IOP: An Idempotent-Like Optimization Method on the Pareto Front of Hypernetwork

Hui Wang[1], **Renyu Yang**[1*], **Jie Sun**[2], **Hao Peng**[1], **Xudong Mou**[1,2], **Tianyu Wo**[1,2], **Xudong Liu**[1,2*]

[1]Beihang University, Beijing, China
[2]Zhongguancun Laboratory, Beijing, China
{whui,renyuyang,penghao,mxd,woty,liuxd}@buaa.edu.cn; sunjie@zgclab.edu.cn

Abstract

Pareto Front Learning (PFL) has been one of the effective means to resolve multi-objective optimization problems through exploring all optimal solutions to learn the entire Pareto front. Pareto Hypernetwork (PHN) is a new promising way to generate the sequence of Pareto-optimal solutions that can be further used as potential solutions to constitute the Pareto front. However, the existing PHN-based approaches suffer from two performance issues: They take as inputs human-crafted preference vector or chunk embedding, rather than the input data samples, and thus vulnerable to data distribution shifts. Such approaches cannot optimize all potential solutions when forming the Pareto front, as they merely optimize the loss pertaining to one single input at a time of optimization round. To improve the quality of the Pareto front, we propose IOP, a novel Idempotent-like Optimization method to learn the entire Pareto front accurately and enhance Hypernetwork's adaptability to distribution shifts. In particular, IOP performs idempotent-like optimization by exploiting manifold space mapping, so that the target networks generated by the optimized Hypernetwork can effectively handle samples with similar distributions of the input samples, without the pre-defined human-crafted inputs. IOP maximizes the Hypervolume indicator that is composed of all potential solutions at a higher level. Experimental results demonstrate that IOP outperforms the state-of-the-art methods by 4.7% on average in producing the Pareto front and has a 10.5% improvement in adaptability.

Introduction

Multi-objective optimization (MOO) is of paramount importance – It unleashes the potential of acquiring optimal trade-offs among multiple conflicting objectives in various machine learning domains such as Multi-object Tracking (Nguyen et al. 2020), Computer Vision (Gao et al. 2024), Reinforcement Learning (Chen et al. 2024), Natural Language Processing (Xiang et al. 2022), etc. In MOO, the collection of all optimal solutions – commonly referred to as the Pareto front – manifests distinct trade-offs amidst conflicting objectives. Most of the existing MOO approaches solely obtain a partial set of optimal solutions on the Pareto front, and typically require a pre-defined strategy for effectively balancing multiple objectives. This setting, however, substantially limits the usability and flexibility, as the decision-making procedure, in most of the use cases, is on the fly without prior knowledge of such specific strategies.

To tackle this issue, Pareto Front Learning (PFL) emerges as an effective paradigm. All the optimal solutions spanning the entire Pareto front can be simultaneously resolved. Pareto Hypernetwork (PHN) (Navon et al. 2020) is a notable implementation of PFL, and it employs a single Hypernetwork (Ha et al. 2016) to take as inputs the preference vectors (Hoang et al. 2023) or chunk embeddings (Lin et al. 2020) and generate a sequence of Pareto-optimal models (aka. target networks) to process data. Although PHN-based approaches exhibit competitive efficiency, there are the following unsettled issues:

i) Relying on human—crafted inputs with limited adaptability to data distribution shift. A huge body of PHN studies depend on human-crafted inputs – often in the form of preference vector or chunk embedding and entirely independent from raw data samples – to train a precise Hypernetwork to acquire optimal target networks. The key concern is that devising high-quality inputs usually requires too much domain expertise and is sometimes impractical in industrial scenarios at scale. Additionally, rather than adopting raw data samples, using man-made inputs inevitably reduces the sensitivity of Hypernetwork to data distribution shifts. For example, as shown in Fig. 1, when the data shifts from the *green* to the *blue* distribution, the Hypernetwork in PHN methods generates more non-optimal solutions. Hence, it is imperative to establish a new competitive and adaptable Hypernetwork in the PHN method free from human-crafted inputs.

ii) Failing to simultaneously optimize multiple potential solutions. PHN methods optimize the Hypernetwork so that each output is associated with an optimal or approximate solution on the Pareto front. The ideal process is to continuously bring all potential solutions to the Pareto front as close as possible. However, the existing PHN methods fail to do so – in each optimization round, their focus is merely minimizing the loss related to the single input and accordingly updating the Hypernetwork's weight. Doing this can hardly accommodate multiple potential solutions around the Pareto front simultaneously; in other words, some solutions move towards but other potential solutions go away. As a working example, in Fig. 1, optimizing the solutions in the dotted

[*] Corresponding authors.

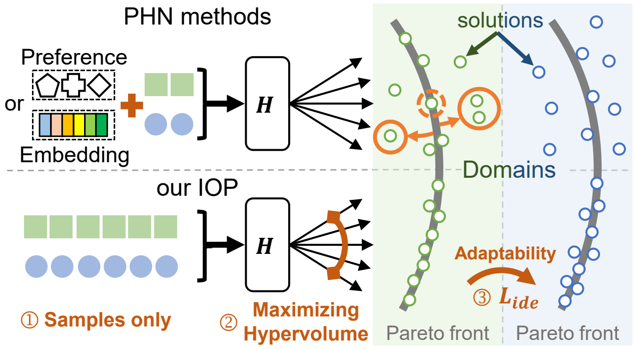


Figure 1: Compared with PHN methods, IOP is free from man-made input and focuses on the relationship among solutions. It comprehensively improves the accuracy by maximizing Hypervolume, and enhances the adaptability of Hypernetwork through the idempotent-like objective L_{ide}.

circle pushes the solutions in the solid circle away from the Pareto front. Undoubtedly, this will substantially diminish the quality of the obtained front and increase the optimization difficulty.

This paper proposes IOP, a novel Idempotent-like Optimization method for PHN. The essence of IOP is devising an idempotent-like objective to optimize the Hypernetwork, in the light of manifold mapping from the sample space $\mathcal{P}_x$ to the target network weight space $\mathcal{P}_\theta$. Without the need of any human-crafted inputs, the optimized Hypernetwork can generate target networks that effectively handle samples with similar distributions of the input samples. We control the range of manifold $\mathcal{P}_\theta$ by using a scaling factor in the objective to *resize* the "similar distributions", thereby improving the adaptability of Hypernetwork. Unlike prior works, our Hypernetwork iteratively receives multiple samples in each optimization round to improve the performance of corresponding target networks (aka. potential solutions). At the core of the improvement is to maximize the Hypervolume indicator that is composed of all potential solutions at a higher level. IOP leverages a gradient-based dual optimization strategy to jointly incorporate the techniques above simultaneously. Extensive experiments are conducted on six multi-task and domain adaptation datasets, and experimental results show the superior effectiveness of IOP against the state-of-the-art approaches. Our main contributions can be summarized as follows:

- We are the first to reveal the main limitations of existing PHN methods, and propose a new approach to improve Hypernetwork performance and adaptability.
- We devise an idempotent-like objective and use a manifold mapping mechanism to iteratively handle any samples that have similar distributions of the input data samples.
- Providing an in-depth theoretical analysis on the convergence of the optimization process of IOP.

Related Work

Multi-Objective Optimization (MOO) methods aim to balance multiple conflicting objectives for optimal solutions (Ehrgott 2005). They can be roughly classified into gradient-based and gradient-free categories. (Sener 2018) proposes an early gradient-based approach for Multi-Task Learning (MTL), extended from MGD (Désidéri 2012). (Lin et al. 2019) proposes Pareto MTL (PMTL) to split the loss space into separate cones. (Mahapatra 2020) proposes EPO, extended from PMTL, to combine gradient descent and controlled ascent to improve optimization convergence. (Ma et al. 2020) proposes a gradient-free approach to extend an optimal solution to its local neighborhood. (Ye et al. 2022) proposes PNG to find solutions by optimizing an extra criterion function. However, the above traditional methods can only obtain a single or partial optimal solution on the Pareto front, leaving the MOO problem to be thoroughly solved.

Pareto Front Learning (PFL) has become the most effective paradigm for MOO by seeking all the optimal solutions and learning the entire Pareto front. Early work (Lin et al. 2019) on PFL trains a Hypernetwork (Ha et al. 2016), mainly through changing the reference directions in the objective so as to achieve a complete fitting of the Pareto front. (Lin et al. 2020; Navon et al. 2020; Ruchte et al. 2021) further use preference or embedding as inputs of Hypertnetwork to improve the controllability of optimization. (Hoang et al. 2023) employs a Hypernetwork to generate multiple solutions from diverse trade-off preferences. However, all of these PFL works still heavily depend upon non-sample inputs and neglect the adaptability of Hypernetwork to data distribution shift. Our work aims to jointly overcome these shortcomings and bring new insights into PFL.

Methodology

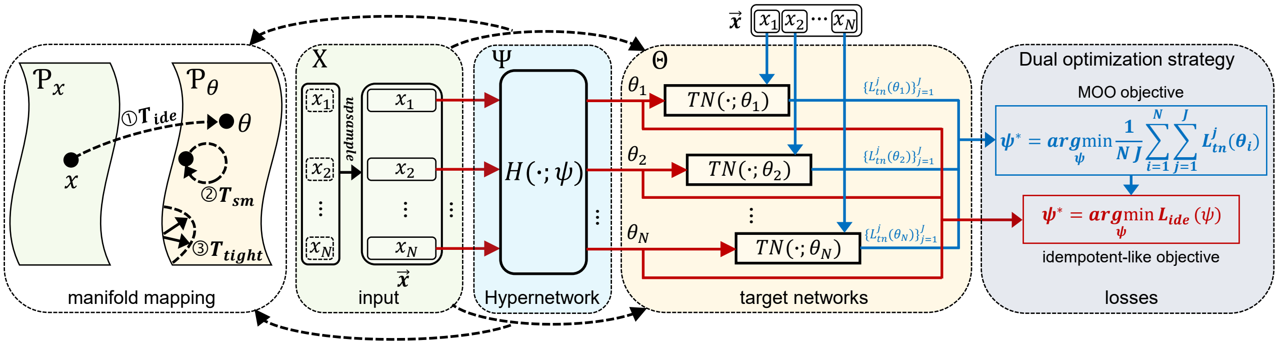
Problem Formulation

As shown in Fig. 2, our Hypernetwork $H(\cdot;\psi), \psi \in \mathbb{R}^q$ receives an input vector $\mathbf{x} = [x_1, ..., x_N], x_i \in \mathbb{R}^d, i \in [N]$ in each training round, and generates N target networks $\{TN(\cdot;\theta_i) : \theta_i = H(x_i;\psi), \theta_i \in \mathbb{R}^d\}_{i=1}^N$. Assuming target network weights and samples have the same dimension, i.e., $\dim(x_i) = \dim(\theta_i), \forall i \in [N]$, which can be achieved by upsampling on x_i, or by modifying the structure of $TN(\cdot;\theta_i)$. $\{L_{tn}^j(\theta_i)\}_{j=1}^J$ is denoted as the J loss functions of $TN(\cdot;\theta_i)$:

$$L_{tn}^j(\theta_i) = l_{tn}^j(\theta_i) - \beta[\cos(\vec{x}_i, \overrightarrow{\Gamma(\theta_i)}) + HV(\Upsilon)] \ , \qquad (1)$$

where $l_{tn}^j(\theta_i)$ is the inner loss function such as cross-entropy, $\Gamma(\theta_i)$=$(l_{tn}^1(\theta_i), ..., l_{tn}^J(\theta_i))$, Υ=$[\Gamma(\theta_1), ..., \Gamma(\theta_N)]$. $HV(\cdot)$ denotes the Hypervolume (Zitzler 1999), the area dominated between the Pareto front and a specified reference point. $\beta \in (0, 0.5]$ is the coefficient. Then the MOO of Hypernetwork can be expressed as follows:

$$\psi^* = \arg\min_{\psi \in \mathbb{R}^q} \mathbb{E}_{x_i \sim \mathcal{P}_x^i} \frac{1}{NJ} \sum_{i=1}^N \sum_{j=1}^J L_{tn}^j(\theta_i) = \arg\min_{\psi \in \mathbb{R}^q} \mathbb{E}_{x_i \sim \mathcal{P}_x^i} \frac{1}{NJ} \sum_{i=1}^N \sum_{j=1}^J l_{tn}^j(\theta_i) - \frac{\beta}{N} \sum_{i=1}^N \cos(\vec{x}_i, \overrightarrow{\Gamma(\theta_i)}) - \beta HV(\Upsilon), \qquad (2)$$



$\vec{x}$ x_1 x_2 $\cdots$ x_N
$\mathcal{P}_x$
$\mathcal{P}_\theta$
①T_{ide}
θ
x
②T_{sm}
③T_{tight}
manifold mapping
X
x_1
x_2
x_N
upsample
$\vec{x}$
input
Ψ
$H(\cdot;\psi)$
Hypernetwork
Θ
θ_1
θ_2
θ_N
$TN(\cdot;\theta_1)$
$TN(\cdot;\theta_2)$
$TN(\cdot;\theta_N)$
$\{L^j_{tn}(\theta_1)\}^J_{j=1}$
$\{L^j_{tn}(\theta_2)\}^J_{j=1}$
$\{L^j_{tn}(\theta_N)\}^J_{j=1}$
target networks
Dual optimization strategy
MOO objective
$\psi^* = \arg\min_{\psi} \frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J} L^j_{tn}(\theta_i)$
$\psi^* = \arg\min_{\psi} L_{ide}(\psi)$
idempotent-like objective
losses

Figure 2: The framework of IOP. In each training round, the Hypernetwork processes a batch of samples and generates target networks (aka. solutions). The objectives include MOO loss L_{tn} (blue), which focuses on improving all solution performance via the Hypervolume term defined by solutions, and idempotent-like loss L_{ide} (red), which enhances Hypernetwork adaptability by controlling solutions' distribution "size" (manifold range). Both objectives are unified through a dual optimization strategy.

where $HV(\cdot) : \mathbb{R}^N \times \mathbb{R}^J \rightarrow \mathbb{R}_+$ is a monotonically decreasing function. For a fixed reference point, a smaller $\{l^j_{tn}(\theta_i)\}^J_{j=1}$ leads to a larger $HV(\cdot)$. If $HV(\Upsilon)$ is maximized, the current solution ψ will be on the Pareto front, which is also beneficial for improving the performance of other potential solutions (Hoang et al. 2023). Herein, $\cos(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^J \rightarrow \mathbb{R}$ denotes the cosine similarity function. It helps to spread the Pareto front and brings the solution closer to the input (Ye et al. 2022), as well as strengthens the correlation between $H(\cdot;\psi)$ and $\vec{x}_i$, improving Hypernetwork's sensitivity to samples. If the dimensions of $\vec{x}_i$ and $\overrightarrow{\Gamma(\theta_i)}$ differ, we perform zero padding on $\Gamma(\theta_i)$, which has proven effective in (Mahapatra 2020).

In addition, we design an idempotent-like objective $L_{ide}(\psi)$ to enhance Hypernetwork's adaptability. We unify the $L_{ide}(\psi)$ and MOO objective in Eq. 2 through a **d**ual **o**ptimization **s**trategy $\mathcal{DOS}$. More details about $L_{ide}(\psi)$ and $\mathcal{DOS}$ will be discussed in subsequent sections. Let denote $L_{tn}(\psi)=\frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J} L^j_{tn}(\theta_i)$ (Eq. 2), the optimization of Hypernetwork in IOP can finally be formalized as:

$$\psi^* = \underset{\psi\in\mathbb{R}^q}{\arg\min}\, \mathcal{DOS}(L_{tn}(\psi), L_{ide}(\psi)) \,. \tag{3}$$

Idempotent-Like Optimization Objective

We design the idempotent-like objective $L_{ide}(\psi)$ to enhance Hypernetwork's adaptability from the perspective of manifold mapping (Shocher et al. 2023). As shown in Fig. 2, the Hypernetwork $H(\cdot;\psi)$ is trained to generate target network weight $\theta \sim \mathcal{P}_\theta$ when given $x \sim \mathcal{P}_x$. We assume the instances in sample distribution $\mathcal{P}_x$ and target distribution $\mathcal{P}_\theta$ have the same dimension. This allows applying $H(\cdot;\psi)$ to both x and θ. Our $L_{ide}(\psi)$ follows the following principles:

(1) For any $x \sim \mathcal{P}_x$, the $H(\cdot;\psi)$ maps it to $\mathcal{P}_\theta$, i.e., $H(x;\psi) = \theta$. We hope that $H(\cdot;\psi)$ can be applied sequentially multiple times and reduce the difference in results from the initial application (i.e., idempotence-like), which endows $H(\cdot;\psi)$ to fine-tune based on its original performance to achieve adaptability to distribution shifts. So $L_{ide}(\psi)$ contains the loss term $T_{ide}(\psi) = \|H(H(x;\psi);\psi) - H(x;\psi)\|_2$, $\|\cdot\|_2$ denotes the L_2 distance. We use the weight-freezing technique (Shocher et al. 2023) to reduce the difficulty of optimizing $T_{ide}(\psi)$ and modify it to $T_{ide}(\psi;\hat{\psi}) = \|H(H(x;\psi);\hat{\psi}) - H(x;\psi)\|_2$, where $\hat{\psi}$ denotes the frozen copy of ψ, i.e., a gradient taken ψ will not affect $\hat{\psi}$.

(2) For any $\theta \sim \mathcal{P}_\theta$, the $H(\cdot;\psi)$ should try to map it to itself, i.e., $H(\theta;\psi) = \theta$. Therefore, $L_{ide}(\psi)$ contains the self-mapping loss term $T_{sm}(\psi) = \|H(\theta;\psi) - \theta\|_2$.

(3) Shrinking $\mathcal{P}_\theta$ can improve the accuracy of the obtained θ, enhancing the performance of target network. However, it inevitably reduces the diversity of θ, affecting the adaptability of Hypernetwork. Therefore, we control the range of $\mathcal{P}_\theta$ through a controllable manifold contraction term $T_{tight}(\psi;\hat{\psi}) = -\|H(H(x;\hat{\psi});\psi) - H(x;\hat{\psi})\|_2$. This technique was first proposed in (Shocher et al. 2023), and we extend it to the cross-space mapping from $\mathcal{P}_x$ to $\mathcal{P}_\theta$. In summary, we modify $L_{ide}(\psi)$ to $L_{ide}(\psi;\hat{\psi})$ and define it as:

$$\begin{aligned} L_{ide}(\psi;\hat{\psi}) = \exp(\, & \|H(H(x;\psi);\hat{\psi}) - H(x;\psi)\|_2 + \\ & \|H(\theta;\psi) - \theta\|_2 - \lambda_{ide}\|H(H(x;\hat{\psi});\psi) - H(x;\hat{\psi})\|_2 \,) \,, \end{aligned} \tag{4}$$

where $\exp(\cdot)$ is a monotonic nonnegative operator, and $\lambda_{ide} \in (0,1]$ is the contraction coefficient of $T_{tight}(\psi;\hat{\psi})$. When $\lambda_{ide} \rightarrow 0$, $\mathcal{P}_\theta$ will not shrink. When $\lambda_{ide} \rightarrow 1$, $T_{tight}(\psi;\hat{\psi})$ is fully expressed, and $\mathcal{P}_\theta$ is totally tightened. Notably, when using gradient-based methods to optimize Eq. 4, the gradient is related to ψ but independent of $\hat{\psi}$.

Theorem 1. *If the capacity of $H(\cdot;\psi)$ is large enough, the terms $\exp(T_{sm}(\psi) - \lambda_{ide}T_{tight}(\psi;\hat{\psi}))$ and $\exp(T_{ide}(\psi;\hat{\psi}))$ in Eq. 4 will have common global minima. Assuming $\psi^* = \underset{\psi\in\mathbb{R}^q}{\arg\min}\exp(T_{sm}(\psi) - \lambda_{ide}T_{tight}(\psi;\psi^*)) = \underset{\psi\in\mathbb{R}^q}{\arg\min}\exp(T_{ide}(\psi;\psi^*))$, then $\exists\, \psi^*$ such that $\mathcal{P}_{\psi^*} = \mathcal{P}_\theta$.*

Dual Optimization Strategy

We use a gradient-based dual optimization strategy ($\mathcal{DOS}$) to unify the MOO objective $L_{tn}(\psi)$ in Eq. 2 and idempotent-

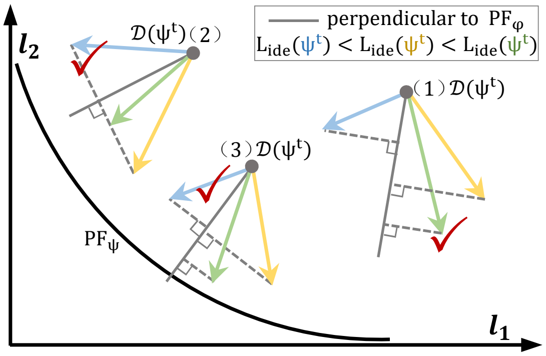

perpendicular to PF_φ
$L_{ide}(\psi^t) < L_{ide}(\psi^t) < L_{ide}(\psi^t)$
l_2
$\mathcal{D}(\psi^t)$ (2)
(1) $\mathcal{D}(\psi^t)$
(3) $\mathcal{D}(\psi^t)$
PF_ψ
l_1

Figure 3: The dual optimization strategy.

like objective $L_{ide}(\psi; \hat{\psi})$ in Eq. 4. Specifically, the gradient update strategy in the t-th $\in [T]$ training round is:

$$\psi^{t+1} \leftarrow \psi^t - \eta \times \mathcal{D}(\psi^t) \; , \tag{5}$$

where η is the learning rate, $\mathcal{D}(\psi^t)$ denotes the gradient of Hypernetwork. As shown in Fig. 3, $\mathcal{DOS}$ includes: (1) When ψ^t is far from the Pareto front PF_ψ, select $\mathcal{D}(\psi^t)$ that achieves the maximum improvement towards PF_ψ, focusing on MOO trade-off. (2) When there are different improvement directions for ψ^t towards PF_ψ, choose the $\mathcal{D}(\psi^t)$ to minimize $L_{ide}(\psi; \hat{\psi})$. (3) When ψ^t approaches PF_ψ, only focus on decreasing $L_{ide}(\psi; \hat{\psi})$. For target network $TN(\cdot; \theta_i), \theta_i = H(x_i; \psi^t), i \in [N]$, we obtain $\mathcal{D}(\psi^t)$ by solving the following optimization problem (Ye et al. 2022).

$$\mathcal{D}(\psi^t) = \arg\min_{\mathcal{G}\in\mathbb{R}^q} \{\frac{1}{2}\|\nabla L_{ide}(\psi^t; \hat{\psi}^t) - \mathcal{G}\|_2^2\}$$
$$s.t. \;\; \nabla L^j_{tn}(\theta_i)^T \mathcal{D}(\psi^t) \geq |\cos(\vec{x}_i, \overrightarrow{\Gamma(\theta_i)})| \cdot S(\psi^t), j \in [J] \; , \tag{6}$$

where $L^j_{tn}(\theta_i)$ denotes the loss function of target network defined in Eq. 1, $\Gamma(\theta_i) = (l^1_{tn}(\theta_i), ..., l^J_{tn}(\theta_i))$, $S(\psi^t)$ represents the Pareto stationary (Lin et al. 2020) of ψ^t. The operator $\|\cdot\|_2^2$ forces the gradient $\mathcal{G}$ of Hypernetwork to approach the idempotent-like objective gradient $\nabla L_{ide}(\psi^t; \hat{\psi}^t)$. Unlike (Ye et al. 2022), we strengthen the constraint lower bound, the gradient descent rate of losses $\{L^j_{tn}(\theta_i)\}_{j=1}^J$ are constrained by $|\cos(\vec{x}_i, \overrightarrow{\Gamma(\theta_i)})| \cdot S(\psi^t)$. Since $\vec{x}_i$ is deterministic, $|\cos(\vec{x}_i, \overrightarrow{\Gamma(\theta_i)})|$ will vary between 0 and 1 in the optimization, which provides an intermediate updating direction between gradient descent on $L_{ide}(\psi^t; \hat{\psi}^t)$ and multiple gradient descent (Désidéri 2012) on $\{L^j_{tn}(\theta_i)\}_{j=1}^J$, making the optimization easier towards PF_ψ. In addition, introducing the Pareto stationary term $S(\psi^t)$ in constraint will provide a dynamic improvement based on the distance between ψ^t and PF_ψ, which has been proven more efficient (Ye et al. 2022). The Eq. 6 can be transformed into a dual form:

$$D(\psi^t) = \nabla L_{ide}(\psi^t; \hat{\psi}^t) + \sum_{j=1}^{J} \lambda_{j,t} \nabla L^j_{tn}(\theta_i) \; , \tag{7}$$

with $\{\lambda_{j,t}\}_{j=1}^J$ the solution of the following problem:

$$\max_{\lambda\in\mathbb{R}^J_+} \sum\nolimits_{j=1}^J \lambda_j \cos(\vec{x}_i, \overrightarrow{\Gamma(\theta_i)}) - \frac{1}{2}\|\nabla L_{ide}(\psi^t; \hat{\psi}^t) + \sum\nolimits_{j=1}^J \lambda_t \nabla L^j_{tn}(\theta_i)\|_2^2 \; , \tag{8}$$

we follow (Ye et al. 2022) and initialize $\{\lambda_{j,t}\}_{j=1}^J$ with zero vector and terminate the optimization when reach the number of iterations T. $\mathcal{DOS}$ provides a progressive optimization towards the Pareto front and ensures that it will not leave its closure once the solution enters the Pareto front.

Theorem 2. *Assuming t_f>0, $\psi^{t_f} \in PF_{t_f}$, $\psi^0 \notin PF_{t_f}$, then $\forall t < t_f$, if $|\cos(\vec{x}_i, \overrightarrow{\Gamma(\theta^t_i)})| \cdot S(\psi^t) > 0$, optimization based on Eq. 6 will bring Pareto improvement for ψ^t. In addition, $\exists t_\epsilon > t_f$ such that $\psi^{t_\epsilon} \in PF_{t_\epsilon}$, then $\forall t \geq t_\epsilon, \psi^t \in \overline{PF_{t_\epsilon}}$, that is, the solution enters PF_{t_ϵ} will not leave closure $\overline{PF_{t_\epsilon}}$.*

Experiments

Experiment Setup

Datasets. We use multi-task learning datasets Multi-MNIST, Multi-Fashion, and Fashion-MNIST (Xiao et al. 2017), to evaluate the MOO capability of IOP, which contains 120K training and 20K testing samples. Each dataset has two objectives, i.e., classifying the top-left and bottom-right images. We use domain adaptation datasets PACS (Li et al. 2017), DomainNet (Peng et al. 2019), and Office-Home (Hemanth 2017) to evaluate the adaptability of IOP.

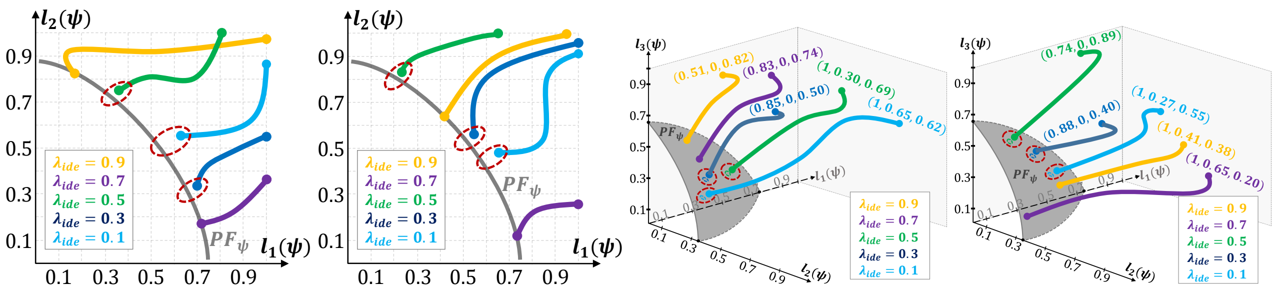
Baselines. Our baselines consist of MOO and PHN groups. The former includes MGD (Sener 2018), UW (Kendall et al. 2018), LS (Boyd et al. 2004), PMTL (Lin et al. 2019), COSMOS (Ruchte et al. 2021), DWA (Liu et al. 2019), NashMTL (Navon et al. 2022), EPO (Mahapatra 2020), FAMO (Liu et al. 2024), and PNG (Ye et al. 2022) methods, they use different techniques to improve the quality of Pareto front. The PHN group contains CPMTL (Lin et al. 2020), PHN-LS, PHN-EPO (Navon et al. 2020), and PHN-HVI (Hoang et al. 2023) methods, they all learn the entire Pareto front using a single Hypernetwork.

Implementation Details. We use a 10-layer Multi-Layer Preceptor and ResNet18 (He et al. 2016) as the backbone of Hypernetwork and target network. We use *DySample* (Liu et al. 2023) to upsample the sample, aligning the dimensions of input and output of Hypernetwork. By default, we set the training round T, learning rate η, input dimension of Hypernetwork d, target network number N, contraction coefficient λ_{ide}, and balance coefficient β to 1000, 0.001, 10^5, 100, 0.75 and 0.1. We use the cross-entropy loss as l_{tn} within L_{tn} in Eq. 1. All the experiments are implemented with *Pytorch* and trained on a single *NVIDIA Tesla V100*.

2/3-Objective Problem Optimization Trajectory

We evaluate the optimization trajectory of the proposed IOP on the synthesized 2/3-objective problem (2/3OP) (Lin et al. 2019). The 2OP contains two loss functions:

$l_1(\psi) = 1 - \exp(-\|\psi - q^{-\frac{1}{2}}\|_2^2),$



(a) 2OP (N = 100)   (b) 2OP (N = 200)   (c) 3OP (N = 100)   (d) 3OP (N = 200)

Figure 4: The loss trajectory in 2/3OP optimal solution search. The optimal solutions of 2/3OP form the Pareto front PF_ψ. The colored curve represents the search path of the solution under different λ_{ide}. We mark the solutions that do not converge accurately on PF_ψ with a red dotted ellipse.

Groups	Methods	Multi-MNIST			Multi-Fashion			Fashion-MNIST		
		Acc ↑	*HV* ↑	*Cost* ↓	*Acc* ↑	*HV* ↑	*Cost* ↓	*Acc* ↑	*HV* ↑	*Cost* ↓
MOO	MGD	89.52 ± 1.41	2.80 ± 0.08	7.25 ± 0.29	88.03 ± 0.42	2.56 ± 0.27	9.32 ± 0.13	89.81 ± 1.62	2.78 ± 0.06	8.59 ± 0.07
	UW	89.56 ± 0.31	2.81 ± 0.24	7.22 ± 0.35	88.81 ± 0.24	2.62 ± 0.12	9.28 ± 0.71	89.33 ± 0.54	2.76 ± 0.09	8.61 ± 0.59
	LS	89.78 ± 0.24	2.85 ± 0.11	7.42 ± 0.13	89.02 ± 0.21	2.63 ± 0.12	9.31 ± 0.04	90.22 ± 1.01	2.79 ± 0.13	8.92 ± 0.03
	PMTL	89.72 ± 1.72	2.84 ± 0.41	7.31 ± 0.67	88.74 ± 0.61	2.61 ± 0.07	9.26 ± 0.33	89.27 ± 1.14	2.76 ± 0.21	$\underline{8.57}$ ± 0.61
	COSMOS	90.67 ± 1.56	2.87 ± 0.23	7.81 ± 0.91	89.69 ± 0.81	2.64 ± 0.12	9.66 ± 0.03	91.52 ± 1.47	2.84 ± 0.21	9.14 ± 0.31
	DWA	90.21 ± 1.23	2.86 ± 0.35	$\underline{7.09}$ ± 0.21	90.01 ± 0.41	2.65 ± 0.41	**9.03** ± 0.31	89.29 ± 1.71	2.76 ± 0.14	8.60 ± 0.24
	NashMTL	90.78 ± 1.24	2.88 ± 0.14	7.18 ± 0.74	90.57 ± 0.52	2.67 ± 0.21	9.39 ± 0.72	89.79 ± 1.37	2.78 ± 0.23	8.58 ± 0.15
	EPO	91.72 ± 0.62	2.90 ± 0.34	**6.73** ± 0.22	89.71 ± 1.81	2.64 ± 0.16	$\underline{9.18}$ ± 0.11	91.81 ± 2.02	2.85 ± 0.12	**8.51** ± 0.21
	FAMO	92.42 ± 1.09	2.91 ± 0.22	7.21 ± 0.13	91.24 ± 0.32	2.69 ± 0.14	9.24 ± 0.32	90.47 ± 1.34	2.80 ± 0.15	8.72 ± 0.35
	PNG	92.74 ± 1.23	2.92 ± 0.31	7.44 ± 0.23	90.37 ± 1.89	2.66 ± 0.25	9.48 ± 0.71	$\underline{92.73}$ ± 1.85	$\underline{2.88}$ ± 0.21	9.04 ± 1.03
PHN	CPMTL	90.45 ± 1.01	2.86 ± 0.21	7.68 ± 0.67	90.04 ± 0.71	2.65 ± 0.02	9.43 ± 0.23	90.57 ± 0.89	2.81 ± 0.11	9.07 ± 0.32
	PHN-LS	90.93 ± 2.84	2.88 ± 0.42	7.72 ± 0.71	90.61 ± 0.71	2.68 ± 0.02	9.46 ± 0.42	90.62 ± 2.78	2.82 ± 0.14	8.98 ± 0.03
	PHN-EPO	91.34 ± 1.43	2.89 ± 0.31	7.61 ± 0.63	91.87 ± 1.42	2.70 ± 0.11	9.53 ± 0.36	91.63 ± 0.78	2.84 ± 0.02	8.97 ± 0.01
	PHN-HVI	$\underline{92.81}$ ± 0.54	$\underline{2.93}$ ± 0.13	7.78 ± 0.23	$\underline{93.01}$ ± 1.03	$\underline{2.71}$ ± 0.09	9.69 ± 0.13	92.41 ± 0.61	2.87 ± 0.08	9.12 ± 0.16
	IOP (ours)	**95.34** ± 0.81	**2.99** ± 0.13	7.82 ± 0.37	**94.78** ± 1.63	**2.78** ± 0.08	9.71 ± 0.18	**94.98** ± 0.73	**2.93** ± 0.11	9.23 ± 0.08

Table 1: Comparison of accuracy, *HV*, and training cost (GFlops, G=10^9). *Acc* denotes the average classification accuracy for the top-left and bottom-right images in the datasets. The best results are highlighted in bold and the second best are underlined.

$$l_2(\psi) = 1 - \exp(-\|\psi + q^{-\frac{1}{2}}\|_2^2),$$

where $q = 10^5$. The 3OP contains three loss functions:

$$l_1(\psi) = \cos(\tfrac{\pi\psi_1}{2})\cos(\tfrac{\pi\psi_2}{2})(\textstyle\sum_{i=3}^{q}(\psi_i - \frac{1}{2})^2 + 1),$$
$$l_2(\psi) = \cos(\tfrac{\pi\psi_1}{2})\sin(\tfrac{\pi\psi_2}{2})(\textstyle\sum_{i=3}^{q}(\psi_i - \frac{1}{2})^2 + 1),$$
$$l_3(\psi) = \sin(\tfrac{\pi\psi_1}{2})(\textstyle\sum_{i=3}^{q}(\psi_i - \frac{1}{2})^2 + 1),$$

where $q = 10^6, \psi = [\psi_1, ..., \psi_q], 0 \leq \psi_i \leq 1, i \in [q]$. Unlike (Lin et al. 2019), we use a larger dimension of q $(10^5, 10^6)$ to simulate the weight number of Hypernetwork, which is more realistic in our scenario. Fig. 4 demonstrates the loss trajectories in the search process for optimal solutions, which start from random weights. Overall, under different target network numbers N and contraction coefficients λ_{ide}, the search paths of solutions can accurately converge to the Pareto front, indicating that our IOP has the ability to optimize towards the Pareto front. Notably, a smaller λ_{ide}, such as 0.1 or 0.3, may lead to the search trajectory's convergence point not being wholly located on the Pareto front, as indicated by red ellipses in Fig. 4, which indicates that insufficient manifold contraction of target distribution $\mathcal{P}_\theta$ in Eq. 4 can produce unstable solutions. However, it is not wise to fully tighten $\mathcal{P}_\theta$ by setting λ_{ide} = 1, as this may reduce the adaptability of Hypernetwork. We provide guidance on empirically setting λ_{ide} in the subsequent section.

Comparison of Accuracy, *HV*, and Training Cost

We evaluate the quality of solutions obtained by different methods on datasets Multi-MNIST, Multi-Fashion, and Fashion-MNIST using accuracy and Hypervolume (*HV*) metrics. We run all methods with five independent trials and report the average value and standard deviation in Table 1. When calculating the *HV* metric, we specify the reference point (2, 2) empirically for all methods. We ensure that the different loss values of each method are not greater than the corresponding coordinate value of the reference point, which is a common practice in MOO (Ye et al. 2022; Hoang et al. 2023). On all datasets, IOP outperforms the methods in MOO and PHN groups by an average of 5.3% and 4.0% on the accuracy, and 5.2% and 3.9% on the *HV*. The approximate positive correlation between accuracy and *HV* indicates the effectiveness of *HV* metric in reflecting performance. In addition, we use the *Torchstat* (Swall0w 2018) tool to estimate the computational cost of different methods in training. The result shows that IOP has an average increase of 3.7% and 2.1% costs compared to MOO and PHN groups. The difference in the increased costs is because IOP not only includes the cost generated by the Hypernetwork, as the methods in the PHN group, but also contains the additional cost brought by the idempotent-like optimization.

Impact of λ_{ide} on the Hypernetwork's Adaptability

Fig. 5 evaluates the impact of contraction coefficient λ_{ide} on Hypernetwork's adaptability. We statistic the average accuracy of target networks generated by Hypernetwork on datasets PACS, DomainNet, and Office-Home. The accuracy varies similarly on different datasets and is unaffected by the amount of target network N. Specifically, as λ_{ide} ranges from 0 to 0.75, the accuracy increases and peaks at approximately 0.75, decreasing as λ_{ide} from 0.75 to 1. The reason for this is straightforward. We have discussed that using a larger λ_{ide} to tighten the target distribution $\mathcal{P}_\theta$ is beneficial for improving the accuracy of solutions. However, from Eq. 4, we know that a larger λ_{ide} limits the solution range and reduces its diversity, damaging Hypernetwork's adaptability.

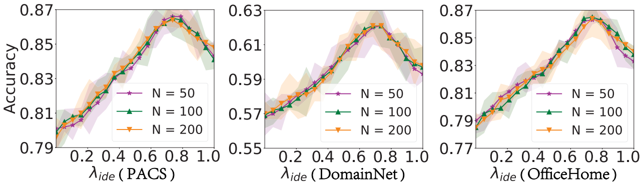


Figure 5: Impact of λ_{ide} on the accuracy of target network.

We use an independent cross-entropy loss function for each domain in the above datasets and construct the MOO problem on them. Table 2 provides the cross-domain average accuracy (CDAA) of different methods. In the MOO group, we use Resnet18 (He et al. 2016) as the backbone of the model in baselines and optimize it using the technique described in the literature. Our IOP outperforms the methods in MOO and PHN groups by an average of 12.3% and 8.7% on CDAA. Although a too-small or too-large λ_{ide} (e.g., 0.35 or 0.95) may weaken this advantage, it does not change the fact that IOP can effectively compensate for the shortcomings of MOO and PHN methods in model adaptability.

Groups	Methods	PACS	DomainNet	Office-Home
MOO	MGD	76.23 ± 1.53	56.12 ± 1.24	77.23 ± 1.73
	UW	74.72 ± 0.45	54.62 ± 1.47	73.62 ± 0.52
	LS	69.14 ± 1.72	51.82 ± 1.24	68.31 ± 1.61
	PMTL	78.43 ± 0.52	58.13 ± 1.43	78.56 ± 0.32
	COSMOS	79.12 ± 1.32	58.72 ± 0.13	78.69 ± 1.31
	DWA	75.03 ± 1.52	55.11 ± 0.87	74.72 ± 1.32
	NashMTL	71.72 ± 1.96	53.85 ± 1.72	72.92 ± 0.89
	EPO	79.04 ± 1.62	57.35 ± 0.35	77.42 ± 0.23
	FAMO	77.04 ± 1.32	56.23 ± 1.09	76.46 ± 1.93
	PNG	79.42 ± 0.72	59.02 ± 1.14	79.42 ± 1.74
PHN	CPMTL	71.32 ± 1.07	55.14 ± 0.59	72.82 ± 1.35
	PHN-LS	79.06 ± 2.11	59.01 ± 0.21	78.13 ± 1.21
	PHN-EPO	80.22 ± 0.13	60.09 ± 0.31	81.56 ± 1.03
	PHN-HVI	80.12 ± 1.46	60.03 ± 0.23	80.43 ± 1.32
	IOP (λ_{ide}=0.35)	83.23 ± 1.31	59.32 ± 1.42	82.08 ± 0.68
	IOP (λ_{ide}=0.75)	**86.25** ± 0.85	**62.07** ± 1.32	**85.46** ± 0.36
	IOP (λ_{ide}=0.95)	84.47 ± 0.63	60.13 ± 0.63	84.86 ± 1.03

Table 2: Cross-domain average accuracy on adaptation datasets PACS, DomainNet, and Office-Home.

Ablation Study

We conduct an ablation experiment to evaluate the contribution of Cosine and Hypervolume terms in MOO loss (Eq. 1), as well as idempotent-like loss (Eq. 4). As shown in Table 3, the results on the multi-task learning datasets indicate a noticeable performance degradation using a single loss term, which proves that the roles of terms Consine and Hypervolume are indispensable. The results on the adaptation datasets show that the idempotent-like loss significantly improves the adaptability of Hypernetwork. Note that adopting idempotent-like loss alone is pointless, as the other two are indispensable constraints.

Loss Type	Multi-MNIST	Multi-Fashion	Fashion-MNIST
Cos	83.12 ± 0.4	82.71 ± 0.3	80.21 ± 1.1
Hyp	85.35 ± 0.2	83.01 ± 0.2	81.14 ± 0.5
Cos + Hyp	95.17 ± 0.2	95.69 ± 0.1	94.87 ± 1.3
Cos + Ide	82.51 ± 1.3	81.73 ± 1.4	79.85 ± 1.1
Hyp + Ide	86.01 ± 0.5	82.89 ± 0.6	80.84 ± 1.4
Cos + Hyp + Ide	**95.19** ± 1.5	**95.72** ± 1.8	**94.93** ± 1.2
Loss Type	**PACS**	**DomainNet**	**Office-Home**
Cos	77.89 ± 1.3	56.93 ± 0.5	76.61 ± 0.8
Hyp	77.21 ± 0.4	56.01 ± 1.1	75.32 ± 0.3
Cos + Hyp	79.78 ± 1.6	58.61 ± 0.4	78.53 ± 0.2
Cos + Ide	81.52 ± 1.1	59.13 ± 0.3	80.17 ± 0.5
Hyp + Ide	83.41 ± 1.6	60.42 ± 1.6	82.07 ± 0.4
Cos + Hyp + Ide	**86.18** ± 1.2	**62.17** ± 0.4	**85.58** ± 0.8

Table 3: Ablation results of loss types. Cos: cosine loss term, Hyp: Hypervolume loss term, Ide: idempotent-like loss.

We also evaluate the effectiveness of IOP under different objective function numbers (OFN) and dimensions of Hypernetwork's input/output (DHIO) on classic MOO problems 2OP (Lin et al. 2019), Bi-CVRP (Zajac et al. 2021), Bi-KP (Ishibuchi et al. 2014), 3OP (Lin et al. 2019), Tri-TSP (Lust et al. 2010), and Tri-CVRP (Zajac et al. 2021). Table 4 shows that the varying DHIO won't impact the optimization – there is negligible performance difference on metrics *HV* and OS in different problems. The training cost, however, will increase with the growth of OFN or DHIO, stemming from upsampling the samples and calculating the similarity between samples and the loss vector (Eq. 1).

Problems	DHIO = 10^5			DHIO = 10^7		
	HV	OS	Cost	*HV*	OS	Cost
OFN = 2						
2OP	0.26 ± 0.01	134 ± 3	4.17 ± 0.2	0.25 ± 0.01	133 ± 3	4.58 ± 0.3
Bi-CVRP	0.42 ± 0.02	251 ± 5	4.42 ± 0.1	0.41 ± 0.01	249 ± 4	4.84 ± 0.3
Bi-KP	0.44 ± 0.02	202 ± 5	4.63 ± 0.5	0.43 ± 0.02	199 ± 4	5.02 ± 0.2
OFN = 3						
3OP	0.50 ± 0.01	532 ± 7	6.12 ± 0.4	0.49 ± 0.02	534 ± 4	6.74 ± 0.1
Tri-TSP	0.57 ± 0.03	968 ± 5	6.53 ± 0.2	0.56 ± 0.04	970 ± 3	7.19 ± 0.3
Tri-CVRP	0.62 ± 0.04	773 ± 5	6.93 ± 0.5	0.61 ± 0.03	777 ± 5	7.63 ± 0.3

Table 4: Experimental results on Hypervolume (*HV*), number of optimal solutions (OS), and training cost (GFlops, G=10^9) under different objective function numbers (OFN) and dimensions of Hypernetwork's input/output (DHIO).

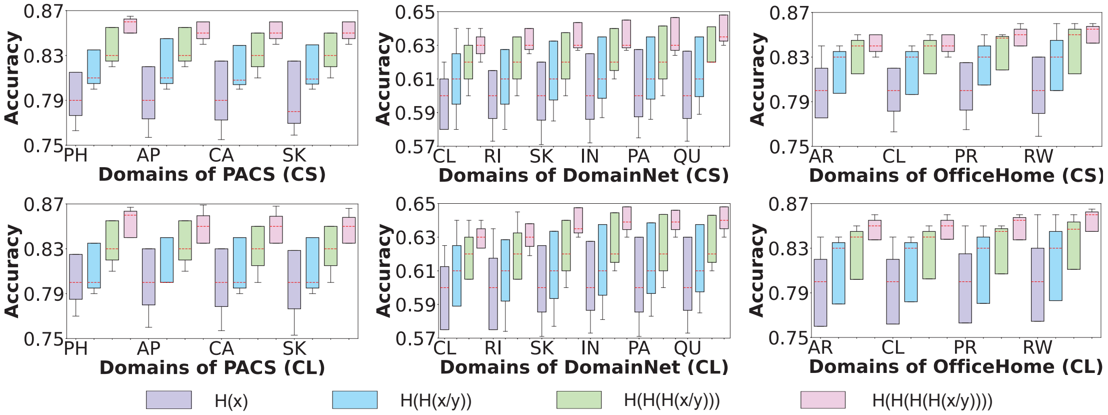


Figure 6: The accuracy distribution of the four levels of target networks. The subfigures in the top line use the CS input, the bottom line uses the CL input. The letters in the horizontal axis denote the domains in the dataset.

Iterative Input for Improving the Performance

Our idempotent-like optimization endows the Hypernetwork with the characteristic of receiving iterative inputs. We evaluate its contribution to improving Hypernetwork performance in Fig. 6. Specifically, we randomly select 100 samples from each domain of the dataset and apply our Hypernetwork multiple times to each sample (up to four), generating four levels of target networks for each sample. We use two input methods: (1) Consistent Sample (CS): Applying the Hypernetwork consecutively on the same sample x, i.e., $H(x;\psi)$, $H(H(x;\psi);\psi)$, $H(H(H(x;\psi);\psi);\psi)$, and $H(H(H(H(x;\psi);\psi);\psi);\psi)$ denote the four levels of target networks. (2) Consistent Label (CL): For each sample x, we randomly sample y with the same label as x from the same domain and apply the Hypernetwork consecutively on y, i.e., $H(x;\psi)$, $H(H(y;\psi);\psi)$, $H(H(H(y;\psi);\psi);\psi)$, and $H(H(H(H(y;\psi);\psi);\psi);\psi)$ denote the four levels of target networks. Fig. 6 shows the accuracy distribution of these target networks. When using the CS input method, not all first-level target networks (apply Hypernetwork once) achieve optimal performance in each domain. Applying the Hypernetwork two or three times can improve the accuracy of target networks. Almost all the target networks achieve upward performance alignment (UPA) when applying the Hypernetwork four times, showcasing cohesion and obtaining a smaller interquartile range. Although there are more non-optimal second-level or third-level target networks for the CL input method than CS, similar UPA can still be achieved after consecutively applying the Hypernetwork to samples with the same label. Experimental results indicate that IOP pioneers a new way to improve Hypernetwork performance by steering the model through iterative inputs for more efficient parameter updates.

General Applicability of IOP

We apply IOP to the optimization of multimodal model Vision-LLMv2 (VLLMv2) (Wu et al. 2024), which includes dialogue, image, and object recognition data simultaneously. We consider VLLMv2 as the target network and connected to our Hypernetwork. Optimizing different types of data in VLLMv2 inevitably leads to conflicts. Therefore, we take the loss function of different types of data used in VLLMv2 as the balancing objective and use the idempotent-like objective to improve VLLMv2's adaptability in downstream tasks. The results in Table 5 indicate that IOP improve the performance of dialogue, recognition, reasoning, and generalization tasks by 2.4%, 6.2%, 3.7%, and 7.5%, separately, which proves the effectiveness of IOP.

Methods	VQAv2	GQA	POPE	SEED	COCO
VLLMv2	81.2	65.2	84.5	63.3/70.3/41.9	77.2/82.8
VLLMv2+IOP	**82.5**	**66.8**	**86.2**	**65.5/71.2/42.7**	**81.3/88.6**

Methods	VCR (Q-A/QA-R/Q-AR)	OdinW13 (*Rabbit*/*Raccoon*/*Vehicle*/*Avg*)
VLLMv2	87.2 / 88.1 / 79.1	72.1 / 56.1 / 60.3 / 62.8
VLLMv2+IOP	**90.4 / 91.2 / 82.1**	**77.3 / 61.6 / 63.5 / 67.5**

Table 5: Comparison on dialogue datasets (VQAv2 (Goyal et al. 2017), GQA (Hudson 2019), POPE (Li et al. 2023), SEED (all/image/video) (Ge et al. 2024)), region recognition dataset COCO (mAP/Acc) (Lin et al. 2014), visual commonsense reasoning dataset VCR (Zellers et al. 2019), and object detection generalization dataset OdinW13 (Li et al. 2022). In VCR, Q, A, and R denote question, answer, and rationale, Q-A means that the model needs to select option A conditioned on Q.

Conclusion

This paper reveals the main limitations of the most existing PHN methods and proposes a novel idempotent-like optimization method on the Pareto front of Hypernetwork, namely IOP. Our theoretical analysis shows that IOP ensures the convergence of the optimization process. Experimental results demonstrate that IOP effectively improves the performance and adaptability of Hypernetwork with low computational cost. In addition, IOP pioneers a new way to improve the performance of Hypernetwork through iterative input.

Acknowledgments

We would very much like to thank the anonymous reviewers for their valuable and constructive comments. This work is supported, in part by Beijing Natural Science Foundation (No. L241050), in part by the National Natural Science Foundation of China (No. 62402024), in part by the Fundamental Research Funds for the Central Universities. For any correspondence, please refer to Renyu Yang (renyuyang@buaa.edu.cn) and Xudong Liu (liuxd@buaa.edu.cn).

References

Boyd, S.; et al. 2004. *Convex optimization*. Cambridge university press.

Chen, J.; Wang, J.; Zhang, Z.; Cao, Z.; Ye, T.; and Chen, S. 2024. Efficient meta neural heuristic for multi-objective combinatorial optimization. *Advances in Neural Information Processing Systems*, 36.

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

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

Ehrgott, M. 2005. *Multicriteria optimization*, volume 491. Springer Science & Business Media.

Gao, Y.; Lu, J.; Li, S.; Li, Y.; and Du, S. 2024. Hypergraph-Based Multi-View Action Recognition Using Event Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ge, Y.; Zhao, S.; Zhu, J.; Ge, Y.; Yi, K.; Song, L.; Li, C.; Ding, X.; and Shan, Y. 2024. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6325–6334.

Ha, D.; et al. 2016. HyperNetworks. In *International Conference on Learning Representations*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hemanth, V. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5018–5027.

Hoang, L. P.; Le, D. D.; Tuan, T. A.; and Thang, T. N. 2023. Improving pareto front learning via multi-sample hypernetworks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7875–7883.

Hudson, D. A. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709.

Ishibuchi, H.; et al. 2014. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, 19(2): 264–283.

Kendall, A.; et al. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.

Li, L. H.; Zhang, P.; Zhang, H.; Yang, J.; Li, C.; Zhong, Y.; Wang, L.; Yuan, L.; Zhang, L.; Hwang, J.-N.; et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10965–10975.

Li, Y.; Du, Y.; Zhou, K.; Wang, J.; Zhao, W. X.; and Wen, J.-R. 2023. Evaluating Object Hallucination in Large Vision-Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 292–305.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Lin, X.; Yang, Z.; Zhang, Q.; and Kwong, S. 2020. Controllable Pareto Multi-Task Learning.

Lin, X.; Zhen, H.-L.; Li, Z.; Zhang, Q.-F.; and Kwong, S. 2019. Pareto multi-task learning. *Advances in neural information processing systems*, 32.

Liu, B.; et al. 2024. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36.

Liu, S.; et al. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1871–1880.

Liu, W.; Lu, H.; Fu, H.; and Cao, Z. 2023. Learning to Upsample by Learning to Sample. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6027–6037.

Lust, T.; et al. 2010. The multiobjective traveling salesman problem: A survey and a new approach. In *Advances in Multi-Objective Nature Inspired Computing*, 119–141. Springer.

Ma, P.; et al. 2020. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learning*, 6522–6531. PMLR.

Mahapatra, D. 2020. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, 6597–6607. PMLR.

Navon, A.; Shamsian, A.; Fetaya, E.; and Chechik, G. 2020. Learning the Pareto Front with Hypernetworks. In *International Conference on Learning Representations*.

Navon, A.; et al. 2022. Multi-Task Learning as a Bargaining Game. In *International Conference on Machine Learning*, 16428–16446. PMLR.

Nguyen, H.; Rezatofighi, H.; Vo, B.-N.; and Ranasinghe, D. C. 2020. Multi-objective multi-agent planning for jointly discovering and tracking mobile objects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7227–7235.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1406–1415.

Ruchte, M.; et al. 2021. Scalable pareto front approximation for deep multi-objective learning. In *2021 IEEE international conference on data mining (ICDM)*, 1306–1311. IEEE.

Sener, O. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

Shocher, A.; Dravid, A.; Gandelsman, Y.; Mosseri, I.; Rubinstein, M.; and Efros, A. A. 2023. Idempotent generative network. *arXiv preprint arXiv:2311.01462*.

Swall0w. 2018. Torchstat. https://github.com/Swall0w/torchstat.

Wu, J.; Zhong, M.; Xing, S.; Lai, Z.; Liu, Z.; Wang, W.; Chen, Z.; Zhu, X.; Lu, L.; Lu, T.; Luo, P.; Qiao, Y.; and Dai, J. 2024. VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks. *ArXiv*, abs/2406.08394.

Xiang, J.; Li, H.; Lian, D.; Huang, G.; Watanabe, T.; and Liu, L. 2022. Visualizing the Relationship Between Encoded Linguistic Information and Task Performance. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Findings of the Association for Computational Linguistics: ACL 2022*, 410–422. Dublin, Ireland: Association for Computational Linguistics.

Xiao, H.; et al. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv e-prints*, arXiv–1708.

Ye, M.; et al. 2022. Pareto navigation gradient descent: a first-order algorithm for optimization in pareto set. In *Uncertainty in Artificial Intelligence*, 2246–2255. PMLR.

Zajac, S.; et al. 2021. Objectives and methods in multi-objective routing problems: a survey and classification scheme. *European Journal of Operational Research*, 290(1): 1–25.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6720–6731.

Zitzler, E. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4): 257–271.