Impartial Multi-task Representation Learning via Variance-invariant Probabilistic Decoding

Anonymous ACL submission

Abstract

Multi-task learning (MTL) enhances efficiency by sharing representations across tasks, but task dissimilarities often cause partial learning, where some tasks dominate while others are neglected. Existing methods mainly focus on balancing loss or gradients but fail to fundamentally address this issue. In this paper, we propose variance-invariant probabilistic decoding for multi-task learning (VIP-MTL), a framework that ensures impartial learning by harmonizing task-specific representation spaces. VIP-MTL decodes task-agnostic shared representations into task-specific probabilistic distributions and applies variance normalization to constrain them to a consistent scale, balancing task influence during training. Experiments on two language benchmarks show that VIP-MTL outperforms 12 comparative methods under the same multi-task settings, especially in heterogeneous and data-constrained scenarios. Further analysis shows that VIP-MTL is robust to sampling distributions, efficient on optimization process, and scale-invariant to task losses. Additionally, the learned task-specific representations are more informative, enhancing the language understanding abilities of pre-trained language models under the multi-task paradigm.

1 Introduction

Multi-task learning (MTL) has emerged as a powerful paradigm in machine learning, enabling models to jointly learn multiple tasks together from the shared representations (Caruana, 1997; Kendall et al., 2018). Unlike single-task learning, MTL paradigm not only allows the learned representations to simultaneously make predictions for several tasks, but also reduces computation costs and improves efficiency (Royer et al., 2023).

However, a persistent challenge in MTL stems from the inherent task dissimilarity, which often leads to the partial learning problem (Liu et al., 2021b). This occurs when the model disproportionately prioritizes certain tasks while neglecting others, resulting in suboptimal overall performance. In multi-task learning, the latent variable distributions of different tasks are often inconsistent. For example, the latent variable distribution of Task A may have a larger variance, while the latent variable distribution of Task B may have a smaller variance. This discrepancy can cause the representations of Task A to dominate the optimization process, while the representations of Task B is neglected.

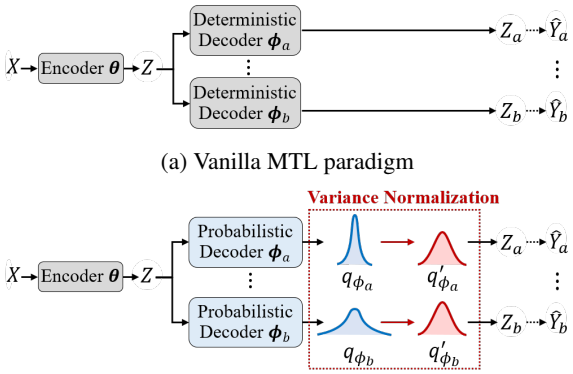
Existing methods (Kendall et al., 2018; Chennupati et al., 2019; Liu et al., 2019a; Yu et al., 2020; Liu et al., 2021b; Lin et al., 2022) primarily focus on balancing task losses or gradients but fail to address the fundamental misalignment in representations. Balancing losses adjusts task weights heuristically, yet it cannot resolve scale disparities in latent spaces. Similarly, gradient balancing harmonizes parameter updates during backpropagation. However, gradients are inherently influenced by the statistical properties of representations (e.g., magnitude, variance). If representations are imbalanced, gradients will inevitably reflect this bias. Specifically, high-variance tasks generate larger gradients, perpetuating their dominance despite gradient normalization efforts. These limitations are particularly pronounced in scenarios involving heterogeneous tasks or limited data, where the disparities in task complexity and data availability exacerbate the imbalance. Therefore, balancing representations offers a more principled and effective solution to the partiality problem in MTL.

In this paper, we introduce a novel framework, variance-invariant probabilistic decoding for multi-task learning (VIP-MTL), which tackles the partial learning problem by harmonizing the representation spaces across tasks. Specifically, our framework decodes task-agnostic shared representations into task-specific probabilistic distributions, where each point in the distribution corresponds to a potential task-specific representation. Unlike prior methods that focus on loss or gradient balancing,

VIP-MTL operates at the level of representation balancing, ensuring impartial learning on representation spaces for all tasks. To address the issue of scale variance across tasks, we apply a variance normalization to these distributions, adaptively constraining them to a consistent scale. By aligning the representation distributions, VIP-MTL prevents any single task from dominating the shared representation space and ensures that the influence of each task remains balanced during training.

We conduct experiments on two multi-task benchmarks, TweetEval and AffectEval for language understanding. The former includes 6 classification tasks, while the latter involves 2 classification tasks and 2 regression tasks in a heterogeneous multi-task setting. The results show that our VIP-MTL consistently surpasses 12 representative multi-task methods across different PLMs under the same multi-task settings. For example, with the RoBERTa backbone, VIP-MTL improves the average relative improvement (Δp) by **+5.06%** on TweetEval and **+7.66%** on AffectEval, compared to the EW baseline. Compared to single task learning baselines, VIP-MTL also achieves better results on most tasks with the same scale of model parameters. Further analysis shows that our method is robust to sampling distributions, efficient on optimization process, and scale-invariant to task losses. Extensive experiments demonstrate that VIP-MTL offers significant advantages in heterogeneous task combinations and data-constrained scenarios. Additionally, the learned task-specific representations are more informative, enhancing the language understanding abilities of pre-trained language models under the multi-task paradigm.

The contributions are as follows: 1) We analyze the limitations of existing methods that balance losses or gradients in addressing the partial learning problem in MTL from the perspective of representation distributions, and introduces a new idea of balancing the representation spaces across tasks to promote impartial learning. 2) We design a probabilistic representation learning framework VIP-MTL to tackle the partial learning problem by harmonizing the representation spaces across tasks. It decodes shared representations into task-specific probabilistic distributions and applies variance normalization to constrain them. 3) Experiments on two language understanding benchmarks show that our method outperforms 12 comparative methods under the same multi-task settings, especially in heterogeneous and data-constrained scenarios. Further analysis shows that VIP-MTL is distribution-robust, efficient, scale-invariant, and the learned task-specific representations are more informative for all tasks.[1]



(a) Vanilla MTL paradigm

(b) VIP-MTL (ours)

Figure 1: Comparison of vanilla MTL paradigm and the proposed VIP-MTL. The deterministic decoder maps each vector point to a fixed vector, while the probabilistic decoder that maps each vector point to a probability distribution.

2 Preliminary

Scope of the Study. The goal of this paper is to study multi-task optimization that typically utilizes a hard parameter-sharing setting (Caruana, 1993), where several lightweight task-specific heads are attached to a heavyweight task-agnostic backbone model. Another orthogonal line of research on multi-task learning mainly emphasizes designing of network architectures that typically use a soft parameter-sharing strategy. Details of the above related studies are listed in Appendix A.

Notations. Define T tasks and the corresponding dataset of task t as $\mathcal{D}_t$. An MTL model typically comprises task-sharing encoder with parameters θ and task-specific decoder with parameters $\{\phi_t\}_{t=1}^{|T|}$, where θ represents parameters in a feature extractor shared by all tasks, and ϕ_t represents parameters in the task-specific output module for task t. Define $\ell_t(\mathcal{D}_t; \theta, \phi_t)$ as the average loss on the dataset $\mathcal{D}_t$ for task t. $\{\lambda_t\}_{t=1}^{|T|}$ is the set of task-specific loss weights with a constraint, where $\lambda_t \geq 0$.

MTL Baseline. The total MTL objective is computed by aggregating multiple objective losses with different weights, i.e., $\mathcal{L}(\theta, \{\phi_t\}_{t=1}^{|T|}) = \sum_{t=1}^{|T|} \lambda_t^l \ell_t(\mathcal{D}_t; \theta, \phi_t)$. A straightforward method

[1]The source code will be available in the future.

involves assigning equal weights to all tasks during training, i.e., $\lambda_t = \frac{1}{|T|}$ for all tasks in every iteration, i.e., a common MTL baseline EW.

3 Methodology

We propose variance-invariant probabilistic decoding for multi-task learning (VIP-MTL), a probabilistic framework that ensures impartial learning. As shown in Figure 1b, the encoder learns task-agnostic shared representations across all tasks. Based on shared representations, VIP-MTL decodes shared representations into task-specific probabilistic distributions and applies variance normalization to constrain them to a consistent scale, balancing task influence during training. Different from the vanilla MTL paradigm (Figure 1a) that jointly learn multiple tasks by balance losses or gradients, VIP-MTL balances representation spaces across tasks to promote impartial learning.

3.1 Probabilistic Decoding for MTL

To decode task-agnostic point-wise shared representations into task-specific probabilistic distributions, we perform probabilistic embedding (Vilnis and McCallum, 2015; Hu et al., 2024a) and task prediction in the multi-task decoding process.

We extend the probabilistic coding technique (Hu et al., 2024a) in single-task learning to the multi-task setting. Specifically, we use variational inference (Hoffman et al., 2013) to map the shared representations z to a set of different distributions in the output space, i.e., $\mathbb{R}^{|\mathcal{Y}_t|}$. Given the input x, the task-agnostic shared representation z shared by all tasks is a function of x by a mapping $p_\theta(z|x)$. For task t, the output representations z_t in the output space can be obtained by a task-specific head $q_{\phi_t}(z_t|z)$, and the corresponding prediction value $\hat{y}_t$ is non-parametric mapping of z_t. The true posterior $p(z_t|x)$ can be approximated as $p(z_t|z)$ where $z \sim p(z|x)$. Let the prior estimate $r(z_t)$ for task t be the isotropic Gaussian distribution, i.e., $r(z_t) \sim \mathcal{N}(z_t; \mathbf{0}, \mathbf{I})$. Let $q_{\phi_t}(z_t|z)$ be a variational estimate of the intractable true posterior $p(z_t|z)$ of z_t given z, and learned by the t-th stochastic head parametrized by ϕ_t. And the objective of probabilistic decoding for MTL can be:

$$\mathcal{L}(\theta, \{\phi_t\}_{t=1}^T) = \mathbb{E}_{t\sim T, z\sim p_\theta(z|x)}\{\mathbb{E}_{z_t\sim q_{\phi_t}(z_t|z)}[-\log s(y_t|z_t)] + \beta KL(q_{\phi_t}(z_t|z); r(z_t))\}, \quad (1)$$

where z_t is randomly sampled from $p_{\phi_t}(z_t|z)$. $s(y_t|z_t)$ is a non-parametric operation on z_t that adapts the output distribution for task prediction (e.g., the Softmax operation for classification). $KL(\cdot)$ denotes the KL-divergence term, which serves as a regularization that forces the variational posterior $q_{\phi_t}(z_t|z)$ to approximately converge to the prior $r(z_t)$. $\beta > 0$ controls the closeness between the learnable variational posterior $q_{\phi_t}(z_t|z)$ and the predefined prior $r(z_t)$. The different values of β means the posterior distribution with different parametric forms.

For task t, we assume the variational posterior $q_{\phi_t}(z_t|z)$ be a multivariate Gaussian with a diagonal covariance structure, i.e.,

$$q_{\phi_t}(z_t^i|z^i) = \mathcal{N}(z_t^i; \mu_t(z^i), \Sigma_t(z^i)), \quad (2)$$

where $\mu_t(z^i)$ and $\Sigma_t(z^i)$ denote the mean and diagonal covariance of sample z^i for task t. Following Hu et al. (2024a), both of their parameters are input-dependent and can be learned by an MLP (a fully-connected neural network with a single hidden layer) for each task, respectively. Next, we sample z_t^i from the approximate posterior $q_{\phi_t}(z_t^i|z^i)$, and obtain the prediction value by $s(y_t^i|z_t^i)$. Since the sampling process of z_t^i is stochastic, we use the re-parameterization trick (Kingma and Welling, 2014) to ensure it trainable, i.e., $z_t^i = \boldsymbol{\mu}_t(z^i) + (\boldsymbol{\Sigma}_t(z^i))^{1/2} \odot \epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\odot$ refers to an element-wise product. Then, the KL term can be calculated by: $KL(q_{\phi_t}(z_t^i|z^i); r(z_t^i)) = -\frac{1}{2}\left(1 + \log \boldsymbol{\Sigma}_t(z^i) - (\boldsymbol{\mu}_t(z^i))^2 - \boldsymbol{\Sigma}_t(z^i)\right)$.

3.2 Variance Normalization on Probabilistic Distributions

By aligning the representation distributions, VIP-MTL prevents any single task from dominating the shared representation space and ensures that the influence of each task remains balanced during training. The technique adjusts the variance of population distribution of each task to a notionally common scale. It can keep the population variance constant between different tasks, and balance the learning processes for different tasks.

For task t, each sample z_t^i follows a multivariate Gaussian as shown in Eq.(2). Then all data points in the output space can be viewed as generated by a mixture of Gaussian distributions, i.e.,

$$q'_{\phi_t}(z_t|z) = \sum_{i=1}^{|\mathcal{D}_t|} \varepsilon_i q_{\phi_t}^i(z_t^i|z^i), \quad (3)$$

where $\varepsilon_1 + \cdots + \varepsilon_{|\mathcal{D}_t|} = 1$, $\varepsilon_i \geq 0$. $|\mathcal{D}_t|$ is the dataset size of task t. $q_{\phi_t}^i$ is independent

of each other. z_t follows a mixture normal distribution consisting of $|\mathcal{D}_t|$ normally distributed components. Besides, let all samples be equally weighted, i.e., $\varepsilon = \frac{1}{|\mathcal{D}_t|}$. Define a sufficiently large number ξ. When $|\mathcal{D}_t| > \xi$, the covariance of this mixture distribution can be approximated as: $\Sigma'_t \approx \frac{1}{|\mathcal{D}_t|}\left(\Sigma_t^1 + \Sigma_t^2 + \cdots + \Sigma_t^{|\mathcal{D}_t|}\right) \leq \max_i\{\Sigma_t^i\}$. Then we use Σ'_t to normalize the diagonal covariance in Eq.(2), i.e.,

$$q'_{\phi_t^*}(z_t^i|z^i) = \mathcal{N}\left(z_t^i; \frac{\mu_t(z^i)}{(\Sigma'_t)^{1/2}}, \frac{\Sigma_t(z^i)}{\Sigma'_t}\right), \text{ where } ||\Sigma'_t|| \leq \delta, \quad (4)$$

where Σ'_t is learned by a linear mapping of task t with parameters τ_t. $\phi_t^* = \{\phi_t, \tau_t\}$. δ is a certain radius for Σ'_t due to the maximum value, $\max_i\{\Sigma_t^i\}$, being constrained by the KL-divergence term in Eq.(1). And the diagonal covariance of q'_{ϕ_t} can be:

$$\Sigma'_{t,norm} \approx \frac{1}{|\mathcal{D}_t|}\left(\frac{\Sigma_t^1}{\Sigma'_t} + \frac{\Sigma_t^2}{\Sigma'_t} + \cdots + \frac{\Sigma_t^{|\mathcal{D}_t|}}{\Sigma'_t}\right) \approx \mathbf{I}. \quad (5)$$

For all jointly trained tasks, after variance normalization, they will consistently follow a mix of Gaussian distributions with approximately unit covariance in the output space. This means that the mixed distributions for all tasks have the property of approximate variance invariance: all mixed distributions in the target space have a globally consistent shape and level of dispersion. Additionally, the expectations under different tasks are scaled to similar magnitudes. While methods UW and IMTL-L also impose constraints on the expectations of different tasks, they do not constrain the variance of the distributions as our method does.

In implementations, we apply a normalization constraint to its stochastic sampled values, i.e., $(\mathbf{z}_t^i)' = \boldsymbol{\mu}_t(z^i)/(\Sigma'_t)^{1/2} + (\boldsymbol{\Sigma}_t(z_t^i)/\boldsymbol{\Sigma}'_t)^{1/2} \odot \epsilon$. To simplify the computation of Σ'_t, we assume the normalization constraint imposed on all dimension of the diagonal covariance have the same scale for task t. We take cross-entropy (CE) and mean squared error (MSE) for classification and regression tasks, respectively, i.e., $-\log \text{Softmax}(z'_t, y_t)$ and $||z'_t - y_t||^2$. As a result, the scale of the normalization constraint approximates $(\boldsymbol{\Sigma}'_t)^{1/2}$ and $\boldsymbol{\Sigma}'_t$ in loss terms.

3.3 VIP-MTL

Under MTL paradigm, we incorporate the variance normalization on the probabilistic decoding framework, named variance-invariant probabilistic representation (VIP-MTL). The total objective of VIP-MTL can be:

$$\begin{aligned}\mathcal{L}_{total}(\theta, \{\phi_t\}_{t=1}^T) = \ & \mathbb{E}_{t\sim T, z\sim p_\theta(z|x)}\{\mathbb{E}_{z_t\sim q_{\phi_t^*}(z_t|z)}[-\log s(y_t|z_t)] \\ & + \beta KL(q_{\phi_t^*}(z_t|z); r(z_t)) + \gamma \log \tau_t\},\end{aligned} \quad (6)$$

where $z = p_\theta(x)$, learned by the shared encoder θ. $q'_{\phi_t^*}(z_t|z)$ is a variational estimate of the posterior probability of t and is learned by the t-th stochastic decoder ϕ_t^* with variance-invariant normalization. $\phi_t^* = \{\phi_t, \tau_t\}$. τ_t is a linear mapping of task t, which represents the approximated variance of a mixture distribution for task t. $\beta > 0$ controls the closeness between the learnable variational Gaussian posterior $q_{\phi_t}(z_t|z)$ and the standard Gaussian prior $r(z_t)$. $\gamma > 0$ is another Lagrange term that constrains the variance τ_t of a mixture distribution for task t. Totally, VIP-MTL can ensure impartial learning by harmonizing task-specific representation spaces.

4 Experiments

4.1 Experimental Setups

Datasets and Tasks We conduct experiments on two multi-task benchmarks, i.e., TweetEval and AffectEval. **TweetEval** (Barbieri et al., 2020) consists of 6 text classification tasks about tweet analysis on social media, EmotionEval (Mohammad et al., 2018) for social emotion detection, HatEval (Basile et al., 2019) for hate speech detection, IronyEval (Hee et al., 2018) for irony detection, OffensEval (Zampieri et al., 2019) for offensive language detection, SentiEval (Rosenthal et al., 2017) for sentiment analysis, and StanceEval (Mohammad et al., 2016) for stance detection. **AffectEval** involves 2 classification tasks and 2 regression tasks in a heterogeneous multi-task setting, i.e., GoEmotions (Demszky et al., 2020) for fine-grained emotion detection, EmotionEval (Mohammad et al., 2018), Emobank (Buechel and Hahn, 2017) for emotion regression, and EI-Reg (Mohammad et al., 2018) for emotion intensity regression. See Appendix B.1 for more detailed descriptions.

Comparison Methods We compare with the following 12 representative methods including Equal Weighting (EW), Scale-invariant Loss (SI), Task Weighting (TW), Uncertainty Weighting (UW) (Kendall et al., 2018), Geometric Loss Strategy (GLS) (Chennupati et al., 2019), Dynamic Weight Average (DWA) (Liu et al., 2019a), Projecting Conflicting Gradient (PCGrad) (Yu et al., 2020), IMTL-L (Liu et al., 2021b), Random Loss Weighting

Methods	TweetEval				AffectEval			
	BERT backbone		*RoBERTa backbone*		*BERT backbone*		*RoBERTa backbone*	
	Avg.	**Δp $\uparrow$**	**Avg.**	**Δp $\uparrow$**	**Avg.**	**Δp $\uparrow$**	**Avg.**	**Δp $\uparrow$**
EW (baseline)	65.62$\pm$0.57	0.00	66.17$\pm$0.43	0.00	52.93$\pm$2.02	0.00	57.64$\pm$2.12	0.00
SI	65.67$\pm$0.66	+0.06	67.16$\pm$1.08	+1.75	53.49$\pm$1.89	+1.80	57.94$\pm$2.02	+0.61
TW	65.68$\pm$0.54	+0.11	67.08$\pm$1.17	+1.55	53.27$\pm$2.12	+0.82	57.70$\pm$1.63	+0.09
UW	66.97$\pm$0.51	+2.22	67.11$\pm$3.47	+1.92	53.79$\pm$1.85	+1.81	59.69$\pm$1.10	+4.05
GLS	66.05$\pm$1.49	+0.60	67.32$\pm$0.38	+1.67	54.56$\pm$0.36	+9.82	57.66$\pm$1.65	-0.23
DWA	65.56$\pm$0.57	-0.09	66.94$\pm$1.13	+1.35	52.88$\pm$1.88	-0.25	57.36$\pm$2.53	-0.51
PCGrad	65.45$\pm$0.33	-0.50	67.42$\pm$0.30	+1.96	51.62$\pm$0.51	-3.09	56.27$\pm$2.16	-2.73
IMTL-L	66.18$\pm$1.45	+0.86	66.54$\pm$1.50	+0.67	53.89$\pm$0.42	+3.41	57.73$\pm$1.20	+0.05
RLW	66.76$\pm$1.42	+1.86	67.07$\pm$0.73	+1.63	51.38$\pm$1.42	-3.03	55.61$\pm$2.32	-4.26
MT-VIB	65.80$\pm$0.23	+0.66	67.14$\pm$0.87	+2.00	50.13$\pm$0.71	-5.09	57.68$\pm$1.56	+0.36
VMTL	65.80$\pm$1.59	+0.65	67.05$\pm$1.06	+1.81	50.02$\pm$0.76	-5.01	57.52$\pm$0.48	+0.20
Hierarchical MTL	66.42$\pm$0.10	+1.76	66.84$\pm$1.68	+1.60	50.55$\pm$0.65	-4.19	55.18$\pm$0.58	-4.74
VIP-MTL (ours)	**67.42**$^*\pm$1.06	**+3.11**	**69.09**$^*\pm$0.09	**+5.06**	**58.16**$^*\pm$0.45	**+17.80**	**61.40**$^*\pm$0.58	**+7.66**

Table 1: Multi-task performance (%) on TweetEval and AffectEval. For all methods with BERT/RoBERTa backbone, we run three random seeds and report the average result on test sets. Best results are highlighted in bold. * represents statistical significance over scores of the baseline under the t-test ($p < 0.05$).

Methods	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**	**Δp $\uparrow$**
EW (baseline)	74.37$\pm$0.56	44.08$\pm$5.26	65.32$\pm$1.84	79.04$\pm$1.43	70.64$\pm$1.71	63.59$\pm$2.43	66.17$\pm$0.43	0.00
SI	75.81$\pm$1.05	46.19$\pm$6.01	66.17$\pm$5.81	78.58$\pm$2.00	71.00$\pm$1.80	65.24$\pm$2.31	67.16$\pm$1.08	+1.75
UW	74.76$\pm$3.08	48.49$\pm$3.21	65.41$\pm$7.01	79.49$\pm$1.48	71.56$\pm$0.74	62.96$\pm$6.84	67.11$\pm$3.47	+1.92
GLS	75.47$\pm$1.15	43.97$\pm$1.13	**69.18**$\pm$2.62	79.46$\pm$0.84	**71.84**$\pm$0.38	64.01$\pm$0.71	67.32$\pm$0.38	+1.67
IMTL-L	75.25$\pm$1.26	45.61$\pm$3.84	65.94$\pm$0.74	79.59$\pm$1.28	71.19$\pm$0.60	61.65$\pm$5.41	66.54$\pm$1.50	+0.67
MT-VIB	74.74$\pm$0.38	48.06$\pm$4.79	66.09$\pm$3.38	78.17$\pm$1.39	70.95$\pm$0.99	64.83$\pm$1.56	67.14$\pm$0.87	+2.00
VMTL	74.07$\pm$0.72	47.44$\pm$3.42	68.55$\pm$2.80	77.95$\pm$0.22	70.52$\pm$1.04	63.76$\pm$2.86	67.05$\pm$1.06	+1.81
VIP-MTL (ours)	**77.36**$^*\pm$0.53	**49.79**$^*\pm$1.37	68.65$^*\pm$1.74	**79.60**$^*\pm$0.89	71.32$^*\pm$0.49	**67.80**$^*\pm$0.33	**69.09**$^*\pm$0.09	**+5.06**

(a) Fine-grained results on TweetEval

Methods	GoEmotions M-F1	EmotionEval M-F1	Emobank V	Emobank A	Emobank D	EI-Reg Pear	EI-Reg Spear	**Avg.**	**Δp $\uparrow$**
EW (baseline)	47.13$\pm$0.33	77.97$\pm$0.63	75.62$\pm$0.79	49.44$\pm$4.70	36.47$\pm$4.02	51.01$\pm$4.62	52.23$\pm$4.68	57.64$\pm$2.12	0.00
SI	47.08$\pm$0.72	78.22$\pm$0.49	75.61$\pm$1.39	50.35$\pm$5.02	37.26$\pm$4.78	51.55$\pm$3.99	52.60$\pm$3.82	57.94$\pm$2.02	+0.61
UW	48.54$\pm$0.55	78.55$\pm$1.14	76.81$\pm$0.28	53.26$\pm$0.44	38.60$\pm$3.32	54.94$\pm$3.14	55.93$\pm$3.00	59.69$\pm$1.10	+4.05
GLS	37.15$\pm$0.43	79.43$\pm$1.34	80.18$\pm$1.47	55.07$\pm$1.07	45.73$\pm$0.61	53.15$\pm$6.16	54.31$\pm$5.96	57.66$\pm$1.65	-0.23
IMTL-L	46.71$\pm$0.38	79.08$\pm$1.02	75.18$\pm$1.03	50.99$\pm$2.68	37.05$\pm$2.13	50.34$\pm$2.94	51.12$\pm$2.78	57.73$\pm$1.20	+0.05
MT-VIB	46.92$\pm$0.29	76.66$\pm$2.31	75.61$\pm$1.96	51.60$\pm$1.01	37.50$\pm$5.59	51.80$\pm$1.39	52.64$\pm$2.19	57.68$\pm$1.56	+0.36
VMTL	46.83$\pm$0.23	75.25$\pm$1.70	77.38$\pm$0.44	51.02$\pm$1.52	37.77$\pm$8.17	51.35$\pm$2.81	53.83$\pm$2.05	57.52$\pm$0.48	+0.20
VIP-MTL (ours)	**49.38**$^*\pm$1.37	**79.47**$^*\pm$0.45	**78.55**$^*\pm$1.01	**55.51**$^*\pm$0.48	**45.73**$^*\pm$1.28	**56.46**$^*\pm$1.17	**57.19**$^*\pm$1.10	**61.40**$^*\pm$0.58	**+7.66**

(b) Fine-grained results on AffectEval

Table 2: Fine-grained results of representative comparison methods and our VIP-MTL. We experiment with the RoBERTa backbone. * represents statistical significance over scores of the baseline under the t-test ($p < 0.05$).

(RLW) (Lin et al., 2022), MT-VIB (Qian et al., 2020), VMTL (Shen et al., 2021), and Hierarchical MTL (de Freitas et al., 2022). Among them, MT-VIB, VMTL, and Hierarchical MTL are probabilistic MTL series. For fair comparison, we reproduce each method under the same experimental setups (e.g., the network backbone). We use a pre-trained language model BERT (Devlin et al., 2019)/RoBERTa (Liu et al., 2019c) as the backbone model. Specifically, we use *bert-base-uncased*[2] and *roberta-base*[2] to initialize BERT and RoBERTa for fine-tuning. We also compare with large language model (LLM) GPT-3.5[3] and single task learning (STL) baseline. Please see Appendix B.2 for details of comparison methods.

Evaluation Metrics We utilize the same evaluation metrics as those used in the original tasks. For classification tasks, the macro-averaged F1 over all classes is employed with three exceptions: stance (macro-averaged of F1 of favor and against classes), irony (F1 of ironic class), and sentiment analysis (macro-averaged recall). For regression tasks, we compute Pearson correlation for each VAD dimension on EmoBank, and use both Pearson and Spearman correlation coefficients on EI-Reg. Following Barbieri et al. (2020), we report a global metric (**Avg.**) based on the average of all task-specific metrics. Following Maninis et al. (2019); Liu et al. (2021a), we also report the average relative improvement over EW baseline on each metric of

[2] https://huggingface.co/
[3] https://chat.openai.com

Methods	# Param	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**
GPT-3.5	(LLMs)	73.23	48.30	66.81	63.71	40.40	39.45	55.32
STL	6×110M	74.49	45.26	53.27	79.20	**72.43**	66.70	65.23
STL with CNN	110M+6×2M	59.11	47.61	52.10	77.80	70.85	57.58	60.84
VIP-MTL	**110M**	**77.29**	**49.73**	**67.88**	**80.02**	71.15	**67.28**	**68.89**

Table 3: Comparison results with different learning paradigms on TweetEval. We experiment with RoBERTa backbone for all methods except for GPT-3.5. STL stands for single-task learning with a cross-entropy loss. STL with CNN indicates fine-tuning task-specific CNN classifiers with a frozen RoBERTa backbone. # Param refers to the number of parameters of the model for all tasks excluding the task-specific linear head.

Methods	TweetEval				AffectEval			
	BERT backbone		*RoBERTa backbone*		*BERT backbone*		*RoBERTa backbone*	
	Avg.	$\mathbf{\Delta p}\uparrow$	**Avg.**	$\mathbf{\Delta p}\uparrow$	**Avg.**	$\mathbf{\Delta p}\uparrow$	**Avg.**	$\mathbf{\Delta p}\uparrow$
VIP-MTL	**67.42**$\pm$1.06	**+3.11**	**69.09**$\pm$0.09	**+5.06**	**58.16**$\pm$0.45	**+17.80**	**61.40**$\pm$0.58	**+7.66**
w/o VI	65.36$\pm$1.14	-0.58	67.59$\pm$1.06	+2.72	53.08$\pm$1.89	+5.17	58.21$\pm$1.96	+1.46
w/o VIP	65.62$\pm$0.57	0.00	66.17$\pm$0.43	0.00	52.93$\pm$2.02	0.00	57.64$\pm$2.12	0.00

Table 4: Ablation study results of our VIP-MTL.

Methods	TweetEval		AffectEval	
	Avg.	$\mathbf{\Delta p}\uparrow$	**Avg.**	$\mathbf{\Delta p}\uparrow$
EW	66.56	0.00	59.47	0.00
VIP-MTL (β=0.001)	**68.75**	**+4.42**	**61.40**	**+7.66**
w/o VI	67.79	+3.26	58.21	+1.46
VIP-MTL (β=0.01)	**69.09**	**+5.06**	**60.52**	**+6.09**
w/o VI	67.59	+2.72	57.45	-0.09
VIP-MTL (β=0.1)	**68.27**	**+3.98**	**60.69**	**+6.42**
w/o VI	67.81	+2.75	56.28	-2.17

Table 5: Results with different sampling distributions.

each task as the performance measure, denoted as $\mathbf{\Delta}p$. See Appendix B.3 for details of metrics. Additionally, we use t-test (Kim, 2015) to verify the statistical significance of differences between results of VIP-MTL and the baseline on the task.

Implementation Details All experiments are conducted on a single NVIDIA Tesla A100 80GB card. The validation sets are used to tune hyperparameters and choose the optimal model. For each method, we run three random seeds and report the average result of the test sets. The network parameters are optimized by using Adamax optimizer (Kingma and Ba, 2015) with the learning rate of $5e^{-5}$. The dropout rate is set to 0.2 for TweetEval and 0 for AffectEval. β is searched from $\{0.001, 0.01, 0.1\}$. γ is searched from $\{1, 10\}$ and $\{0.1, 1\}$ for classification and regression. More details are listed in Appendix B.4.

4.2 Main Results

Overall Results for MTL The overall results on both benchmarks are reported in Table 1, where the homogeneous TweetEval contains six different classification tasks, and heterogeneous AffectEval includes two classification and two regression tasks. VIP-MTL consistently obtains the best average performance over comparison methods on both benchmarks with different backbone models. Specifically, compared to EW baseline, VIP-MTL with BERT/RoBERTa backbone improves Avg. by **+1.80%/+2.92%** and increases Δp by **+3.11%/+5.06%** on TweetEval. VIP-MTL with BERT/RoBERTa backbone gains improvements in Avg. by **+5.23%/+3.76%** and an increase in Δp by **+17.80%/+7.66%** on AffectEval.

Fine-grained Results Table 2 summarizes fine-grained results of VIP-MTL, the EW baseline, and 6 representative comparison MTL methods (including 4 task-balanced and 2 probabilistic methods). Our VIP-MTL consistently outperforms the EW baseline on all tasks of both benchmarks, achieving the best fine-grained results on most tasks. This indicates the effectiveness of VIP-MTL.

Comparison with STL and LLM We compare our VIP-MTL with the single-task learning (STL) baseline and the large language model (LLM) GPT-3.5. For STL, each task is trained with a separate model. For GPT-3.5, predictions are made under the zero-shot setting using task descriptions and instructions. As shown in Table 3, our VIP-MTL outperforms GPT-3.5 on all tasks significantly. Compared to the STL baselines, our method also achieves superior results on most tasks with the same scale of model parameters.

4.3 Ablation Study

We conduct ablation studies by removing the variance normalization (w/o VI) and further removing probabilistic representation (w/o VIP). As shown

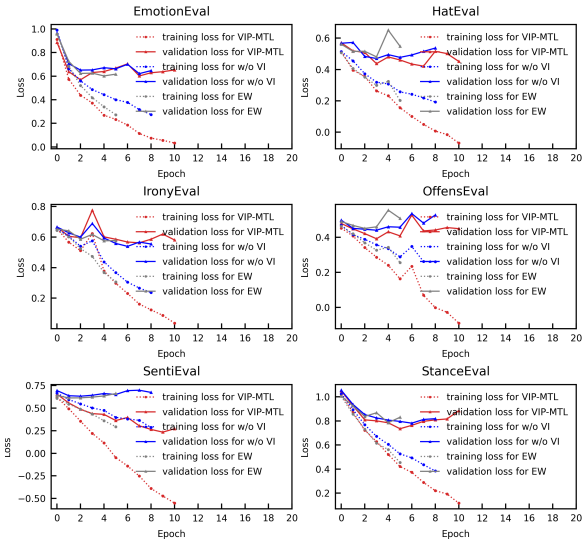


Figure 2: Loss analysis during training on TweetEval.

in Table 4, compared with two ablation models, the full VIP-MTL consistently obtains the best performance in terms of Avg. and Δp on TweetEval and AffectEval. The results reveal the effectiveness of both components for MTL. Additionally, VIP-MTL applies variance normalization to constrain task-specific probabilistic distribution to a consistent scale, showing a smaller variance than the ablation w/o VI on all benchmarks.

4.4 Robustness Evaluation on Sampling Distribution

We evaluate the robustness on different sampling distributions. β controls the closeness between the learnable variational Gaussian posterior distribution and predefined standard Gaussian prior. We adjust values of β to obtain sampling distributions with different Gaussian forms. As shown in Table 5, VIP-MTL outperforms EW baseline across different posterior distributions, which shows the robustness of VIP-MTL on sampling distribution. Additionally, compared with w/o VI, VIP-MTL consistently achieves superior performance across different values of β. It indicates that variance normalization exhibits promising performance under different probabilistic distributions.

4.5 Optimization Efficiency Evaluation

We further evaluate optimization efficiency on the MTL paradigm. Figure 2 shows loss curves for each task on TweetEval. VIP-MTL performs better on both the training and validation sets and converges faster, indicating that the optimization process is more efficient. From results, we have: 1) VIP-MTL exhibits a steeper slope in the training loss for each task, particularly during the early stages of training. This indicates that the method is capable of reducing the training error for multiple tasks more rapidly during the training process. 2) During the training process, the validation loss of VIP-MTL is lower than that of other methods in most cases (except during the early stages of training for IronyEval[4]), demonstrating that our VIP-MTL performs better on unseen data and possesses stronger multi-task generalization capabilities.

4.6 Evaluation of Scale-invariance Property

To analyze the impartial ability, we evaluate the scale-invariance property of pairwise task combinations within AffectEval. The scale invariance of a method generally refers to the invariance to individual loss scales. We experiment involving two heterogeneous and two homogeneous pair-wise MTL settings (More experimental details and results can be found in Appendix C.2). The results show that VIP-MTL achieves the best performance in terms of Avg. and Δp on all scenarios. Then, we show loss curves on pairwise task combinations in Figure 3 (loss curve results on other two task combinations are listed in Appendix C.4). The task losses obtained by VIP-MTL are closer to each other on both heterogeneous and homogeneous combinations, showing that our method is scale-invariant to task losses.

4.7 Evaluation under Data-constrained Conditions

We evaluate VIP-MTL and 7 representative comparison methods when training with limited data by adjusting different ratios of the training set. Following Hu et al. (2024a), all methods are trained on randomly sampled subsets from the original training set, and we report the average results on the test set. Table 6 shows overall results against different sizes of training set where RoBERTa is the default backbone model. VIP-MTL achieves superior average performance against different ratios of the training set. This suggests that VIP-MTL is capable of learning sufficient representations, improving the efficiency of utilizing limited data.

[4]In the early stage, VIP-MTL mainly focuses on balancing overall tasks rather than individual tasks, leading to IronyEval—requiring complex semantic understanding—receiving more attention only in the later stage of training.

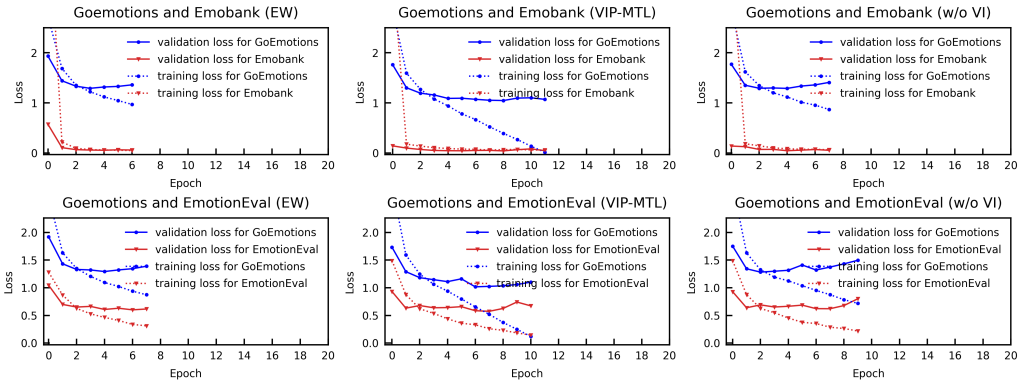


Figure 3: Loss analysis during training phase on pair-wise tasks on AffectEval. RoBERTa is the default backbone model. Results on other pair-wise task combinations are listed in Appendix C.4.

Methods	Data per	TweetEval		AffectEval	
		Avg.	**Δp $\uparrow$**	**Avg.**	**Δp $\uparrow$**
EW	20%	62.43	0.00	43.99	0.00
SI	20%	62.23	-0.34	43.08	-1.86
UW	20%	61.78	-1.59	48.93	+.17
GLS	20%	61.33	-2.32	49.32	+29.91
IMTL-L	20%	60.66	-3.38	48.94	+20.88
MT-VIB	20%	60.00	-4.18	44.35	+4.30
VMTL	20%	58.34	-7.30	42.82	-0.40
VIP-MTL	20%	**64.41**	**+3.20**	**50.51**	**+33.80**
EW	40%	66.01	0.00	51.03	0.00
SI	40%	65.95	-0.11	51.60	+0.68
UW	40%	64.35	-2.82	52.91	+5.60
GLS	40%	63.63	-4.13	54.07	+8.19
IMTL-L	40%	64.16	-3.22	51.00	+0.92
MT-VIB	40%	63.58	-3.90	49.42	-1.84
VMTL	40%	63.36	-4.33	49.37	-2.47
VIP-MTL	40%	**66.29**	**+0.73**	**56.74**	**+15.51**
EW	60%	66.38	0.00	55.03	0.00
SI	60%	66.31	-0.24	54.13	-1.71
UW	60%	66.17	-0.45	55.27	+1.00
GLS	60%	66.33	-0.04	56.10	+2.26
IMTL-L	60%	66.96	+1.02	54.99	+0.27
MT-VIB	60%	66.31	+0.04	52.85	-3.94
VMTL	60%	65.00	-1.95	53.47	-2.27
VIP-MTL	60%	**67.12**	**+1.35**	**58.79**	**+8.57**
EW	80%	66.34	0.00	56.75	0.00
SI	80%	67.33	+1.98	56.17	-1.13
UW	80%	66.93	+1.30	58.71	+4.49
GLS	80%	66.43	+0.23	57.05	+0.86
IMTL-L	80%	66.59	+0.84	56.31	-0.65
MT-VIB	80%	65.34	-1.57	54.80	-3.39
VMTL	80%	65.07	-2.33	55.72	-0.94
VIP-MTL	80%	**67.97**	**+2.73**	**60.54**	**+8.19**

Table 6: Results against different training data size.

4.8 Representation Quality Evaluation

To analyze the quality of the learned representations, we evaluate the clustering performance of output representations obtained by different objectives. Following Hu et al. (2024a), we apply silhouette coefficient (SC) and adjusted rand index (ARI) to measure the clustering ability relevant to input data and target tasks, respectively. Figure 4 shows SC and ARI values of representations. learned by 5 representative comparison objectives, VIP-MTL and its ablation w/o VI on TweetEval. Both VIP-MTL and its ablation w/o VI achieve higher ARI

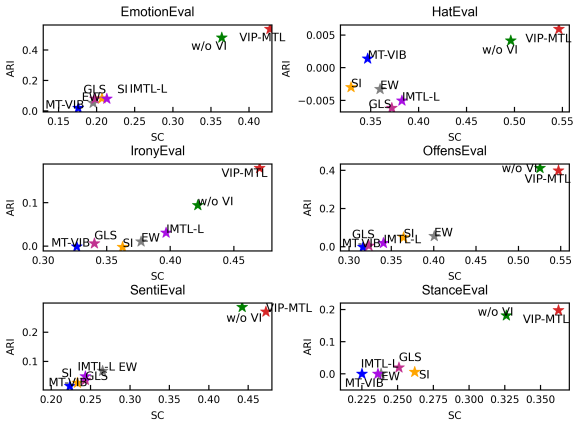


Figure 4: Quality analysis of the learned task-specific representations by different objectives. The X-axis and Y-axis refer to silhouette coefficient (SC) and adjusted rand index (ARI) of task-specific representations.

and SC values on six tasks. This reveals that our method can learn compact and informative output representations for all tasks.

5 Conclusion

This paper proposes a novel VIP-MTL to harmonize task-specific representation spaces to ensure impartial learning. VIP-MTL decodes task-agnostic shared representations into task-specific probabilistic distributions and applies variance normalization to constrain them to a consistent scale, balancing task influence during training. Experiments on two language benchmarks demonstrate that VIP-MTL achieves superior performance in heterogeneous and data-constrained MTL scenarios. Further analysis shows that VIP-MTL is robust to sampling distributions, efficient on optimization process, scale-invariant to task losses, and learns more informative task-specific representations.

Limitations

This paper conducts experiments on the task of natural language understanding such as classification and regression tasks. The performance on generation tasks is still unexplored, which will be left for the future work.

References

Cédric Archambeau, Shengbo Guo, and Onno Zoeter. 2011. Sparse bayesian multi-task learning. In *NeurIPS*, pages 1755–1763.

Bart Bakker and Tom Heskes. 2003. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99.

Francesco Barbieri, José Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In *EMNLP (Findings)*, pages 1644–1650.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *SemEval@NAACL-HLT*, pages 54–63.

Sven Buechel and Udo Hahn. 2017. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 578–585. Association for Computational Linguistics.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning, Proceedings of the Tenth International Conference, University of Massachusetts, Amherst, MA, USA, June 27-29, 1993*, pages 41–48. Morgan Kaufmann.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28:41–75.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 793–802. PMLR.

Sumanth Chennupati, Ganesh Sistu, Senthil Kumar Yogamani, and Samir A. Rawashdeh. 2019. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1200–1210. Computer Vision Foundation / IEEE.

João Machado de Freitas, Sebastian Berg, Bernhard C. Geiger, and Manfred Mücke. 2022. Compressed hierarchical representations for multi-task learning and task clustering. In *IJCNN*, pages 1–8.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan S. Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. In *ACL*, pages 4040–4054.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1923–1933. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *SemEval@NAACL-HLT*, pages 39–50.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John W. Paisley. 2013. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347.

Dou Hu, Lingwei Wei, Yaxin Liu, Wei Zhou, and Songlin Hu. 2024a. Structured probabilistic coding. In *AAAI*, pages 12491–12501.

Dou Hu, Lingwei Wei, Yaxin Liu, Wei Zhou, and Songlin Hu. 2024b. Structured probabilistic coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Hal Daumé III. 2009. Bayesian multitask learning with latent hierarchies. In *UAI*, pages 135–142.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7482–7491. Computer Vision Foundation / IEEE Computer Society.

Donggyun Kim, Seongwoong Cho, Wonkwang Lee, and Seunghoon Hong. 2022. Multi-task processes. In *ICLR*.

Tae Kyun Kim. 2015. T test as a parametric statistic. *Korean journal of anesthesiology*, 68(6):540–546.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W. Tsang. 2022. Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Trans. Mach. Learn. Res.*, 2022.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021a. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18878–18890.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021b. Towards impartial multi-task learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2019a. End-to-end multi-task learning with attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1871–1880. Computer Vision Foundation / IEEE.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4487–4496. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. 2019. Attentive single-tasking of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1851–1860. Computer Vision Foundation / IEEE.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3994–4003. IEEE Computer Society.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *SemEval@NAACL-HLT*, pages 1–17.

Saif M. Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *SemEval@NAACL-HLT*, pages 31–41.

Weizhu Qian, Bowei Chen, and Franck Gechter. 2020. Multi-task variational information bottleneck. *CoRR*, abs/2007.00339.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *SemEval@ACL*, pages 502–518. Association for Computational Linguistics.

Amelie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. 2023. Scalarization for multi-task and multi-domain learning at scale. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent multi-task architecture learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4822–4829. AAAI Press.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 525–536.

Jiayi Shen, Xiantong Zhen, Marcel Worring, and Ling Shao. 2021. Variational multi-task learning with gumbel-softmax priors. In *NeurIPS*, pages 21031–21042.

Naftali Tishby, Fernando C Pereira, and William Bialek. 1999. The information bottleneck method. In *Proc. of the 37th Allerton Conference on Communication and Computation*.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *ITW*, pages 1–5.

Michalis K. Titsias and Miguel Lázaro-Gredilla. 2011. Spike and slab variational inference for multi-task and multiple kernel learning. In *NeurIPS*, pages 2339–2347.

Matías Vera, Leonardo Rey Vega, and Pablo Piantanida. 2017. Compression-based regularization with an application to multi-task learning. *CoRR*, abs/1711.07099.

Luke Vilnis and Andrew McCallum. 2015. Word representations via gaussian embedding. In *ICLR*.

Fariba Yousefi, Michael Thomas Smith, and Mauricio A. Álvarez. 2019. Multi-task learning for aggregated data using gaussian processes. In *NeurIPS*, pages 15050–15060.

Kai Yu, Volker Tresp, and Anton Schwaighofer. 2005. Learning gaussian processes from multiple tasks. In *ICML*, volume 119, pages 1012–1019.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *SemEval@NAACL-HLT*, pages 75–86.

Appendix Overview

In this appendix, we provide: (i) the related work, (ii) detailed experimental setups, and (iii) supplementary results.

A Related Work

Existing works on multi-task learning (MTL) can be categorized into two groups: multi-task optimization and network architecture design.

A.1 Multi-task Optimization

The optimization of MTL aims to improve the MTL training process by balancing the training dynamics of different tasks. This line of studies typically employs a hard parameter-sharing pattern (Caruana, 1993), where several light-weight task-specific heads are attached upon the heavy-weight task-agnostic backbone. Recent works on multi-task optimization are roughly divided into two parts: task-balanced and probabilistic methods.

Task-balanced methods aims to balance the learning process across multiple tasks via loss-based and gradient-based methods. Loss-based methods focus on aligning the task losses magnitudes by rescaling loss scales (Kendall et al., 2018; Chennupati et al., 2019; Liu et al., 2019a, 2021b; Lin et al., 2022). These works can prevent MTL from being biased in favor of tasks with large loss scales, but cannot ensure the impartial learning of the shared parameters. Gradient-based methods (Sener and Koltun, 2018; Chen et al., 2018; Yu et al., 2020) aims to find an aggregated gradient to balance different tasks. Moreover, Liu et al. (2021b) and Lin et al. (2022) also provide the gradient-based version, and the overall effects are comparable to their loss-based version. While gradient balance can evenly learn task-shared parameters, they also incur a higher compute and memory training cost. Unlike existing optimization methods via balancing loss and gradients, this paper focus on directly constrain the representation space to address the task interference issue by probabilistic embedding.

Probabilistic methods aims to explore shared priors for all tasks (Yousefi et al., 2019; Kim et al., 2022; Qian et al., 2020; Shen et al., 2021; de Freitas et al., 2022). To explore task relatedness, some works study design priors over model parameters under the Bayesian framework (Yu et al., 2005; Titsias and Lázaro-Gredilla, 2011; Archambeau et al., 2011; Bakker and Heskes, 2003), or share the covariance structure of parameters (III, 2009). Additionally, some works (Vera et al., 2017; Qian et al., 2020; de Freitas et al., 2022) introduce the information bottleneck (IB) principle (Tishby et al., 1999; Tishby and Zaslavsky, 2015) into the information encoding process of MTL. They typically enhance the adaptability to noisy data by compressing task-irrelevant redundant information and learning compact intermediate representations. For example, Qian et al. (2020) use variational inference to learn probabilistic representations for multiple tasks based on the information bottleneck. de Freitas et al. (2022) propose a hierarchical variational MTL method that restricts information individual tasks can access from a task-agnostic latent representation.

A.2 Architectures for MTL

Orthogonal to our work, another line of studies emphasizes on designing neural network architectures by optimizing the allocation of shared versus task-specific parameters (Misra et al., 2016; Hashimoto et al., 2017; Ruder et al., 2019; Liu et al., 2019a,b). Some of these methods utilize soft parameter sharing, allowing for parameter sharing among tasks to a large extent. However, they often result in higher inference cost. The scope of our study is complementary to this line of work, since we focus on how to balancing multiple tasks that is agnostic to the architecture employed.

B Experimental Setups

B.1 Details of Datasets and Downstream Tasks

We conduct experiments on TweetEval and AffectEval benchmarks. The statistics are summarized in Table 7.

TweetEval benchmark contains 6 classification tasks. *EmotionEval* (Mohammad et al., 2018) involves detecting the emotion evoked by a tweet and is based on the Affects in Tweets conducted during SemEval-2018. Following Barbieri et al. (2020), the most common four emotions (i.e., anger, joy, sadness, and optimism) are selected as the label sets. *HatEval* (Basile et al., 2019) stems from SemEval-2019 Hateval challenge and is used to predict whether a tweet is hateful towards immigrants or women. *IronyEval* (Hee et al., 2018) is from SemEval-2018 Irony Detection and consists of identifying whether a tweet includes ironic intents or not. *OffensEval* (Zampieri et al., 2019) is

Dataset	Task	Task Type	# Label	# Train	# Val	# Test	# Total
Homogeneous benchmark: *TweetEval*							
EmotionEval	Social emotion detection	Classification	4	3,257	374	1,421	5,502
HatEval	Hate speech detection	Classification	2	9,000	1,000	2,970	12,970
IronyEval	Irony detection	Classification	2	2,862	955	784	4,601
OffensEval	Offensive language detection	Classification	2	11,916	1,324	860	14,100
SentiEval	Sentiment analysis	Classification	3	45,389	2,000	11,906	59,295
StanceEval	Stance detection	Classification	3	2,620	294	1,249	4,163
Heterogeneous benchmark: *AffectEval*							
GoEmotions	Fine-grained emotion detection	Classification	28	36,308	4,548	4,591	45,447
EmotionEval	Social emotion detection	Classification	4	3,257	374	1,421	5,502
EmoBank	Emotion regression	Regression	-	8,062	1,000	1,000	10,062
EI-Reg	Emotion intensity regression	Regression	-	7,102	1,464	4,068	12,634

Table 7: Dataset statistics on TweetEval and AffectEval. The homogeneous TweetEval contains six different classification tasks, and heterogeneous AffectEval includes two classification and two regression tasks.

Hyperparameter		TweetEval	AffectEval
BERT	Trade-off weight β	0.001	0.1
	Trade-off weight γ	10 for Cls. and 0.1 for Reg.	
	Number of epochs	20	20
	Patience	3	3
	Max length	256	256
	Batch size	128	128
	Dropout	0.2	0
	Learning rate	$5e^{-5}$	$5e^{-5}$
RoBERTa	Trade-off weight β	0.01	0.001
	Trade-off weight γ	10 for Cls. and 0.1 for Reg.	
	Number of epochs	20	20
	Patience	3	3
	Max length	256	256
	Batch size	128	128
	Dropout	0.2	0
	Learning rate	$5e^{-5}$	$5e^{-5}$

Table 8: Hyperparameters of VIP-MTL on TweetEval and AffectEval.

from SemEval-2019 OffensEval and involves predicting if a tweet contains any form of offensive language. *SentiEval* (Rosenthal et al., 2017) comes from SemEval 2017 and includes data from previous runs (2013, 2014, 2015, and 2016) of the same task. The goal is to determine if a tweet is positive, negative, or neutral. *StanceEval* (Mohammad et al., 2016) involves determining if the author of a piece of text has a favorable, neutral, or negative position towards a proposition or target.

AffectEval includes 2 classification and 2 regression tasks. *GoEmotions* (Demszky et al., 2020) is a corpus of comments from Reddit, with human annotations to 27 emotion categories or neutral. It is used fine-grained emotion prediction. Following Hu et al. (2024b), nearly 16% of multi-label data was removed from the source corpus to better evaluate the performance of multi-class classification. *EmotionEval* (Mohammad et al., 2018) involves detecting the emotion evoked by a tweet and is based on the Affects in Tweets conducted during SemEval-2018. *Emobank* (Buechel and Hahn, 2017) is a large-scale text corpus across 6 domains and 2 perspectives and manually annotated with continuous VAD scores. Each sentence has three scores representing VAD in the range of 1 to 5. Following Buechel and Hahn (2017), we use the average of VAD scores as the overall metric. *EI-Reg* (Mohammad et al., 2018) is an emotion intensity regression task and is from SemEval-2018 Task 1: Affect in Tweets. The goal is to determine the intensity of the emotion E that best represents the mental state of the twitter. The intensity is a real-valued score between 0 (least E) and 1 (most E). In this task, we did not use additional emotion labels in the dataset to better evaluate this regression task.

B.2 Description of Comparison Methods

Equal Weighting (EW) is a typical baseline that applies equal weights for each task. **Task Weighting** (TW) utilizes loss weights to each task based on the ratio of task examples. **Uncertainty weighting** (UW) (Kendall et al., 2018) uses the homoscedastic uncertainty quantification to adjust task weights. **Geometric Loss Strategy** (GLS) (Chennupati et al., 2019) uses the geometric mean of task losses to the weighted average of task losses. **Dynamic Weight Average** (DWA) (Liu et al., 2019a) sets the loss weight of each task to be the ratio of two adjacent losses. **PCGrad** (Yu et al., 2020) removes conflicting components of each gradient w.r.t the other gradients. **IMTL-L** (Liu et al., 2021b) dynamically reweighs the losses such that they all have the same magnitude. **Random Loss Weighting** (RLW) (Lin et al., 2022) with normal distribution, scales the losses according to randomly sampled task weights. **MT-VIB** (Qian et al., 2020) is a

Methods	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**	$\mathbf{\Delta p} \uparrow$
VIP-MTL	**77.36**$\pm$0.53	**49.79**$\pm$1.37	**68.65**$\pm$1.74	79.60$\pm$0.89	**71.32**$\pm$0.49	**67.80**$\pm$0.33	**69.09**$\pm$0.09	**+5.06**
w/o VI	76.02$\pm$1.10	49.30$\pm$3.75	64.63$\pm$3.14	79.44$\pm$1.93	71.67$\pm$1.25	64.47$\pm$1.65	67.59$\pm$1.06	+2.72
w/o VI & Pro.	74.37$\pm$0.56	44.08$\pm$5.26	65.32$\pm$1.84	79.04$\pm$1.43	70.64$\pm$1.71	63.59$\pm$2.43	66.17$\pm$0.43	0.00

(a) Ablation results on TweetEval

Methods	GoEmotions M-F1	EmotionEval M-F1	Emobank V	Emobank A	Emobank D	EI-Reg Pear	EI-Reg Spear	**Avg.**	$\mathbf{\Delta p} \uparrow$
VIP-MTL	**49.38**$\pm$1.37	**79.47**$\pm$0.45	**78.55**$\pm$1.01	**55.51**$\pm$0.48	**45.73**$\pm$1.28	**56.46**$\pm$1.17	**57.19**$\pm$1.10	**61.40**$\pm$0.58	**+7.66**
w/o VI	48.87$\pm$0.79	78.15$\pm$0.57	74.23$\pm$4.01	51.02$\pm$3.15	39.62$\pm$3.19	50.62$\pm$3.79	51.16$\pm$4.20	58.21$\pm$1.96	+1.46
w/o VI & Pro.	47.13$\pm$0.33	77.97$\pm$0.63	75.62$\pm$0.79	49.44$\pm$4.70	36.47$\pm$4.02	51.01$\pm$4.62	52.23$\pm$4.68	57.64$\pm$2.12	0.00

(b) Ablation results on AffectEval

Table 9: Fine-grained ablation study of VIP-MTL. We experiment with the RoBERTa backbone.

Methods	GoEmotions M-F1	Emobank V	Emobank A	Emobank D	**Avg.**	$\mathbf{\Delta p} \uparrow$
EW (baseline)	46.69	73.10	48.17	33.09	49.07	0.00
SI	46.59	73.10	49.04	34.59	49.42	+0.95
UW	48.91	77.70	53.97	44.87	53.88	+11.36
GLS	46.23	**79.24**	51.73	44.27	52.32	+7.77
IMTL-L	48.37	76.18	51.82	38.48	51.93	+6.47
MT-VIB	46.28	74.65	48.84	30.83	48.86	-1.00
VMTL	46.32	75.61	51.30	41.06	51.16	+5.27
VIP-MTL	**50.67**	78.86	**55.84**	**45.81**	**55.42**	**+14.63**

(a)

Methods	EmotionEval M-F1	EI-Reg Pear	EI-Reg Spear	**Avg.**	$\mathbf{\Delta p} \uparrow$
EW (baseline)	76.96	55.94	56.38	66.56	0.00
SI	78.07	55.44	56.36	66.99	+0.49
UW	78.83	59.26	59.95	69.22	+4.28
GLS	77.48	59.49	60.19	68.66	+3.62
IMTL-L	78.83	59.62	**60.30**	69.40	+4.60
MT-VIB	76.53	58.74	59.11	67.72	+2.18
VMTL	75.14	58.93	59.39	67.15	+1.48
VIP-MTL	**79.30**	**60.20**	59.85	**69.66**	**+4.96**

(b)

Table 10: Results on heterogeneous multi-task scenarios. We experiment with the RoBERTa backbone.

Methods	Emobank A	Emobank V	Emobank D	EI-Reg Pear	EI-Reg Spear	**Avg.**	$\mathbf{\Delta p} \uparrow$
EW (baseline)	79.40	55.52	46.71	57.96	58.83	59.47	0.00
SI	80.50	56.35	49.38	59.82	60.50	61.12	+2.94
UW	**81.40**	51.34	44.99	61.48	62.19	60.54	+1.50
GLS	80.11	56.23	48.13	60.05	60.90	60.98	+2.65
IMTL-L	81.32	50.79	44.58	**61.93**	**62.48**	60.55	+1.48
MT-VIB	79.92	54.83	47.39	60.09	60.74	60.57	+1.88
VMTL	79.70	54.87	46.94	59.75	60.49	60.31	+1.43
VIP-MTL	81.21	**56.61**	**50.94**	60.79	61.40	**62.01**	**+4.53**

(a)

Methods	GoEmotions M-F1	EmotionEval M-F1	**Avg.**	$\mathbf{\Delta p} \uparrow$
EW (baseline)	46.69	77.05	61.87	0.00
SI	47.13	77.09	62.11	+0.50
UW	48.01	77.23	62.62	+1.53
GLS	42.41	**79.02**	60.72	-3.31
IMTL-L	47.62	76.94	62.28	+0.93
MT-VIB	46.19	77.99	62.09	+0.08
VMTL	46.05	77.20	61.63	-0.59
VIP-MTL	**50.17**	78.07	**64.12**	**+4.39**

(b)

Table 11: Results on homogeneous multi-task scenarios. We experiment with the RoBERTa backbone.

variational MTL method based on information bottleneck. **VMTL** (Shen et al., 2021) is a variational MTL framework that uses Gumbel-Softmax priors for both representations and weights. **Hierarchical MTL** (de Freitas et al., 2022) is a hierarchical variational MTL method with compressed task-specific representations based on information bottleneck.

For LLM, we compare with GPT-3.5, an enhanced generative pre-trained transformer model based on text-davinci-003[5], optimized for chatting.

B.3 Evaluation Metrics

The average performance $\mathbf{Avg.}$ is computed as,

$$\mathbf{Avg.} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{n-1}^{N_t} M_{t,n},$$

where $M_{t,n}$ denotes the performance of a task balancing method for the n-th metric in task t. N_t denotes the number of metrics in task t. T refers to the number of tasks.

$\mathbf{\Delta p}$ measures the average of the relative improvement over the baseline EW on each metric of each task, i.e.,

$$\mathbf{\Delta p} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{n-1}^{N_t}\frac{(-1)^{p_{t,n}}(M_{t,n} - M_{t,n}^{EW})}{M_{t,n}^{EW}},$$

where $M_{t,n}^{EW}$ is the n-th metric score for EW on task t. $p_{t,n} = 0$ if a higher value is better for the n-th metric in task t and 1 otherwise (Maninis et al., 2019; Liu et al., 2021a).

B.4 Implementation Details

We conduct experiments using an epoch number of 20, a total batch size of 128, and a maximum

[5]We present the results of the snapshot from June 13th 2023 based on specific inputs, including task descriptions, task instructions, and evaluation texts.

Methods	TweetEval				AffectEval			
	BERT backbone		*RoBERTa backbone*		*BERT backbone*		*RoBERTa backbone*	
	Avg.	**Δp $\uparrow$**	**Avg.**	**Δp $\uparrow$**	**Avg.**	**Δp $\uparrow$**	**Avg.**	**Δp $\uparrow$**
EW	49.07	0.00	66.56	0.00	61.87	0.00	59.47	0.00
VIP-MTL (β =0.001)	67.42	+3.11	68.75	+4.42	56.73	+13.95	61.40	+7.66
w/o VI	65.36	-0.58	67.79	+3.26	52.90	+2.63	58.21	+1.46
VIP-MTL (β =0.01)	66.75	+1.96	69.09	+5.06	56.43	+11.59	60.52	+6.09
w/o VI	65.16	-0.84	67.59	+2.72	53.21	+3.72	57.45	-0.09
VIP-MTL (β =0.1)	67.18	+2.62	68.27	+3.98	58.16	+17.80	60.69	+6.42
w/o VI	65.40	-0.42	67.81	+2.75	53.08	+5.17	56.28	-2.17

Table 12: Results with different sampling distributions

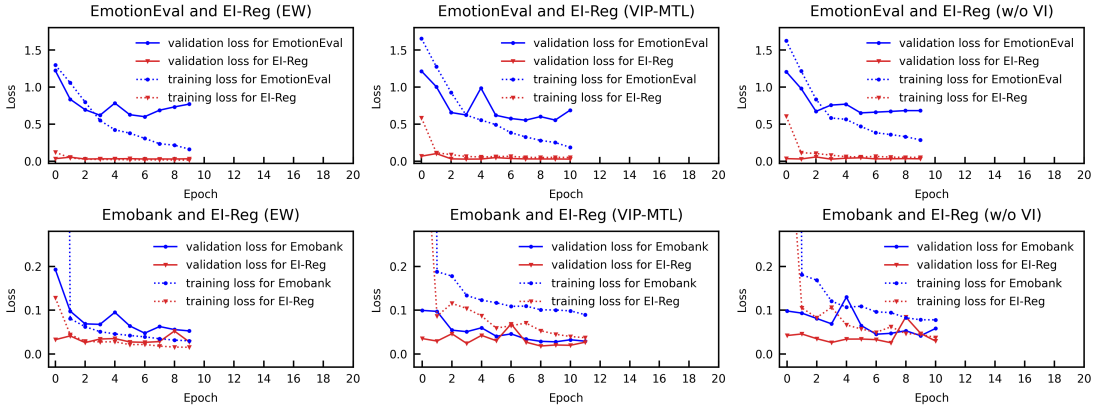


Figure 5: Loss analysis during training phase on pair-wise tasks on AffectEval. RoBERTa is the default backbone model.

token length of 256. The maximum patience for early stopping is set to 3 epochs. Following Liu et al. (2019b), we clip the gradient norm within 1 for all methods to avoid the exploding gradient problem. We report the detailed hyperparameter settings of VIP-MTL with RoBERTa and BERT backbone models on two benchmarks in Table 8.

For each comparison method, we fine-tune the key parameters following the original paper for fair comparison and to obtain corresponding optimal performance.

C Supplementary Results

C.1 Fine-grained Results of Ablation Study

Table 9 shows fine-grained ablation results of each task on TweetEval and AffectEval.

C.2 Fine-grained Results across Different Pair-wise Task Combinations

For multi-task evaluations on pairs of tasks, we consider two distinct combinations of tasks: homogeneous scenarios (i.e., *EmotionEval & GoEmotions*, and *Emobank & EI-Reg*), and heterogeneous scenarios (i.e., *EmotionEval & EI-Reg*, and *GoEmotions & Emobank*). Table 10 and Table 11 shows fine-grained results across pair-wise heterogeneous and homogeneous MTL scenarios. VIP-MTL achieves the best performance in terms of Avg. and Δp on all scenarios. This emphasizes the effectiveness of VIP-MTL in both heterogeneous and homogeneous MTL settings.

C.3 Supplementary Results of Sampling Distribution

Table 12 shows results against different sampling distributions.

C.4 Supplementary Results of Scale-invariance Property Evaluation

Figure 5 shows loss curves on two pairwise task combinations with AffectEval.