{101}^*$. We then linearly interpolated $\boldsymbol{f}(\boldsymbol{x}_i^*), i = 0, 1, 2, \cdots, 100$ and treat the resulting piecewise linear spline as the (empirical) Pareto front.

The experiment in Figure 4 of the main paper was conducted as follows: starting with a randomly chosen $\boldsymbol{x}_i^*$, we plotted $\boldsymbol{f}(\boldsymbol{x}_i^* + s\boldsymbol{d}), s \in [-0.5, 0.5]$ where $\boldsymbol{d}$ is a unit vector of the approximated tangent direction. We got the tangent direction by running 50 MINRES iterations to solve Equation (6) of the main paper with $\boldsymbol{\beta}$ sampled from a standard normal distribution. In particular, we found gradient correction (Equation (5) of the main paper) useful in this example. We then ran 50 iterations of gradient-descent (GD) to minimize f_1, f_2, and $w_1 f_1 + w_2 f_2$ respectively. Here w_1 and w_2 are perturbed from the corresponding $\boldsymbol{\alpha}$ vector at $\boldsymbol{x}_i^*$. This shows how well gradient-descent can explore the Pareto front within the time budget of 50 times of back-propagation. We used $1/\sqrt{t+1}$ where t is the iteration index to decay the learning rate in GD from 0.005.

2.3. MultiMNIST and Its Variants

Dataset and Task Description We followed Sabour et al. (2017) to generate MultiMNIST, FashionMNIST, and MultiFashionMNIST. We first created 36×36 images by placing two 28×28 images from MNIST or FashionMNIST (Xiao et al., 2017) in the upper-left and lower-right corner with a random shift of up to 2 pixels in each direction. The synthesized images were then resized to 28×28 and normalized with a mean of 0.1307 and a standard deviation of 0.3081. No data augmentation was used for training or testing. Following ParetoMTL (Lin et al., 2019), we built MultiMNIST from MNIST, MultiFashion from FashionMNIST, and MultiFashionMNIST from both (Figure 1). Each dataset has 60,000 training images and 10,000 test images. The objectives are the cross entropy losses of classifying the upper-left and lower right items in the image.

Network Architecture The backbone network is a modified LeNet (LeCun et al., 1998). Our network starts from two convolutional layers with a 5×5 kernel and a stride of 1 pixel. The two layers have 10 and 20 channels respectively. A fully connected layer of 50 channels appends the convolutional layers, which is then followed by two 10-channel fully connected layers, one for each task. We add a 2×2 max pooling layer right after each convolutional layer and use ReLU as the nonlinear function. The network contains 22,350 trainable parameters.

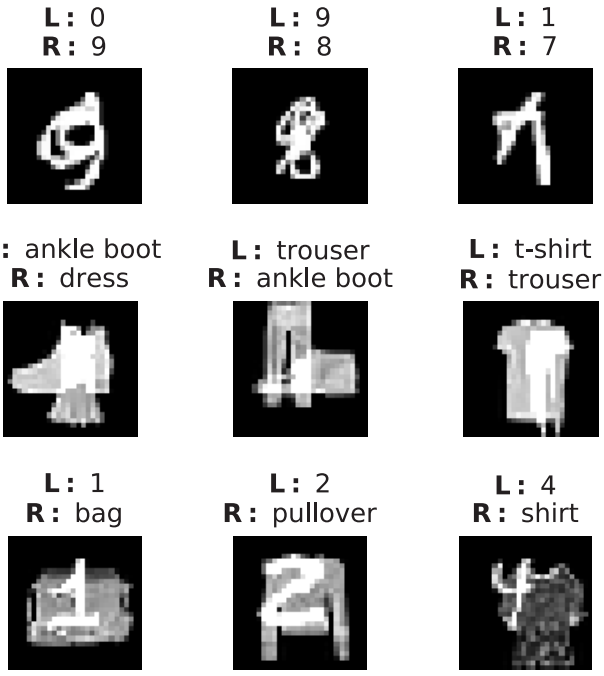

Figure 1. Sample images from MultiMNIST (top), MultiFashion (middle), and MultiFashionMNIST (bottom). Above each image are the labels of the upper-left (L) and lower-right (R) items.

Training We trained all baselines for 30 epochs of SGD. We used 256 as our mini-batch size and set the momentum to 0.9. The learning rate started from 0.01 and decayed with a cosine annealing scheduler.

For our method, we used 50 iterations of MINRES to solve Equation (4) of the main paper with the right-hand side sampled between $\nabla f_1(\boldsymbol{x}_0^*)$ and $\nabla f_2(\boldsymbol{x}_0^*)$. We did not correct the gradients (Equation (5) of the main paper) in this example as we found using the original gradients were more effective.

2.4. UCI Census-Income

Dataset and Task Description UCI Census-Income (Kohavi, 1996) is a demographic dataset consisting of information about around 300,000 adults in the United States. Lin et al. (2019), one of the state-of-the-art baselines, proposed three tasks on this dataset: 1) whether the person's income exceeds \$50K/year, 2) whether the person's education level is at least college, and 3) whether the person is never married. We did not use their first task because the results are highly imbalanced (93.8% of the dataset would have the

same label). Instead, our first task is whether the person's age is greater than or equal to 40. The tasks were evaluated by cross-entropy losses. We converted all categorical data into one-hot vectors and concatenated them along with continuous data into a 487 dimensional feature vector. After removing invalid data, training and test sets have 199,523 and 99,762 samples respectively.

Network Architecture We used a multilayer perceptron (MLP) with two hidden layers of 256 and 128 channels as the shared feature extractor and a fully connected layer as the classifier for each task. We chose ReLU as the nonlinear activation function. This network contains 158,598 trainable parameters in total.

Training We trained all baselines with 30 epochs of SGD and used a mini-batch of size 256 and a momentum of 0.9. The learning rate started from 0.001 and decayed with a cosine annealing scheduler.

For our method, we used 100 iterations of MINRES to solve the tangent direction and gradient correction was not used. The right-hand side of Equation (4) was sampled as follows: for each task f_i, we flipped a coin to determine a binary label $l_i \in \{0, 1\}$. The right-hand side was then the sum of all $\nabla f_i(\boldsymbol{x}_0^*)$ with $l_i = 1$. We skipped a sample if $l_1 = l_2 = l_3$.

2.5. UTKFace

Dataset and Task Description UTKFace (Zhang et al., 2017) is a dataset of over 20,000 face images. Each image has 200×200 pixels and 3 color channels. We considered three tasks on this dataset: 1) predicting the age of each face, 2) classifying the gender, and 3) classifying the race. We used the Huber loss with $\delta = 1$ for task 1 and cross entropy losses for task 2 and 3. We preprocessed the age information by normalizing it to the standard normal distribution. Moreover, each image was resized to 64×64 and each pixel was further normalized with mean values and standard deviations from ImageNet (Deng et al., 2009). We created the training and test set with an 80/20 split of UTKFace. After data cleaning, our training and test sets have 18,964 and 4,741 images respectively.

Network Architecture Our network was built upon a standard ResNet18 (He et al., 2016) by appending a fully connected layer to it for each task. Batch normalization (Ioffe & Szegedy, 2015) was used with a momentum of 0.1. The network contains 11,180,616 trainable parameters.

Training We ran all baseline experiments with 30 epochs of SGD and a mini-batch size 256. We used a weight decay of $1e{-5}$ and a momentum of 0.9. The learning rate started from 0.01 and decayed with a cosine annealing scheduler. Batch normalization was frozen when we were expanding the Pareto front from a Pareto optimal network.

For our method, the training process was the same as in UCI Census-Income except that we used 50 MINRES iterations instead of 100.

3. Synthetic Examples

3.1. ZDT2-variant

Here we present more experimental results on ZDT2-variant from multiple random seeds. Figure 2 left shows 40 random Pareto optimal solutions and expansions from them with tangent directions and gradients. This essentially repeated Figure 2 in the main paper 40 times. It can be seen that tangent directions behave consistently better than gradients in terms of exploring the Pareto front across all random samples. Figure 2 middle and right implemented Algorithm 1 from 10 random seeds and collected 10 Pareto optimal solutions each time. This experiments duplicated Figure 3 in the main paper with 10 different random seeds, and they show that using tangent directions allows us to slide on the Pareto front closely as expected. It is worth noting that part of the solutions optimized by MGDA clustered along the line segment $f_1 = 0, f_2 \geq 1$. Due to the design of ZDT2-variant, solutions like these are not Pareto optimal but Pareto stationary, and thus MGDA could not make further progress from them. Moreover, we report the time cost in Table 1, which confirms that the time saving mostly comes from the near-optimal inputs to `ParetoOptimize`.

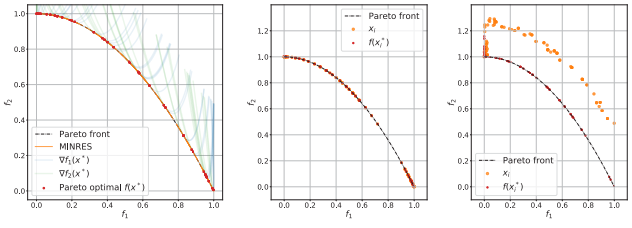

Figure 2. More experimental results on ZDT2-variant. Left: plotting $\boldsymbol{f}(\boldsymbol{x}^* + s\boldsymbol{d}), s \in [-0.1, 0.1]$ with 40 random $\boldsymbol{x}^*$ (red) and $\boldsymbol{d}$ being tangent directions (orange) and gradients (blue and green); Middle and right: running Algorithm 1 with MGDA as the optimizer and comparing two expansion strategies: moving along the tangent directions from MINRES after 2 iterations (middle) and walking along the perturbed weighted sum of gradients (right). The experiments were repeated on 10 random seeds, and all explored points on the Pareto front are colored in red. Results returned by `ParetoExpand` are colored in orange.

3.2. MultiMNIST Subset

We now present more results on MultiMNIST Subset in Figure 3 as we extended the experiments in Figure 4 of the main paper. We sampled 26 Pareto optimal points $\{\boldsymbol{x}_i^*\}$

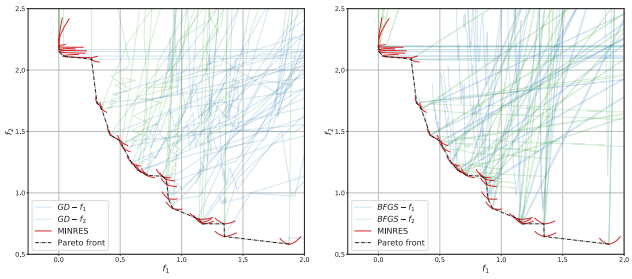

Figure 3. Comparisons of three Pareto expansion strategies on MultiMNIST Subset. The empirical Pareto front is plotted in black with Pareto optimal solutions $\boldsymbol{x}^*$ drawn as 26 red dots. The red curves in both figures show $\boldsymbol{f}(\boldsymbol{x}^* + s\boldsymbol{v}), s \in [-0.5, 0.5]$ where $\boldsymbol{v}$ is the tangent from 50 iterations of MINRES. We used 50 iterations of GD (left) and BFGS (right) to minimize f_1 and f_2 from $\boldsymbol{x}^*$, with intermediate solutions shown in blue and green respectively.

evenly distributed on the empirical Pareto front. For each of them, we depicted $\boldsymbol{f}(\boldsymbol{x}_i^* + s\boldsymbol{v}), s \in [-0.5, 0.5]$ (red) where $\boldsymbol{v}$ is returned by running MINRES after 50 iterations. Furthermore, we minimized f_1 and f_2 from $\boldsymbol{x}_i^*$ with 50 iterations of GD and BFGS and plotted the trajectory of intermediate solutions at each iteration (blue and green). It can be seen from Figure 3 that the tangent directions expanded the empirical Pareto front more accurately and clearly dominated the region explored by GD or BFGS.

4. Pareto Expansion

In this section, we repeated the two experiments described in Section 6.3 of the main paper on all five datasets with more random seeds. Essentially, the results in this sections extend Figure 5 and Figure 6 of the main paper. To recap, the first experiment uses our Pareto expansion method to grow dense Pareto fronts from known Pareto optimal solutions, and the second experiment compares our method to the WeightedSum baseline to establish the necessity of using tangent directions. For simplicity, we will call them sufficiency and necessity experiments respectively.

Table 1. The number of evaluations of objectives ($\boldsymbol{f}$), gradients ($\nabla \boldsymbol{f}$), and Hessian-vector products ($\nabla^2 \boldsymbol{f}$) in Figure 2 middle (ours) and right (WeightedSum). The abbreviation EXP and OPT means the time cost from `ParetoExpand` and `ParetoOptimize` respectively.

METHOD	$\#\boldsymbol{f}$	$\#\nabla \boldsymbol{f}$	$\#\nabla^2 \boldsymbol{f}$
OURS (EXP)	0	50	300
OURS (OPT)	100	100	0
WEIGHTEDSUM (EXP)	0	50	0
WEIGHTEDSUM (OPT)	17931	1818	0

4.1. MultiMNIST and Its Variants

Figure 4 displays the results of our sufficiency experiment on MultiMNIST and its two variants. We grew Pareto fronts from 5 seeds optimized by two baselines: WeightedSum and ParetoMTL. This figure is an extension to Figure 5 in the main paper. We stress again that growing such dense Pareto fronts only took a fraction of the training time spent on getting one Pareto optimal solution from baselines.

Similarly, we reran the necessity experiment and summarized the results in Figure 5. For each dataset, we repeated the experiment on 5 different Pareto optimal solutions found by ParetoMTL (squares and triangles in Figure 4). We second that in all figures, lower left indicates better performances, and the region expanded by our method (orange lines) dominates SGD with various learning rates and weight combinations.

4.2. UCI Census-Income

Figure 6 displays the result of the sufficiency experiment. Note that this dataset has three objectives. We repeated this experiment with 5 random seeds. For each random seed, we ran SGD 10 times with different weight combinations to generate 10 Pareto optimal solutions that are evenly distributed on the Pareto front, which is roughly a concave surface viewed from the camera position. Points with smaller values (farther away from the camera in the figure) are preferred.

Furthermore, Figure 7 summarizes the necessity experiment on this dataset. We first ran SGD to minimize a combination of three objectives with a preference vector $(1/3, 1/3, 1/3)$ to obtain a Pareto optimal solution $\boldsymbol{x}^*$. We then considered three pairs of losses (f_i, f_j) where $(i, j) \in \{(1, 2), (2, 3), (3, 1)\}$. For each (f_i, f_j) pair, we ran MINRES from $\boldsymbol{x}^*$ and compared it to SGD baselines with different weight combinations and learning rates. For all figures, lower left region is Pareto optimal. In most cases, Pareto fronts revealed by our method dominate SGD results.

4.3. UTKFace

The sufficiency experiment is reported in Figure 8. We randomly picked 5 initial networks and ran SGD to minimize a combination of three objectives with a weight vector $(1/3, 1/3, 1/3)$. We then expanded the local Pareto front with our method by running 50 MINRES iterations 6 times, generating 6 trajectories from the Pareto optimal solution. The choice of 6 comes from the fact that three objectives have 8 possible combinations of binary labels (see Section 2.4) and we skipped combinations of all-zero or all-one labels in the sufficiency experiment.

We present the necessity experiment in Figure 9. Since both

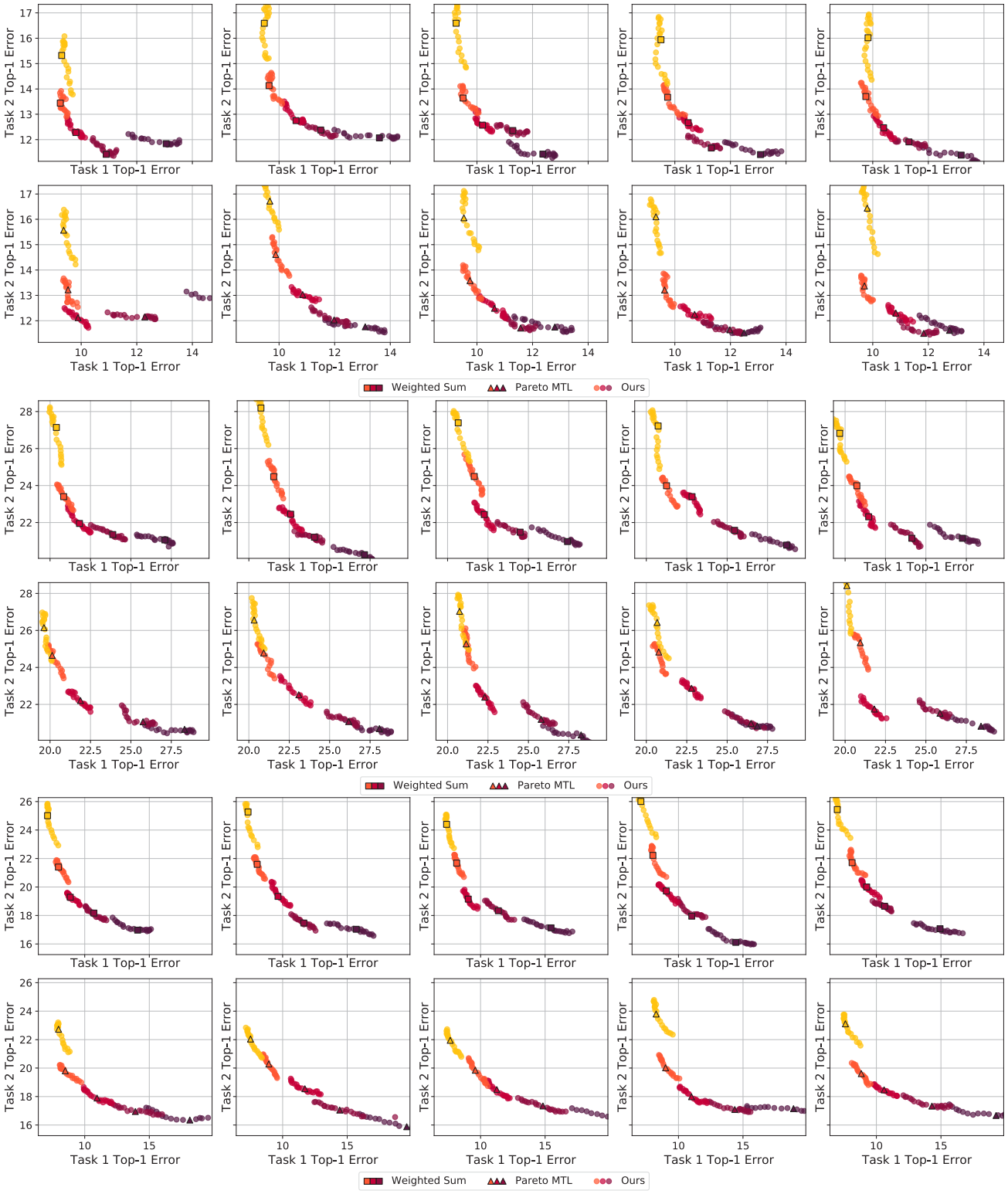

Figure 4. Expanding the Pareto front with our method on MultiMNIST (top two rows), MultiFashion (middle two rows), and MultiFashionMNIST (bottom two rows) from 5 Pareto optimal seeds generated by the WeightedSum baseline (squares) and ParetoMTL (triangles) with different initial random guesses (left to right). Our method grew dense Pareto fronts (colorful circles) from these 5 seeds.

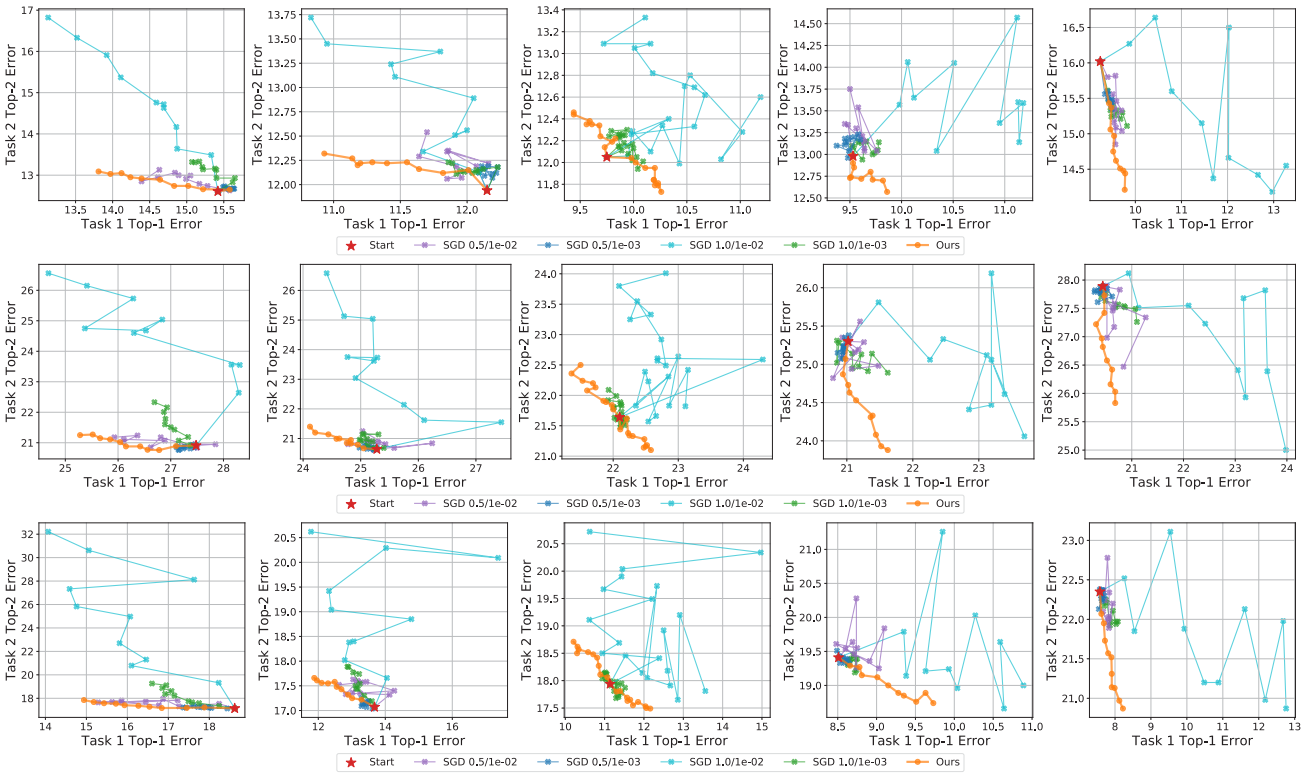

Figure 5. Comparisons of two expansion methods (ours and running SGD with a weighted sum) from a given Pareto optimal solution (red star). Top to bottom: results on MultiMNIST, MultiFashion, and MultiFashionMNIST. Left to right: we started the experiments from five different Pareto optimal solutions found by ParetoMTL. In all figures, lower left means better solutions. All SGD methods are labeled with preference on task 1/learning rate.

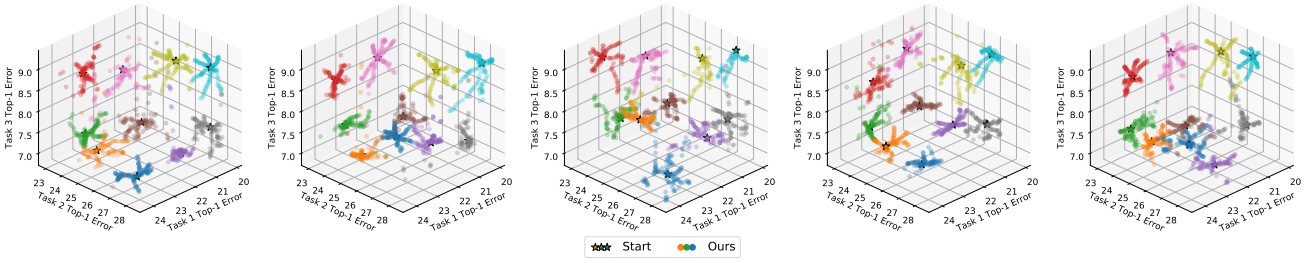

Figure 6. Expanding the Pareto front with our method on UCI Census-Income from 10 Pareto optimal seeds generated by the WeightedSum baseline. Five random initial guesses (left to right) were used to generate these results.

UTKFace and UCI Census-Income have three objectives, we inherited the same experiment setup from UCI Census-Income. Methods that can explore towards the lower left region are preferred. Among the 15 experiments and 4 SGD baselines reported in Figure 9, we summarize that our method almost dominated all SGD baselines in 5 experiments (row 1 column 4, row 2 column 3, and the rightmost column), was clearly outperformed by one SGD baseline in our experiment (purple in row 2 column 4), and performed comparably in the remaining experiments.

5. Continuous Parametrization

In this section, we present results that extend Figure 8 of the main paper. The main idea we want to demonstrate is twofold: locally, we show that Pareto optimal solutions found by our method can be used as backbones to grow a continuous, approximated Pareto front; Globally, such

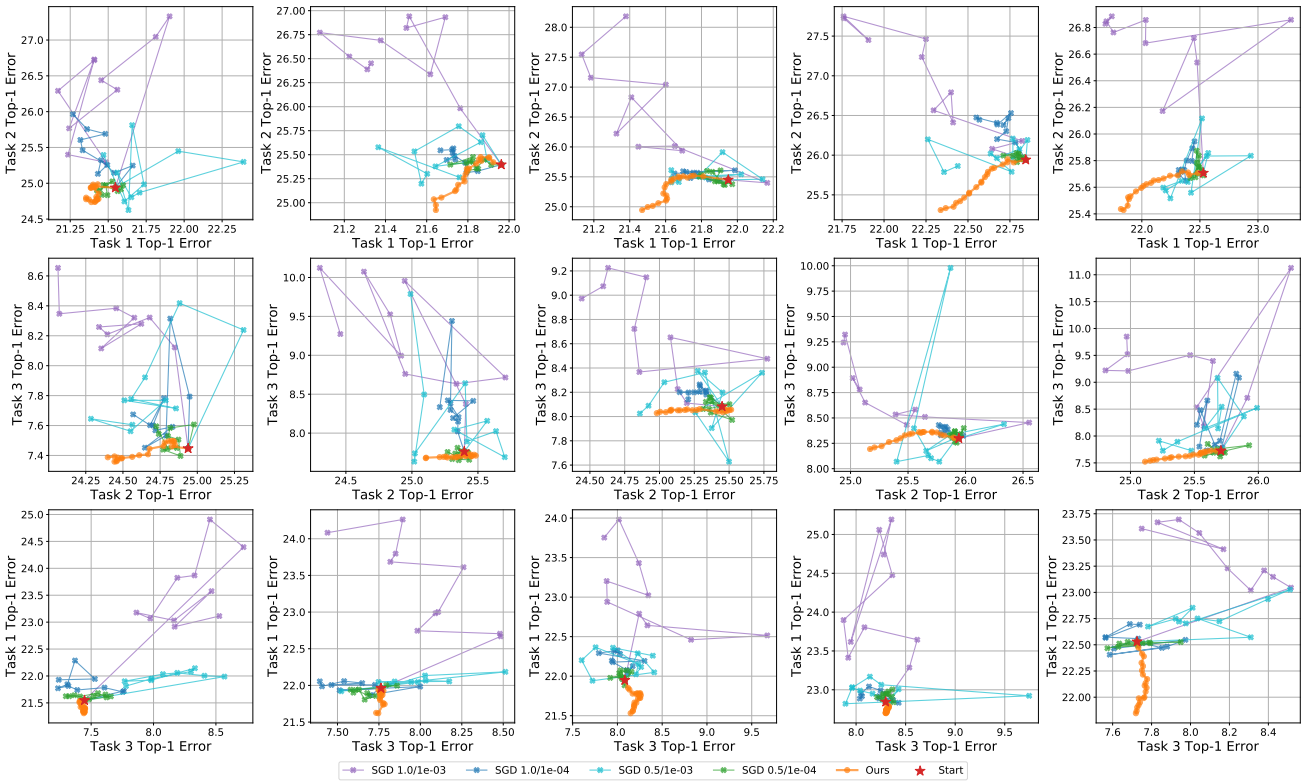

Figure 7. Comparisons of two expansion methods (ours and running SGD with a weighted sum) from a given Pareto optimal solution (red star) on UCI Census-Income. Left to right: we started the experiments from five different Pareto optimal solutions found by SGD with weights $(1/3, 1/3, 1/3)$. In all figures, lower left means better solutions. All SGD methods are labeled with preference on task of the horizontal axis/learning rate.

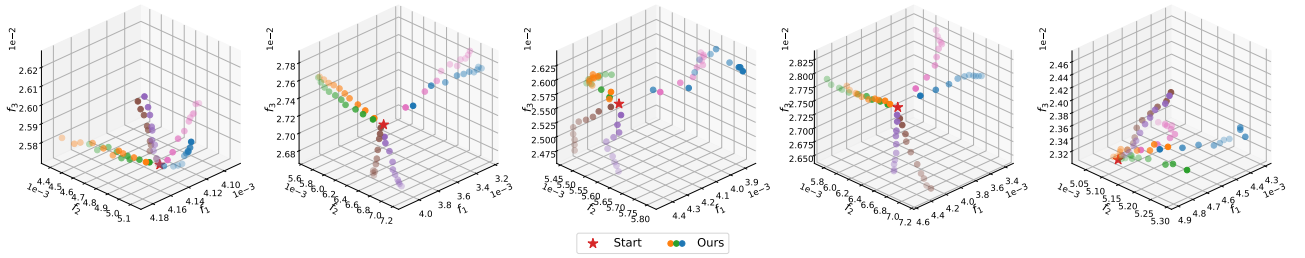

Figure 8. Expanding the Pareto front with our method on UTKFace from five random initializations. We grew our solutions from a seed (red star) to 6 directions (colorful circles) computed by 50 MINRES iterations.

Pareto fronts can be stitched together to cover a wide range of solutions with varying trade-offs.

5.1. MultiMNIST and Its Variants

Figure 10 depicts the continuous parametrization on MultiMNIST and its two variants. For each dataset, we gradually increased the number of Pareto optimal seeds from 3 to 25 and reconstructed a continuous approximation of the Pareto front (a curve in this 2D case) from each seed. It can be seen that as we added more seeds, the continuous Pareto fronts became more connected. By stitching them together, we have created a union of continuous Pareto fronts that offers very diverse choices of trade-offs. We further reparametrized it with a single scalar for easy manipulation and intuitive visualization.

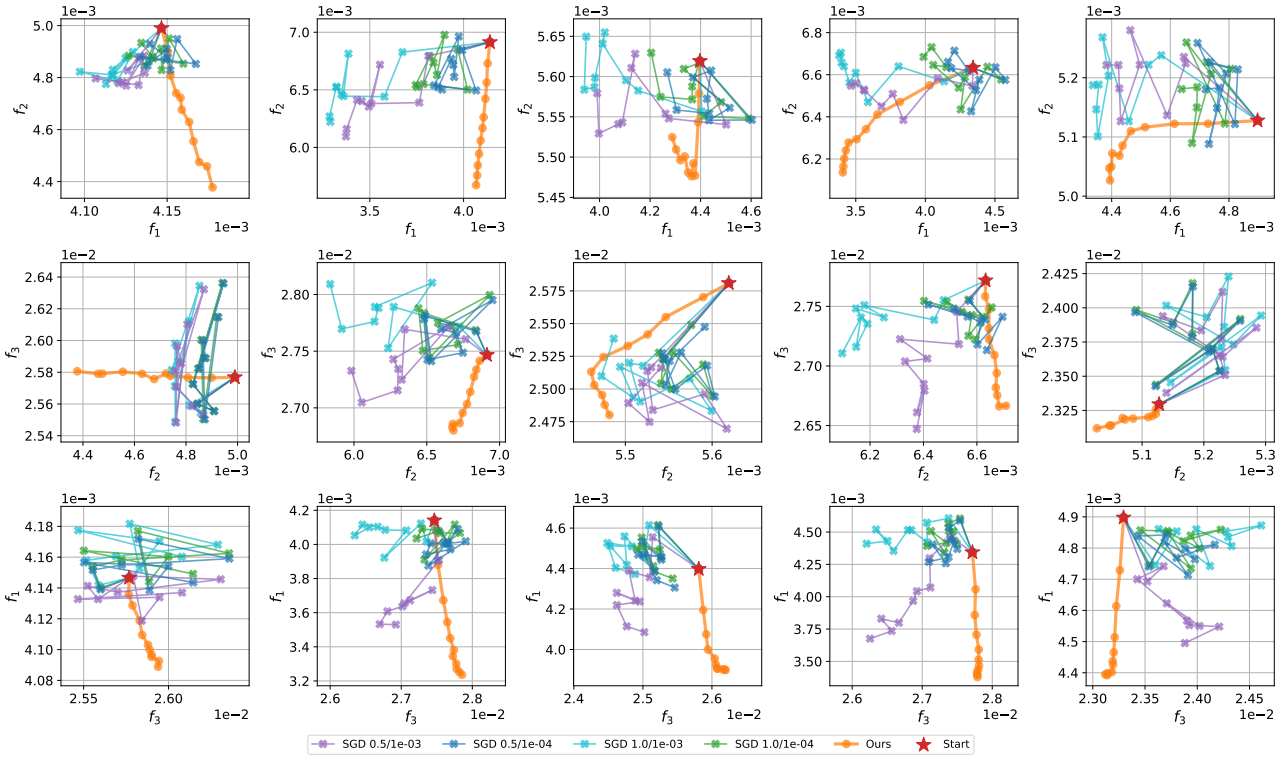

Figure 9. Comparisons of two expansion methods (ours and running SGD with a weighted sum) from a given Pareto optimal solution (red star) on UTKFace. Left to right: we started the experiments from five different Pareto optimal solutions found by SGD with weights $(1/3, 1/3, 1/3)$. In all figures, lower left means better solutions. All SGD methods are labeled with preference on task of the horizontal axis/learning rate.

5.2. UCI Census-Income

We display the continuous parametrization results on UCI Census-Income in Figure 11. We started with 36 Pareto optimal seeds, densely sampled the continuous Pareto set grown from each seed to evaluate their performances, and labeled samples from the same patch with a unique color. We gradually increased the number of samples in order to show how our continuous Pareto fronts were constructed progressively. Additionally, we reconstructed a 3D surface mesh from the Pareto fronts for better visualization.

5.3. UTKFace

Figure 12 shows the continuous parametrization results on UTKFace. The setup and visualization is the same as in UCI Census-Income except that the continuous Pareto front was reconstructed from only 1 Pareto optimal seed. Therefore, a single color was used for all samples.

6. Ablation Study

Finally, we present more results of ablation study on MultiMNIST and its two variants in Figure 13. For each dataset, we ran ParetoMTL to generate 5 Pareto optimal solutions that are evenly distributed on the Pareto front. From each solution, we conducted the ablation study on hyperparameters k and s as described in the main paper and produced one column of Figure 13. It can be seen that our claims in the main paper on the influence of k and s are consistently observed across these 5 solutions with various trade-offs on these datasets.

References

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Désidéri, J.-A. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

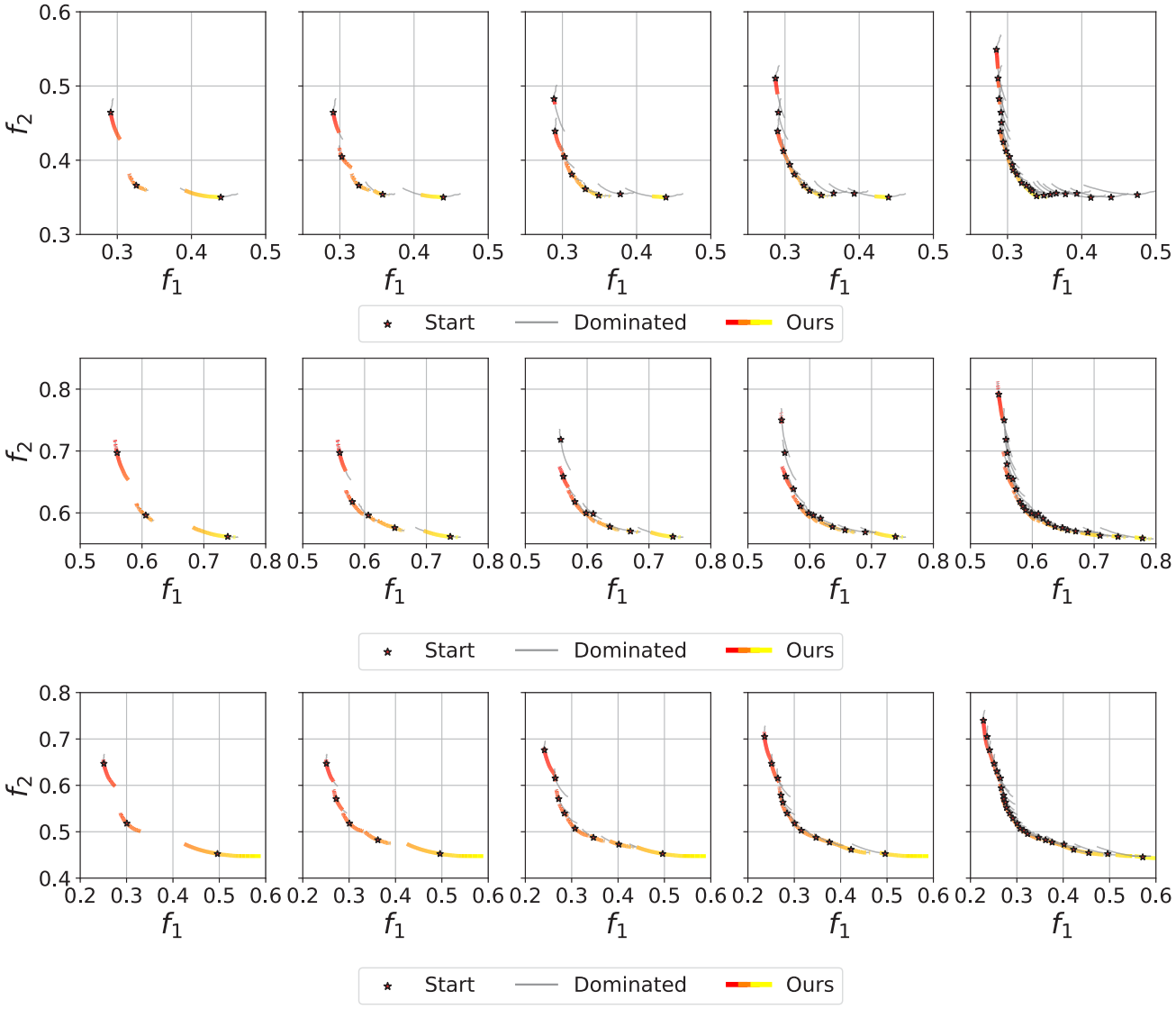

Figure 10. Continuous parametrization on MultiMNIST (top), MultiFashion (middle), and MultiFashionMNIST (bottom). From left to right: we gradually increased the number of Pareto optimal solutions (red stars) obtained from running SGD with different weights. We then ran Algorithm 1 to grow a continuous Pareto front from each solution (colorful circles) and filtered out dominated solutions (gray). The red-to-yellow color indicates the value of the scalar parameter that traverses the whole final Pareto front.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lin, X., Zhen, H.-L., Li, Z., Zhang, Q.-F., and Kwong, S. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, pp. 12037–12047, 2019.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing

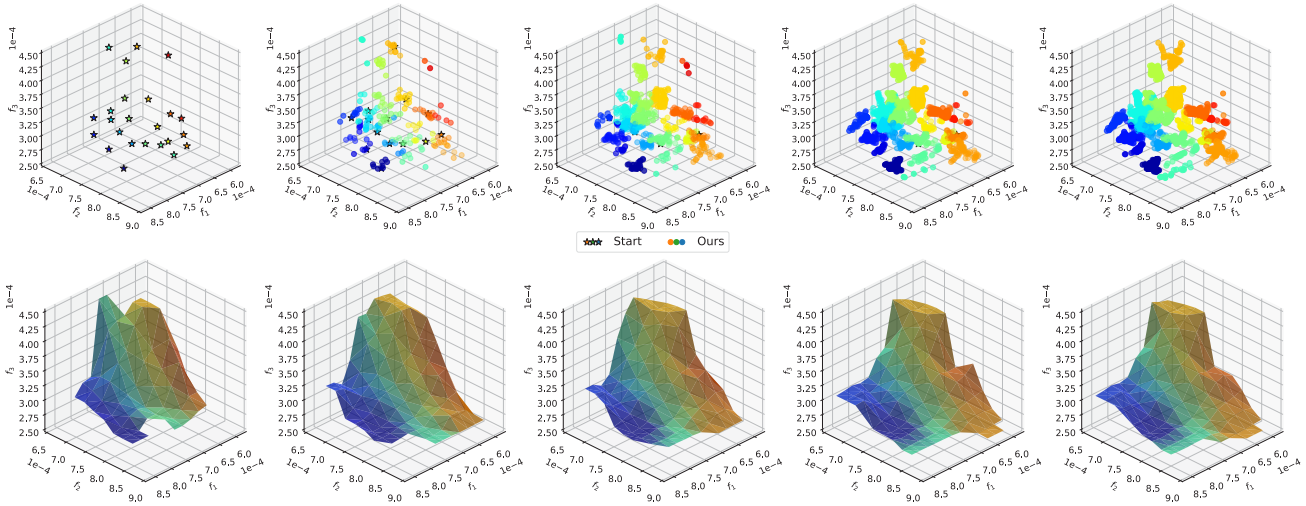

Figure 11. Continuous parametrization on UCI Census-Income. Top: starting with discrete Pareto optimal seeds returned by running SGD with various weights on objectives (colorful stars), we constructed the continuous Pareto sets (not shown) with our method, densely sampled new solutions from these Pareto sets, and plotted their performances (colorful circles). Samples from the same seed share the same color. Bottom: we reconstructed a continuous surface mesh to approximate the Pareto front revealed by these samples above. The color on the surface mesh has a one-to-one correspondence to the color of Pareto optimal seeds; Left to right: We gradually increased the number of samples to show the progress of our reconstruction.

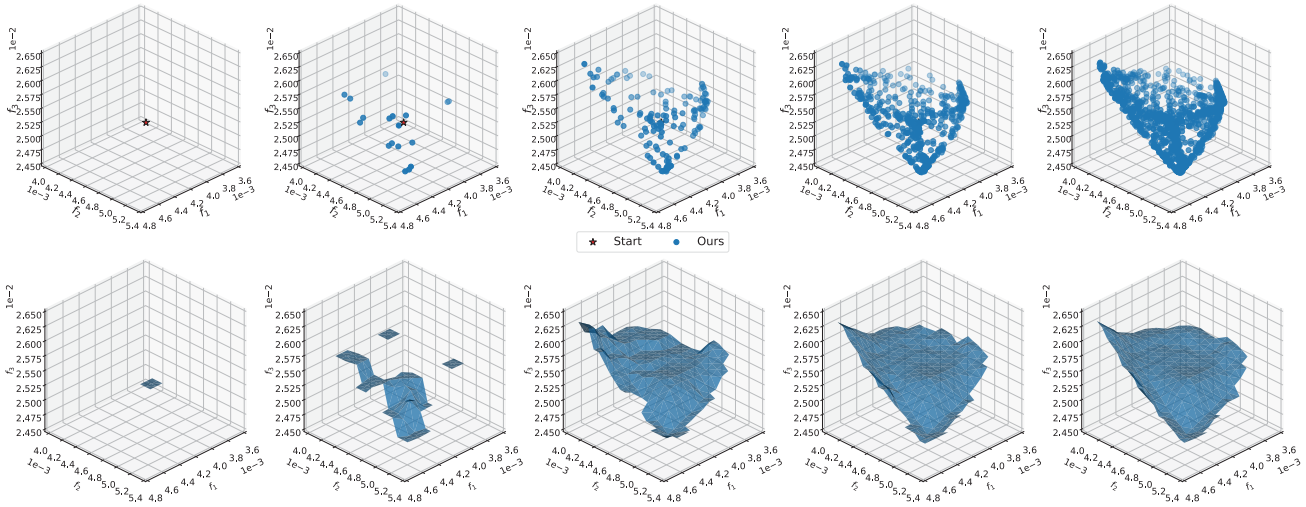

Figure 12. Continuous parametrization on UTKFace. The setup is identical to Figure 11 except that one Pareto seed obtained from ParetoMTL was used.

between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zhang, Z., Song, Y., and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5810–5818, 2017.

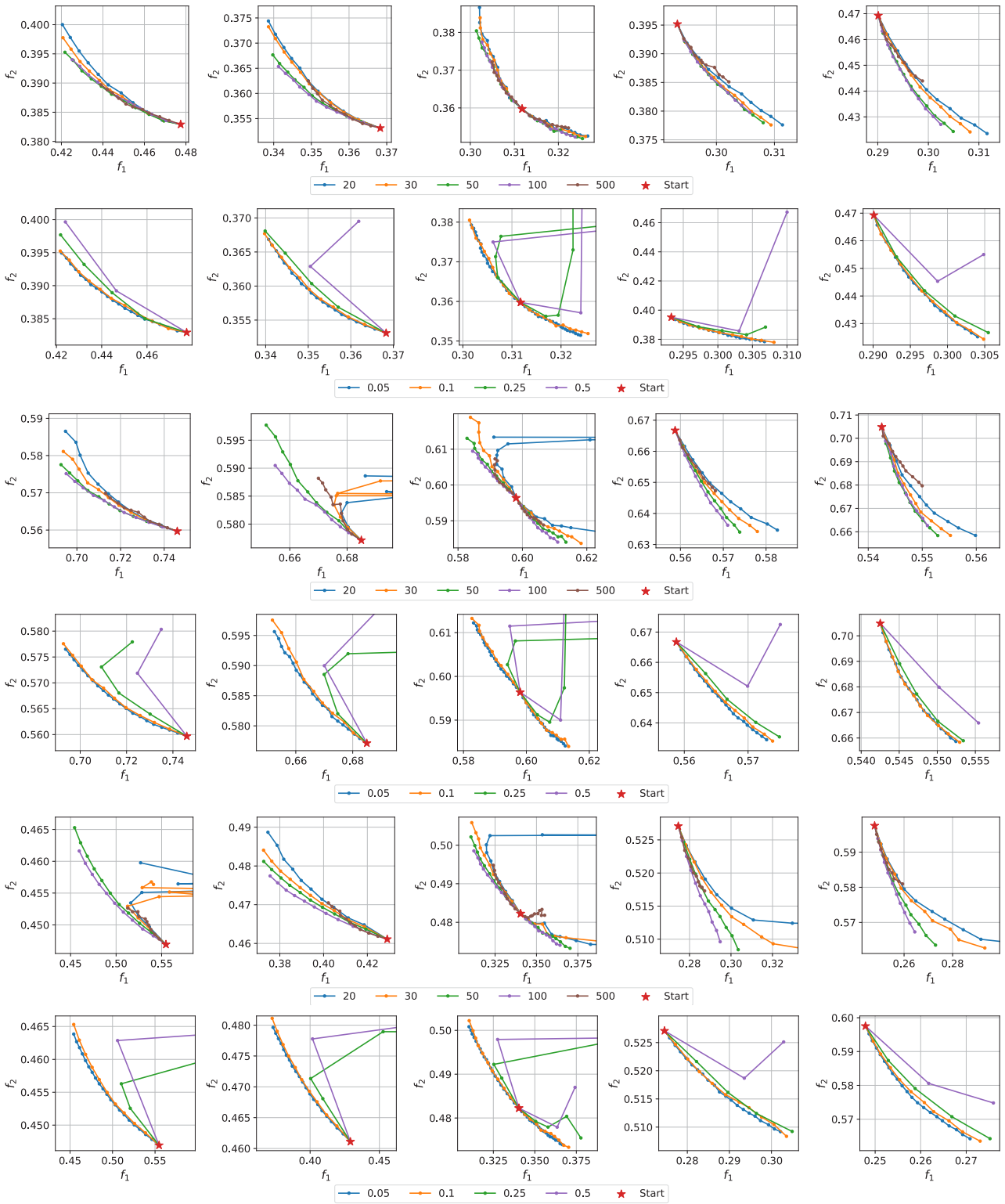

Figure 13. Ablation study on the maximum number of MINRES iterations k and the step size s on MultiMNIST (top two rows), MultiFashion (middle two rows), and MultiFashionMNIST (bottom two rows). We repeated the experiments from different Pareto optimal solutions returned by ParetoMTL (left to right).