Multi-Task Representation Alignment on Language Understanding: A Mutual Information Perspective

Dou Hu[1] **and Lingwei Wei**[2*] **and Hongjiang Xiao**[1*] **and Songlin Hu**[2,3] **and Yuan Zhang**[1]
[1] State Key Laboratory of Media Convergence and Communication,
Communication University of China
[2] Institute of Information Engineering, Chinese Academy of Sciences
[3] School of Cyber Security, University of Chinese Academy of Sciences
{hudou, xiaohj, yzhang}@cuc.edu.cn, {weilingwei, husonglin}@iie.ac.cn

Abstract

Multi-task learning (MTL) enables joint learning over multiple tasks based on shared representations, but suffers from task interference issue during optimization. Existing works mainly focus on task balancing or probabilistic modeling but fail to address the issue since they struggle to learn sufficient representations for all target tasks. To address this, we propose a multi-task representation alignment (MTRA) framework to achieve task-specific alignment and self-alignment on the shared representations from a mutual information perspective. MTRA ensures that the learned representations contain task-relevant features while mitigating the negative effects of task-irrelevant features. First, we design a task-specific alignment objective to align the shared representations and task-specific representations with the expected targets of all tasks via information maximization. Besides, we design a self-alignment objective to eliminate task-irrelevant features via conditional information minimization. Experiments on two multi-task language benchmarks show that MTRA outperforms 13 representative MTL methods under the same settings, particularly under label-noisy and data-constrained conditions. Further analysis shows that the learned shared representations exhibit sufficient task informativeness and superior alignment properties.

1 Introduction

Multi-task learning (MTL) has become a powerful paradigm in machine learning, enabling joint learning over multiple tasks and making predictions based on shared representations (Caruana, 1993; Ruder et al., 2019). Unlike single-task learning, MTL can improve training efficiency and reduce computational costs. The optimization of MTL aims to optimize learning process across multiple 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. Different from the soft parameter sharing that allowing for controlled parameter sharing among tasks, the hard pattern has the advantages of lower inference cost and simpler requirements on the network architecture.

Multi-task optimization often suffers from task interference (Standley et al., 2020; Xin et al., 2022) where optimizing for one task degrades performance on others. Existing works mainly balances the learning process of different tasks via adjusting loss weights or gradients (Kendall et al., 2018; Chennupati et al., 2019; Liu et al., 2019a; Yu et al., 2020; Liu et al., 2021b; Lin et al., 2022), or facilitate cross-task cooperation via probabilistic modeling (Qian et al., 2020; Shen et al., 2021; de Freitas et al., 2022; Hu et al., 2025b). However, both task balancing and cross-task cooperation struggle to learn sufficient representations for all target tasks, resulting in the task interference issue remaining unresolved. During training, neural networks often compress information (Shwartz-Ziv and Tishby, 2017; Kawaguchi et al., 2023), which leads to the loss of features valuable for other tasks when the shared representations are optimized for specific ones. The shared representations would retain some task-irrelevant features and face the task-specific insufficiency issue (Hu et al., 2025b). As a result, the learned representations struggle to align with the expected objectives of all tasks, leading to imbalance and competition across tasks. As shown in left part of Figure 1, the compressed shared representations Z learned by vanilla MTL objective are typically insufficient for target tasks. For instance, Z in the red circle lose necessary features for task Y_a, as shown in red-shaded region.

In this paper, we propose a new multi-task representation alignment (MTRA) framework to achieve task-specific alignment and self-alignment on the

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 45722–45740
July 2-7, 2026 ©2026 Association for Computational Linguistics

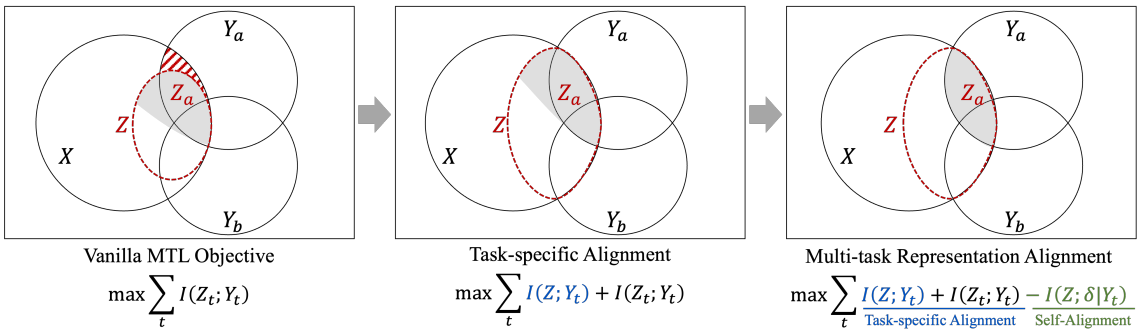

Figure 1: Venn information diagram comparison of our methods with vanilla MTL objective. Given the input X, shared representations Z, task-specific output representations Z_t, and the prediction $\hat{Y}_t$, the Markov chain for each task t is $Y_t \rightarrow X \rightarrow Z \rightarrow Z_t \rightarrow \hat{Y}_t$. We take two tasks as an example, namely Y_a and Y_b. The shared representations Z is circled by the red dashed line. Task-specific representations Z_t are marked by gray shading.

shared representations from a mutual information perspective. The framework ensures that the learned representations contain task-relevant features while mitigating the negative effects of task-irrelevant features. It can enhance language understanding of pre-trained language models (PLMs) under the multi-task paradigm.

First, we design a task-specific alignment objective to align the shared representations and task-specific representations with the expected targets of all tasks. It maximizes the mutual information between the shared representations Z and all task targets Y_t as well as maximizes the mutual information between the task-specific representations Z_t and Y_t, i.e., $\max \sum_t I(Z; Y_t) + I(Z_t; Y_t)$. In the implementation, we apply an MINE estimator based on task loss to approximate the lower bound of $I(Z; Y_t)$. Unlike vanilla MTL objective that maximizes $I(Z_t; Y_t)$, which can be viewed as a lower bound of $I(Z; Y_t)$, our method explicitly estimates $I(Z; Y_t)$ using MINE estimator based on task loss, accordingly obtaining a tighter lower bound. As shown in the middle part of Figure 1, by maximizing $\sum_t I(Z; Y_t)$, task-specific alignment enables Z to cover the red shaded area that is lost under vanilla MTL objective, thereby making Z more sufficient for all tasks.

Besides, we design a self-alignment objective to mitigate the negative effect of task-irrelevant features. Maximizing $I(Z; Y_t)$ may introduces excessive task-irrelevant information in the shared representations Z. Under the objective $\max I(Z_t; Y_t)$ of each task t, the task-specific representations Z_t still contain some information in Z that is irrelevant to Y_t, as shown in the gray-shaded region in the middle part of Figure 1. The self-alignment objective minimizes the conditional mutual information between Z and input perturbations δ given each target task Y_t. In the implementation, we introduce an adversarial estimator to approximately estimate the minimization $I(Z; \delta | Y_t)$. As shown in the right part of Figure 1, self-alignment objective can adversarially mitigate negative task-irrelevant features in the representations.

We experiment on two multi-task language benchmarks, i.e., TweetEval and AffectEval. Results show that MTRA outperforms 13 representative MTL methods across different PLMs under the same settings. For example, with RoBERTa backbone, MTRA improves the average performance (Avg.) by +3.8% and increases average relative improvement (Δp) by +7.1% on TweetEval, and yields improvements in Avg. by +3.6% and an increase in Δp by +7.2% on AffectEval, compared to the EW baseline. Compared to single-task learning baselines, MTRA achieves superior results on most tasks with comparable scale of model parameters. Extensive experiments show that MTRA achieves advantages under data-scarce and label-noisy conditions. Furthermore, our learned shared representations exhibit higher task informativeness and superior alignment properties.

Our contributions are summarized as follows: 1) We reframe task interference in MTL as a representation alignment problem, and introduce a new idea of aligning multi-task representations to address it. It is a significant departure from existing works that focus on task balancing or probabilistic modeling. 2) We propose an MTRA framework to achieve dual representation alignments on the shared representations from a mutual information perspective. It ensures task-specific alignment by mutual information maximization with the MINE estimator, and self-alignment by conditional infor-

mation minimization with the adversarial estimator. 3) Experiments on two language multi-task benchmarks show our MTRA outperforms 13 representative MTL methods under the same settings, particularly under label-noisy and data-constrained conditions. Further analysis show that our shared representations exhibit sufficient task informativeness and superior alignment properties.[1]

2 Preliminary

Scope of the Study. This paper aligns with the multi-task optimization paradigm that typically adopts a hard parameter-sharing pattern (Caruana, 1993), where multiple lightweight, task-specific heads are built on top of a heavyweight, task-agnostic backbone. Another orthogonal line of MTL research explores soft parameter-sharing by designing more flexible network architectures. And the scope of our study is complementary to the architecture design line, as we focus on how to align multi-task representations in MTL.

Notations. Define T tasks and dataset $\mathcal{D}_t$ for task t. An MTL model usually contains two parts of parameters: task-sharing encoder parameters θ and task-specific decoder parameters $\{\phi_t\}_{t=1}^{T}$, where θ denotes parameters in a feature extractor shared by all tasks, ϕ_t refers to parameters in the task-specific output module for task t, and T is the number of tasks. Let $\ell_t(\mathcal{D}_t; \theta, \phi_t)$ be the average loss on $\mathcal{D}_t$ for task t. $\{\lambda_t\}_{t=1}^{|T|}$ is a set of task-specific loss weights with a constraint that $\lambda_t^l \geq 0$.

MTL Baseline. Since there are multiple losses in MTL, they usually are aggregated as a single one via loss 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 for loss weighting is to assign the same weight to all tasks during training, i.e., $\lambda_t = \frac{1}{|T|}$ for all tasks in every iteration. It is a common baseline in MTL named Equal Weighting (EW) in this paper. MT-DNN (Liu et al., 2019b) is a version of EW baseline with the BERT backbone.

3 Methodology

We propose a multi-task representation alignment (MTRA) framework to achieve dual alignments on the shared representations in MTL. The goal is to ensure task-specific alignment by mutual information maximization with the MINE estimator, and self-alignment by conditional information minimization with the adversarial estimator.

3.1 Task-specific Alignment via Information Maximization

We design a task-specific alignment objective to align the shared representations and task-specific representations with the expected targets of all tasks. To ensure the shared representations $z = p_\theta(x)$ to encode sufficient semantic information for all tasks, we maximize the sum of mutual information terms between the shared representations Z and each task Y_t, i.e., $\max \sum_{t=1}^{T} I(Z; Y_t)$. Besides, following the vanilla MTL objective, we also maximize the mutual information between the task-specific representations Z_t and Y_t to ensure the task-specific representations Z_t be be sufficient for the target task. Thus, the objective of task-specific alignment can be,

$$\max \sum_{t=1}^{T} I(Z; Y_t) + \beta I(Z_t; Y_t), \tag{1}$$

where β is a trade-off hyperparameter.

In the implementation, since direct computation of $I(Z; Y_t)$ is intractable, we adopt the Mutual Information Neural Estimator (MINE) (Belghazi et al., 2018) to maximize $I(Z; Y_t)$, which provides the following variational lower bound.

We maximize $I(Z; Y_t)$ between continuous shared representations and discrete task labels, whereas traditional MINE applications typically maximize $I(X; Z)$ in self-supervised settings. This shift is essential for learning sufficient shared representations in the multi-task regime. Specifically, we use a lightweight neural network to approximate the lower bound of $I(Z; Y_t)$, i.e.,

$$\begin{aligned} I(Z; Y_t) &\geq I_\Omega(Z; Y_t) \\ &= \sup_{\omega \in \Omega} \mathbb{E}_{\mathbb{P}_{ZY_t}}[V_\omega(y_t|z)] - \log(\mathbb{E}_{\mathbb{P}_Z \otimes \mathbb{P}_{Y_t}}[e^{V_\omega(y_t|\tilde{z})}]), \end{aligned} \tag{2}$$

where $\mathbb{P}_{ZY_t}$ is the joint distribution and $\mathbb{P}_Z \otimes \mathbb{P}_{Y_t}$ is the product of the marginals. Following Hu et al. (2024), the statistics network V_ω is parametrized by an MLP (i.e., a fully-connected neural network with two hidden layers) with the parameter $\omega \in \Omega$, shared by all tasks. The expectations of $I_\Omega(Z; Y_t)$ are estimated by shuffling the samples from the joint distribution along the batch axis. Maximizing this bound encourages a semantic mapping between the representation space and the joint task label space, which enhances the expressiveness and transferability of Z across tasks.

[1]The source code is available at https://github.com/zerohd4869/MTRA.

Then, we apply logarithmic operation on task-level loss $\ell_{t,\text{TA}}$ (i.e., $\ell_{t,\text{Task}}+\beta\ell_{t,\text{MI}}$) to rescale each loss. The task-specific alignment loss is

$$\mathcal{L}_{\text{TA}}(\theta,\{\phi_t\}_{t=1}^T,\omega)=\sum_{t=1}^T\log(\ell_{t,\text{Task}}(\theta,\phi_t)+\beta\ell_{t,\text{MI}}(\theta,\omega)), \tag{3}$$

where

$$\begin{aligned}\ell_{t,\text{MI}}(\theta,\omega)=-\mathbb{E}_{z\sim p_\theta(z|x)}[\mathbb{E}_{\mathbb{P}_{ZY_t}}[V_\omega(y_t|z)]\\ -\log(\mathbb{E}_{\mathbb{P}_Z\otimes\mathbb{P}_{Y_t}}[e^{V_\omega(y_t|\tilde{z})}])].\end{aligned} \tag{4}$$

And $\ell_{t,\text{Task}}$ refers to the task loss, i.e., cross-entropy (CE) loss for classification tasks and mean squared error (MSE) loss for regression tasks. Different from traditional methods that implicitly maximize $I(Z;Y_t)$ via task loss (e.g., cross-entropy loss can be viewed as $\max I(Z_t;Y_t)$, that is a lower bound of $I(Z;Y_t)$), our task-specific alignment explicitly estimates $I(Z;Y_t)$ using MINE estimator based on task loss, accordingly obtaining a tighter lower bound of $I(Z;Y_t)$. As a result, the shared representations Z learned by our method can be more sufficient for all tasks.

3.2 Self-Alignment via Conditional Information Minimization

While maximizing $I(Z;Y_t)$ improves semantic expressiveness, it may introduce excessive task-irrelevant features in the shared representations. Under the objective $\max I(Z_t;Y_t)$ of each task t, the task-specific representations Z_t still contain some information in Z that is irrelevant to Y_t. To avoid this, we design a self-alignment to eliminate task-irrelevant features. It minimizes the conditional mutual information between the shared representations Z and local perturbations δ given the target task Y_t, i.e.,

$$\min\sum_{t=1}^T I(Z;\delta|Y_t). \tag{5}$$

For implementation, inspired by Terzi et al. (2021); Hu et al. (2024), we use gradient-based adversarial training (i.e., FGM (Goodfellow et al., 2015; Miyato et al., 2017)) to approximately estimate the conditional information minimization. Formally, let $\{(x^i,y^i)\}_{i=1}^{|\mathcal{B}|}$ denote a batch input sampled from the dataset $\mathcal{D}$ and $p(y|x;\Theta)$ as a model with the parameters $\Theta=\{\theta,\phi_t,\omega\}$. The worst-case perturbation δ under the given Y_t can be computed by the back-propagation in the network:

$$\max_{\|\delta\|_q\le\epsilon} D_{KL}(p_{\theta_\delta}(z|x);p_\theta(z|x)), \tag{6}$$

with a Lagrange constraint $\ell_{t,\text{TA}}(x,y;\Theta_\delta)$ where $\Theta_\delta=\{\theta_\delta,\phi_t,\omega\}$, and an L_2 norm constraint. The KL divergence D_{KL} quantifies the sensitivity of the representation Z to the perturbation δ with a fixed radius ϵ. Maximizing it enforces $p_{\theta_\delta}(z|x)\approx p_\theta(z|x)$, which represents local invariance or smoothness.

At each step of training, we identify the adversarial perturbations δ against the current network with the parameter $\hat{\theta}$, and put the perturbations on the weights of the embedding layer of the encoder network. With a linear approximation (Goodfellow et al., 2015), an L_2 norm-ball constraint, a fixed radius ϵ for δ, and the original objective $\ell_{t,\text{TA}}$ with parameters $\Theta=\{\theta,\phi_t,\omega\}$, the formulation of self-alignment objective $\ell_{t,\text{SA}}$ is,

$$\begin{aligned}&\min_\Theta\mathbb{E}_{(x,y)\sim D}\max_{\|\delta\|_2\le\epsilon}\ell_{t,\text{TA}}(x,y;\Theta_\delta),\\ &\text{where }\delta=-\epsilon g/\|g\|_2,\ g=\nabla_x\ell_{t,\text{TA}}(x,y;\hat{\Theta}).\end{aligned} \tag{7}$$

The conditional information minimization loss $\ell_{t,\text{SA}}$ can be viewed as an adversarial regularizer. This ensures local stability of Z under small input variations, promoting the representation smoothness. It implicitly minimizes the conditional mutual information $I(Z;\delta|Y_t)$, promoting self-alignment by enforcing local linearity around X.

3.3 Overall Objective and Training

Combining information maximization in Eq.(1) and conditional information minimization in Eq.(5), the total optimization principle of MTRA can be,

$$\max\sum_{t=1}^T I(Z;Y_t)+\beta I(Z_t;Y_t)-I(Z;\delta|Y_t). \tag{8}$$

Under this principle, the overall objective of MTL is computed as,

$$\begin{aligned}\mathcal{L}_{\text{MTRA}}(\theta,\{\phi_t\}_{t=1}^T,\omega)&=\mathcal{L}_{\text{TA}}+\mathcal{L}_{\text{SA}}\\ &=\sum_{t=1}^T\log(\ell_{t,\text{TA}}+\ell_{t,\text{SA}}).\end{aligned} \tag{9}$$

The training procedure of MTRA is summarized in Algorithm 1. First, task-specific alignment maximizes the mutual information between the shared representations and all task targets. Second, self-alignment minimizes the conditional mutual information between the shared representations and input perturbations given each target task. Totally, MTRA ensures that the learned representations contain sufficient task-relevant features while mitigating the negative effects of task-irrelevant features, thereby alleviating the task interference issue.

Algorithm 1 MTRA

Input: Dataset $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T}$ with T tasks; hyperparameter β and ϵ; learning rate η; pretrained encoder θ_0; network parameters $\Theta = \{\theta, \{\phi_t\}_{t=1}^{T}, \omega\}$ including a task-shared encoder θ, task-specific decoders $\{\phi_t\}_{t=1}^{T}$, and an MINE net ω.
Initialize: $\theta \leftarrow \theta_0$
repeat
  Randomly split dataset $\mathcal{D}$ into multiple task batches $\{\mathcal{B}_1^k\}_{k=1}^{K_1}, \{\mathcal{B}_2^k\}_{k=1}^{K_2}, ..., \{\mathcal{B}_t^k\}_{k=1}^{K_t}, ...$
  for each batch $\mathcal{B}_t^k = \{(x_t^{ik}, y_t^{ik})_{i=1}^{|\mathcal{B}_t^k|}\}$ **do**
    Compute the shared representations : $z \leftarrow p_\theta(x)$
    // Task-specific alignment
    $\ell_{t,\text{MI}}(\theta, \omega) = -\mathbb{E}_{z\sim p_\theta(z|x)}[\mathbb{E}_{\mathbb{P}_{ZY_t}}[V_\omega(y_t|z)] - \log(\mathbb{E}_{\mathbb{P}_Z \otimes \mathbb{P}_{Y_t}}[e^{V_\omega(y_t|\tilde{z})}])]$.
    $\ell_{t,\text{TA}} = \ell_{t,\text{Task}} + \beta \ell_{t,\text{MI}}$
    // Self-alignment
    $\delta \leftarrow -\epsilon \nabla_x \ell_{t,\text{TA}} / \|\nabla_x \ell_{t,\text{TA}}\|_2$, and $z_\delta \leftarrow p_{\theta_\delta}(x)$
    $\ell_{t,\text{SA}} = \sup_{\|\delta\|_2 \le \epsilon} \ell_{t,\text{TA}}(x, y; \Theta_\delta)$,
    // Update the network parameters
    $\{\theta, \phi_t, \omega\} \leftarrow \{\theta, \phi_t, \omega\} - \eta \nabla \log(\ell_{t,\text{TA}} + \ell_{t,\text{SA}})$
  end for
until convergence
Output: $\Theta^* = \{\theta^*, \{\phi_t^*\}_{t=1}^{T}, \omega^*\}$

The total objective in Eq.(9) is only used during the training phase (to update learnable parameters). During the validation phase, the loss function only includes the task-specific alignment term $\mathcal{L}_{\text{TA}}$ in Eq.(3), eliminating the need to compute gradients in the self-alignment process.

4 Experiments

4.1 Experimental Setups

Datasets and Tasks Following previous works (Hu et al., 2025a,b), we experiment on two public multi-task benchmarks, which consider task similarity (whether tasks differ or consist of multiple sub-tasks) and multi-task type (heterogeneous or isomorphic). **TweetEval** (Barbieri et al., 2020) comprises 6 isomorphic classification tasks about tweet analysis on social media including 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** comprises 2 classification and 2 regression tasks in a heterogeneous multi-task setting including 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 details of datasets and tasks.

Comparison Methods We compare with the following 13 representative MTL methods, i.e., 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 (RLW) (Lin et al., 2022), MT-VIB (Qian et al., 2020), VMTL (Shen et al., 2021), Hierarchical MTL (de Freitas et al., 2022), and InfoMTL (Hu et al., 2025b). We use pre-trained language models BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019c) as the model backbone. We also compare with large language models (i.e., GPT-3.5, Llama-3, and Qwen-2.5), and single task learning baselines. See Appendix B.2 for details.

Evaluation Metrics We compute the same evaluation metrics as used in original tasks. See Appendix B.3 for details of the metrics. We use t-test (Kim, 2015) to verify the statistical significance of differences between results of MTRA and baseline.

Implementation Details We experiment 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. Following Liu et al. (2019b); Hu et al. (2025b), we optimize the network parameters using the Adamax optimizer (Kingma and Ba, 2015) and clip the gradient norm to 1 to prevent exploding gradients. See Appendix B.4 for details.

4.2 Main Results

Overall Results for MTL The overall results for MTL are summarized in Table 1. MTRA consistently obtains the best average performance over comparison methods on both benchmarks with different backbone models. Specifically, compared to EW baseline, MTRA with BERT/RoBERTa improves Avg. by +2.5%/+3.8% and increases Δp by +4.4%/+7.1% on TweetEval, and MTRA with BERT/RoBERTa enhances Avg. by +4.9%/+3.6% and increases Δp by +16.0%/+7.2% on AffectEval. We also evaluate on different pair-wise task combinations in Appendix C.1.

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±0.57	0.00	66.17±0.43	0.00	52.93±2.02	0.00	57.64±2.12	0.00
SI	65.67±0.66	+0.06	67.16±1.08	+1.75	53.49±1.89	+1.80	57.94±2.02	+0.61
TW	65.68±0.54	+0.11	67.08±1.17	+1.55	53.27±2.12	+0.82	57.70±1.63	+0.09
UW	66.97±0.51	+2.22	67.11±3.47	+1.92	53.79±1.85	+1.81	59.69±1.10	+4.05
GLS	66.05±1.49	+0.60	67.32±0.38	+1.67	54.56±0.36	+9.82	57.66±1.65	-0.23
DWA	65.56±0.57	-0.09	66.94±1.13	+1.35	52.88±1.88	-0.25	57.36±2.53	-0.51
PCGrad	65.45±0.33	-0.50	67.42±0.30	+1.96	51.62±0.51	-3.09	56.27±2.16	-2.73
IMTL-L	66.18±1.45	+0.86	66.54±1.50	+0.67	53.89±0.42	+3.41	57.73±1.20	+0.05
RLW	66.76±1.42	+1.86	67.07±0.73	+1.63	51.38±1.42	-3.03	55.61±2.32	-4.26
MT-VIB	65.80±0.23	+0.66	67.14±0.87	+2.00	50.13±0.71	-5.09	57.68±1.56	+0.36
VMTL	65.80±1.59	+0.65	67.05±1.06	+1.81	50.02±0.76	-5.01	57.52±0.48	+0.20
Hierarchical MTL	66.42±0.10	+1.76	66.84±1.68	+1.60	50.55±0.65	-4.19	55.18±0.58	-4.74
InfoMTL	67.51±0.60	+3.70	68.50±0.58	+3.97	54.75±2.21	+8.13	57.91±1.40	+0.57
MTRA	**68.16***±0.58	**+4.37**	**69.96***±1.62	**+7.06**	**57.83***±0.96	**+15.96**	**61.20***±0.65	**+7.18**

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±0.56	44.08±5.26	65.32±1.84	79.04±1.43	70.64±1.71	63.59±2.43	66.17±0.43	0.00
SI	75.81±1.05	46.19±6.01	66.17±5.81	78.58±2.00	71.00±1.80	65.24±2.31	67.16±1.08	+1.75
UW	74.76±3.08	48.49±3.21	65.41±7.01	79.49±1.48	71.56±0.74	62.96±6.84	67.11±3.47	+1.92
MT-VIB	74.74±0.38	48.06±4.79	66.09±3.38	78.17±1.39	70.95±0.99	64.83±1.56	67.14±0.87	+2.00
InfoMTL	76.90±0.62	48.44±2.15	**68.94**±1.86	79.78±0.86	71.92±0.36	65.02±1.81	68.50±0.58	+3.97
MTRA	**76.97***±1.54	**55.48***±4.24	68.11*±2.59	**80.10***±0.10	**72.05***±0.81	**67.02***±2.61	**69.96***±1.62	**+7.06**

(a) Fine-grained results (%) on TweetEval.

Methods	GoEmotions	EmotionEval	Emobank			EI-Reg		**Avg.**	**Δp** $\uparrow$
	M-F1	M-F1	V	A	D	Pear	Spear		
EW (baseline)	47.13±0.33	77.97±0.63	75.62±0.79	49.44±4.70	36.47±4.02	51.01±4.62	52.23±4.68	57.64±2.12	0.00
SI	47.08±0.72	78.22±0.49	75.61±1.39	50.35±5.02	37.26±4.78	51.55±3.99	52.60±3.82	57.94±2.02	+0.61
UW	48.54±0.55	78.55±1.14	76.81±0.28	53.26±0.44	38.60±3.32	54.94±3.14	55.93±3.00	59.69±1.10	+4.05
MT-VIB	46.92±0.29	76.66±2.31	75.61±1.96	51.60±1.01	37.50±5.59	51.80±1.39	52.64±2.19	57.68±1.56	+0.36
InfoMTL	**49.21**±0.71	78.17±0.59	73.81±4.77	48.21±4.46	35.02±3.33	51.72±2.83	52.09±2.96	57.91±1.40	+0.57
MTRA	48.48*±0.87	**78.90***±1.39	**79.59***±1.21	**55.40***±0.65	**44.45***±0.93	**57.38***±1.39	**57.81***±1.05	**61.20***±0.65	**+7.18**

(b) Fine-grained results (%) on AffectEval.

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

Fine-grained Results Table 2 presents the fine-grained results of MTRA and several representative MTL methods, and InfoMTL on each task of TweetEval and AffectEval. MTRA consistently outperforms EW across all tasks and achieves the best performance on most tasks.

Comparison with STL and LLM Table 3 compares MTRA with single-task learning (STL) and large language models (LLMs). For STL, each task is trained using a separate model with the same backbone. For LLMs, we compare with GPT-3.5, Llama-3, and Qwen-2.5 evaluated in zero-shot or in-context settings. Compared to STL baselines, MTRA yields better results on most tasks with comparable scale of parameters. Besides, MTRA outperforms the three LLMs on all tasks significantly. It shows that our framework with small LMs can achieve competitive results without relying on massive model scale.

4.3 Ablation Study

We evaluate key components in MTRA by removing information maximization for task-specific alignment (w/o TA), removing conditional information minimization for self-alignment (w/o SA), and ablating the logarithmic operation (w/o SI). For w/o TA, we remove $\ell_{t,\text{MI}}$ and preserve $\ell_{t,\text{Task}}$ in Eq(3) for task prediction. Table 4 shows the ablation study results on TweetEval and AffectEval. The full MTRA consistently obtains the best performance in terms of all metrics on both benchmarks. The results validate that jointly achieving task-specific alignment and self-alignment is essential for effective multi-task representation learning.

Methods	# Param	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**
Llama-3†	1.2B	58.07	-	35.80	-	58.78	-	-
Qwen-2.5†	1.5B	63.61	-	65.80	-	52.39	-	-
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
MTRA	**110M**	**76.97**	**55.48**	**68.11**	**80.10**	72.05	**67.02**	**69.96**

Table 3: Comparison results (%) with different learning paradigms on TweetEval. We experiment with RoBERTa backbone for all methods except for Llama-3, Qwen-2.5, and GPT-3.5. † and ‡ refer to results obtained from Zhang et al. (2025) and Hu et al. (2025b), respectively. 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.	**Δp ↑**	**Avg.**	**Δp ↑**	**Avg.**	**Δp ↑**	**Avg.**	**Δp ↑**
MTRA	**68.16**±0.58	**+4.37**	**69.96**±1.62	**+7.06**	**57.83**±0.96	**+15.96**	**61.20**±0.65	**+7.18**
w/o TA	67.57±1.16	+3.05	67.10±1.86	+1.48	54.81±1.75	+5.27	59.50±0.40	+3.57
w/o SA	66.93±0.91	+2.09	67.22±0.96	+2.09	57.01±1.09	+12.89	60.60±0.50	+6.06
w/o TA & SA	65.67±0.66	+0.06	67.16±0.96	+1.75	53.49±1.89	+1.80	57.94±2.02	+0.61
w/o TA & SA & SI	65.62±0.57	0.00	66.17±0.43	0.00	52.93±2.02	0.00	57.64±2.12	0.00

Table 4: Ablation study results (%). We report their fine-grained results on each task in Appendix C.2.

Methods	Training		Validation	
	TA Loss	SA Loss	TA Loss	SA Loss
EW	0.7048	0.3358	0.6243	0.3353
MTRA	**0.3210**	**0.2848**	**0.6126**	**0.2891**
w/o SA	0.3710	0.3068	0.6198	0.3114
w/o TA & SA	0.3733	0.3682	0.6313	0.3689

Table 5: Results of alignment property. *TA Loss* and *SA Loss* are task-specific alignment and self-alignment losses where lower values indicate better alignment.

Methods	Training $I(Z;Y_t)$	Validation $I(Z;Y_t)$
MTRA (β=1)	**0.0583**	**0.0433**
w/o SA	0.0351	0.0261
MTRA (β=0.5)	0.0217	0.0180
MTRA (β=2)	0.0404	0.0329

Table 6: Results of $I(Z;Y_t)$. Larger $I(Z;Y_t)$ refers to the learned shared representations contain more task-relevant information.

Besides, the ablation w/o TA implicitly maximizes $I(Z;Y_t)$ via $\ell_{t,\text{Task}}$ (e.g., cross-entropy loss for classification), which is a lower bound of $I(Z;Y_t)$. Our TA explicitly estimates $I(Z;Y_t)$ using $\ell_{t,\text{MI}}$ based on $\ell_{t,\text{Task}}$, accordingly obtaining a tighter lower bound of $I(Z;Y_t)$. The comparison results show that for maximizing $I(Z;Y_t)$, selecting the MINE estimator is a superior choice compared to the traditional cross-entropy loss.

4.4 Representation Quality Evaluation

MTRA aims to achieve task-specific alignment and self-alignment on the shared representations. To further explain how does MTRA work well, we perform quantitative representation evaluation by analyzing alignment property and the task informativeness of the shared representations.

Alignment Property Analysis To analyze the alignment properties of the shared representations, we evaluate the task-specific loss and the self-alignment loss obtained by different objectives. The task-specific loss (e.g., cross-entropy loss for classification) is used to estimate the mutual information between the task-specific output representations Z_t and the target task Y_t, which serves as a lower bound of $I(Z;Y_t)$. A smaller task-specific loss implies a tighter lower bound on $I(Z;Y_t)$, indicating better sufficiency of both the shared representations and the task-specific representations in capturing the information relevant to the target task. Self-alignment loss (i.e., alignment in Wang and Isola (2020)) is calculated by the expected distance between the positive paired embeddings from the same input sample. For a sample X with the shared representations Z, its positive key is the augmented representations Z' obtained by dropout in the shared encoder. A smaller self-alignment loss indicates greater invariance of the learned representations to task-irrelevant perturbations, (geometric alignment). As shown in Table 5, MTRA achieves lower task-specific loss and self-alignment loss, suggesting that MTRA promotes better alignment

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	+19.17
MT-VIB	20%	60.00	-4.18	44.35	+4.30
InfoMTL	20%	**64.83**	**+4.07**	**49.44**	**+24.07**
MTRA	20%	64.59	+3.67	48.86	+17.77
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
MT-VIB	60%	66.31	+0.04	52.85	-3.94
InfoMTL	60%	66.71	+0.50	54.13	-1.39
MTRA	60%	**67.14**	**+1.27**	**58.80**	**+8.31**
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
MT-VIB	80%	65.34	-1.57	54.80	-3.39
InfoMTL	80%	67.88	+2.53	56.96	+1.09
MTRA	80%	**68.46**	**+3.91**	**59.81**	**+6.57**

Table 7: Results (%) against different training data size. RoBERTa is the default backbone model. See Appendix C.3 for fine-grained results.

Methods	Label noise	**Avg.**	**Δp** $\uparrow$
EW	20%	68.03	0.00
SI	20%	66.86	-2.07
UW	20%	67.62	-0.72
MT-VIB	20%	66.95	-1.56
InfoMTL	20%	67.82	-0.31
MTRA	20%	**69.67**	**+2.55**
EW	40%	67.05	0.00
SI	40%	64.75	-3.20
UW	40%	67.17	+0.55
MT-VIB	40%	67.57	+1.29
InfoMTL	40%	67.47	+1.09
MTRA	40%	**69.24**	**+4.01**
EW	60%	66.70	0.00
SI	60%	65.22	-2.39
UW	60%	64.36	-4.06
MT-VIB	60%	65.65	-1.83
InfoMTL	60%	65.77	-1.92
MTRA	60%	**68.87**	**+3.17**

Table 8: Results (%) against different ratios of label noises on TweetEval. RoBERTa is the default backbone model. See Appendix C.4 for their fine-grained results.

properties on the shared representations in MTL.

Task Informativeness Analysis We evaluate the effect of task-specific alignment module on the task informativeness of the learned shared representations by comparing MTRA and its variants under different values of β, as well as the ablation w/o self-alignment (SA). Different values of β refer to the trade-off between $\mathcal{L}_{\text{Task}}$ and $\mathcal{L}_{\text{MI}}$. Specifically, we compute $I(Z;Y_t)$ using MINE estimator. Higher $I(Z;Y_t)$ refers to more sufficient task informativeness. Table 6 shows the mutual information between the shared representations Z and the task Y_t on training and validation sets of TweetEval. MTRA with $\beta = 1$ achieves the highest $I(Z;Y_t)$, indicating an optimal balance between task supervision and task-specific alignment. When $\beta = 0.5$, $I(Z;Y_t)$ drops markedly due to insufficient emphasis on task-specific alignment. MTRA with $\beta = 2$ still underperforms the balanced setting of $\beta = 1$. These findings highlight the importance of properly tuning the task-specific alignment objective to ensure both informativeness and generalization. Additionally, MTRA achieves higher mutual information values across tasks compared to the ablation w/o SA. Results show that self-alignment objective is beneficial to mitigate the negative effects of task-irrelevant features on the shared representations.

4.5 Further Analysis

Evaluation under Data-constrained Conditions The evaluation aims to assess the effectiveness in real-world scenarios where the partiality problem is more severe. We evaluate MTRA and 5 representative MTL methods when training with limited data (see Appendix C.3 for details of experimental setups). As shown in Table 7, MTRA achieves superior average performance against different ratios of the training set. This suggests that MTRA can learn sufficient representations, improving the efficiency of utilizing limited data.

Evaluation under Label-noisy Conditions We evaluate the robustness against noisy labels during training by adjusting ratios of training set (see Appendix C.4 for details of experimental setups). As shown in Table 8, MTRA consistently achieves better performance against different ratios of label noises, showing that MTRA performs more robustly on noisy label. Besides, compared to EW, MTRA improves Δp by +2.6%, +4.0%, and +3.2% with noise ratio of 20%, 40%, and 60% on TweetEval. The results prove that MTRA can better control and utilize randomness and uncertainty of data.

Additionally, we provide sensitivity analysis of hyperparameters, and efficiency analysis in Appendix C.5, and C.6, respectively.

5 Related Work

Multi-task learning (MTL) enables joint learning over multiple tasks and making predictions based on shared representations (Caruana, 1993; Ruder et al., 2019). Unlike single-task learning, MTL can improve training efficiency and reduce computational costs. Following Hu et al. (2025b), existing

works can be roughly categorized into multi-task optimization and multi-task network architectures.

5.1 Multi-task Optimization

The optimization of MTL aims to optimize learning process across multiple 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. Different from the soft parameter sharing that allowing for controlled parameter sharing among tasks, the hard pattern has the advantages of lower inference cost and simpler requirements on the network architecture. Existing works on multi-task optimization can be broadly categorized into two groups: task-balanced methods and probabilistic methods.

Task-balanced methods adjust loss weights or gradients to mitigate conflicts between tasks. Loss-based methods (Kendall et al., 2018; Chennupati et al., 2019; Liu et al., 2019a, 2021b; Lin et al., 2022) adjust loss scales or magnitudes to align the task losses. Specifically, loss-based methods require computing all task-specific losses per batch to update weighting strategies, often demanding large batch sizes or task-balanced sampling. Gradient-based methods (Sener and Koltun, 2018; Chen et al., 2018; Yu et al., 2020; Liu et al., 2023) balance the gradient magnitudes and avoid conflicts in gradient directions to align the task gradients. Compared to loss-based methods, gradient-based methods need to operate gradients across tasks, introducing higher computations and memory costs during training.

Probabilistic methods model task relatedness or reduce redundancy to facilitate the cooperative learning across tasks. Some works explore task relatedness by designing the shared priors under the Bayesian framework (Yu et al., 2005; Titsias and Lázaro-Gredilla, 2011; Archambeau et al., 2011; Bakker and Heskes, 2003) or sharing the covariance structure of parameters (III, 2009). Additionally, some works (Vera et al., 2017; Qian et al., 2020; de Freitas et al., 2022; Hu et al., 2025a,b) compress task-irrelevant redundant information or cross-task redundant interference under the information bottleneck principle (Tishby et al., 1999; Tishby and Zaslavsky, 2015). Specifically, Hu et al. (2025b) propose an information-theoretic framework with variational implementation to learn noise-invariant sufficient representations for all tasks. Hu et al. (2025a) introduce a representation balancing framework to ensures impartial multi-task learning by harmonizing representation spaces across tasks. However, shared representations learned by these information-theoretic MTL methods may still suffer from task-specific insufficiency, leading to competition across tasks. In contrast, we depart from the information bottleneck principle and formulate MTL as information maximization and conditional information minimization under shared representations in MTL, explicitly targeting task sufficiency and task interference mitigation. Besides, existing information-theoretic MTL methods rely on probabilistic modeling and variational inference with sampling. Our MTRA framework avoids these high-cost procedures. We instead employ a lightweight MINE estimator for task-specific alignment and an adversarial estimator for self-alignment, achieving a simpler and more scalable design.

5.2 Multi-task Network Architectures

Another orthogonal line of MTL research explores soft parameter-sharing by designing more flexible network architectures (Misra et al., 2016; Hashimoto et al., 2017; Ruder et al., 2019; Liu et al., 2019a,b). It enables controlled parameter sharing across tasks but often leads to increased inference costs. The focus of our research is complementary to this line of work, as we focus on align multi-task representations to optimize the learning process across multiple tasks that is agnostic to the underlying architecture.

6 Conclusion

To mitigate task interference in MTL, we propose a new MTRA framework to achieve dual alignments on the shared representations from a mutual information perspective. It ensures task-specific alignment by mutual information maximization, and self-alignment by conditional information minimization. Experiments on two multi-task language understanding benchmarks demonstrate that MTRA consistently outperforms comparison MTL baselines, especially in label-noisy and data-constrained scenarios. Further analysis confirms that the shared representations learned by MTRA exhibit sufficient task informativeness and superior alignment properties.

Limitations

This work investigates to mitigate the task interference problem in multi-task learning (MTL) via a new multi-task representation alignment framework, showing the effectiveness in label-noisy and data-constrained settings. Nevertheless, the work is subject to several limitations. First, the experiments focus only on natural language understanding tasks, i.e., classification and regression. The generalizability of MTRA to generative tasks remains unexplored. Besides, the scalability of the framework to larger model architectures such as large language models (LLMs), has not been investigated in this paper. Moreover, the proposed MTRA is a representation learning framework and should be combined with fairness-aware evaluation and mitigation strategies when applied in socially sensitive domains.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. U24A20335), the Fundamental Research Funds for the Central Universities (No. CUC25SG002), the Postdoctoral Fellowship Program of China Postdoctoral Science Foundation (No. GZC20232969), and the Public Computing Cloud, CUC. The authors thank the anonymous reviewers and the meta-reviewer for their helpful comments.

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.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. 2018. Mutual information neural estimation. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 530–539.

Sven Buechel and Udo Hahn. 2017. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *EACL*, pages 578–585.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, pages 41–48.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 793–802.

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 *CVPR Workshops*, pages 1200–1210.

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.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR (Poster)*.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *EMNLP*, pages 1923–1933.

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.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

Dou Hu, Lingwei Wei, Wei Zhou, and Songlin Hu. 2024. Representation learning with conditional information flow maximization. In *ACL*, pages 14088–14103.

Dou Hu, Lingwei Wei, Wei Zhou, and Songlin Hu. 2025a. Impartial multi-task representation learning via variance-invariant probabilistic decoding. In *ACL*, pages 19883–19897.

Dou Hu, Lingwei Wei, Wei Zhou, and Songlin Hu. 2025b. An information-theoretic multi-task representation learning framework for natural language understanding. In *AAAI*, pages 17276–17286.

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

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. 2023. How does information bottleneck help deep learning? In *ICML*, pages 16049–16096.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491.

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)*.

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, Yihao Feng, Peter Stone, and Qiang Liu. 2023. FAMO: fast adaptive multitask optimization. In *NeurIPS*.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. 2021a. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, 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 *ICLR*.

Shikun Liu, Edward Johns, and Andrew J. Davison. 2019a. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. In *ACL*, pages 4487–4496.

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 *CVPR*, pages 1851–1860.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *ICLR (Poster)*.

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.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent multi-task architecture learning. In *AAAI*, pages 4822–4829.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *NeurIPS*, 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.

Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810.

Trevor Standley, Amir Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2020. Which tasks should be learned together in multi-task learning? In *ICML*, pages 9120–9132.

Matteo Terzi, Alessandro Achille, Marco Maggipinto, and Gian Antonio Susto. 2021. Adversarial training reduces information and improves transferability. In *AAAI*, pages 2674–2682. AAAI Press.

Naftali Tishby, Fernando C Pereira, and William Bialek. 1999. The information bottleneck method. In *Proc. of the 37th Allerton Conference on Communication and Computation*, pages 368–377.

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.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pages 9929–9939.

Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. 2022. Do current multi-task optimization methods in deep learning even help? In *NeurIPS*.

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

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *NeurIPS*.

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.

Yice Zhang, Guangyu Xie, Jingjie Lin, Jianzhu Bao, Qianlong Wang, Xi Zeng, and Ruifeng Xu. 2025. Targeted distillation for sentiment analysis. In *EMNLP*, pages 22158–22181.

Appendix Overview

In this appendix, we provide: (i) theoretical explanations, (ii) detailed experimental setups, and (iii) supplementary results.

A Theoretical Explanations

From a theoretical perspective, MINE provides a useful lower bound for mutual information maximization (Belghazi et al., 2018; Hjelm et al., 2019). The connection between adversarial training and conditional mutual information minimization has been theoretically established (Terzi et al., 2021) and empirically validated (Hu et al., 2024). These established results provide an important theoretical foundation for the validity of our MTRA.

Building upon these foundations, our contribution is not merely to introduce these techniques into the multi-task learning (MTL) paradigm to explicitly address the task interference problem. In addition, we redesign and integrate these components to accommodate the shared-representation setting of MTL. In particular, our MI formulation is structurally different from prior uses of MINE. We maximize $I(Z;Y_t)$ between continuous shared representations Z and discrete task labels Y_t, whereas traditional MINE applications typically maximize $I(X;Z)$ in self-supervised representation learning settings. This shift is crucial for learning task-sufficient shared representations in the multi-task regime, where the objective is not generic representation learning but preserving task-relevant information under representation sharing.

Building upon prior theoretical results, we further analyze MINE-based task-specific alignment in the multi-task setting. Taking a classification task as an example, the cross-entropy loss $\mathcal{L}_{\text{CE}}$ satisfies:

$$\begin{aligned}\mathcal{L}_{\text{CE}} &= H(Y_t|Z_t) + \mathbb{E}_{Z_t} KL(p(\cdot|Z_t)) \\ &\geq H(Y_t|Z_t),\end{aligned} \tag{10}$$

which implies:

$$\begin{aligned}I(Z_t;Y_t) &= H(Y_t) - H(Y_t|Z_t) \\ &\geq H(Y_t) - \mathcal{L}_{\text{CE}}.\end{aligned} \tag{11}$$

Thus, task loss maximizes a lower bound of $I(Z_t;Y_t)$, not $I(Z;Y_t)$. Since $Z_t = g(Z)$ is deterministic:

$$\begin{aligned}I(Z;Y_t) &= I(Z_t;Y_t) + I(Z;Y_t|Z_t) \\ &\geq I(Z_t;Y_t).\end{aligned} \tag{12}$$

The gap $I(Z;Y_t|Z_t)$ vanishes only is Z_t is sufficient for Y_t. Therefore, loss-based MI maximization is inherently limited. Our method instead directly estimates $I(Z;Y_t)$ via the DV variational lower bound, i.e.,

$$\begin{aligned}I(Z;Y_t) \geq \sup_{\omega\in\Omega} &\mathbb{E}_{\mathbb{P}_{ZY_t}}[V_\omega(y_t|z)] \\ &- \log(\mathbb{E}_{\mathbb{P}_Z\otimes\mathbb{P}_{Y_t}}[e^{V_\omega(y_t|\tilde{z})}]),\end{aligned} \tag{13}$$

which targets the true objective. With a sufficiently expressive statistics network (universal approximation), this bound can approach $I(Z;Y_t)$, yielding a strictly tighter estimate than the loss-induced bound limited by $I(Z_t;Y_t)$.

B Experimental Setups

B.1 Details of Datasets and Downstream Tasks

We experiment on TweetEval and AffectEval benchmarks. The statistics are listed in Table 9.

TweetEval benchmark contains 6 classification tasks. *EmotionEval* (Mohammad et al., 2018) focuses on detecting the emotion in tweets, and originates from the Affects in Tweets held at SemEval-2018. Following Barbieri et al. (2020), we select the most common four emotions (i.e., anger, joy, sadness, and optimism) 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 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 tasks and 2 regression tasks. *GoEmotions* (Demszky et al., 2020) comprises Reddit comments annotated with 27 emotion categories or neutral. It is used fine-grained emotion prediction. Following Hu et al. (2024), we remove approximately 16% of multi-label data from the original corpus to facilitate a

Dataset	Task	Task Type	# Label	# Train	# Val	# Test	# Total
Homogeneous multi-task 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 multi-task 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 9: Dataset statistics on TweetEval and AffectEval. The homogeneous TweetEval contains six different classification tasks, and heterogeneous AffectEval includes two classification tasks and two regression tasks.

clearer evaluation of multi-class classification performance. *EmotionEval* (Mohammad et al., 2018), derived from SemEval-2018 Affects in Tweets, focuses on detecting the emotion evoked by a tweet. *Emobank* (Buechel and Hahn, 2017) is a large-scale text corpus covering 6 domains and 2 perspectives with manual annotations of continuous Valence-Arousal-Dominance (VAD) scores (ranging from 1 to 5 per dimension). 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, derived from SemEval-2018 Task 1: Affect in Tweets. It aims to identify 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). We did not use additional emotion labels in the dataset to better evaluate this regression task.

B.2 Description of Comparison Methods

Since most MTL methods use different benchmarks and experimental setups, it is difficult to fairly compare with different methods. We reproduced 13 representative MTL methods under the same settings (e.g., network architecture).

Equal Weighting (EW) is a typical baseline that applies equal weights for each task. Scale-Invariant loss (SI) is invariant to rescaling each loss with a logarithmic operation. 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 weight 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. Projecting Conflicting Gradient (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 variational method based on information bottleneck. VMTL (Shen et al., 2021) is a variational framework that uses Gumbel-Softmax priors for both representations and weights. Hierarchical MTL (de Freitas et al., 2022) is a hierarchical variational method with compressed task-specific representations based on information bottleneck. InfoMTL (Hu et al., 2025b) is an information-theoretic framework with variational implementation that simultaneously ensures sufficient shared representations and low-redundancy task-specific representations.

For a fair comparison, we reimplement each method under the same experimental setups (e.g., the model backbone). We use pre-trained language models BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019c) as the model backbone. We initialize BERT and RoBERTa via *bert-base-uncased*[2] and *roberta-base*[2] for fine-tuning. For LLMs, we compare with GPT-3.5, Llama-3, and Qwen-2.5. Following Hu et al. (2025b), GPT-3.5 is evaluated based on text-davinci-003, and predictions are obtained in a zero-shot setting by prompting with task descriptions and instructions. Results of Llama-3 and Qwen-2.5 are obtained via in-context learning with 4 demonstrations following Zhang et al. (2025).

[2] https://huggingface.co/

Hyperparameter	TweetEval	AffectEval
BERT backbone		
Trade-off weight β of $\ell_{t,\text{MI}}$	1	2
Dimension of MINE estimator	128	128
Perturbation radius ϵ	1	1
Number of epochs	20	20
Patience	3	3
Max length	256	256
Batch size	128	128
Dropout	0.2	0
Learning rate	$5e^{-5}$	$5e^{-5}$
Weight decay	0	0
RoBERTa backbone		
Trade-off weight β of $\ell_{t,\text{MI}}$	1	1
Dimension of MINE estimator	128	128
Perturbation radius ϵ	5	5
Number of epochs	20	20
Patience	3	3
Max length	256	256
Batch size	128	128
Dropout	0.2	0
Learning rate	$5e^{-5}$	$5e^{-5}$
Weight decay	0	0

Table 10: Hyperparameters of MTRA.

B.3 Evaluation Metrics

For classification, we primarily use the macro-averaged F1 over all classes 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, we utilize Pearson correlation for each VAD dimension on EmoBank, and use both Pearson and Spearman correlation coefficients on EI-Reg.

Then, we report a global metric based on the average (**Avg.**) of all task-specific metrics (Barbieri et al., 2020), i.e.,

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

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.

We also compute the average relative improvement ($\boldsymbol{\Delta}p$) of each method over the EW baseline as the multi-task performance measure (Maninis et al., 2019; Liu et al., 2021a), i.e.,

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

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).

Methods	GoEmotions	Emobank			Avg.	$\boldsymbol{\Delta}\mathbf{p}\uparrow$
	M-F1	V	A	D		
EW	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
MT-VIB	46.28	74.65	48.84	30.83	48.86	-1.00
InfoMTL	48.76	75.92	51.83	40.51	52.42	+7.86
MTRA	**50.84**	**78.78**	**55.66**	**45.89**	**55.48**	**+14.77**

(a)

Methods	EmotionEval	EI-Reg		Avg.	$\boldsymbol{\Delta}\mathbf{p}\uparrow$
	M-F1	Pear	Spear		
EW	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
MT-VIB	76.53	58.74	59.11	67.72	+2.18
InfoMTL	78.71	54.48	55.19	66.77	-0.04
MTRA	**79.73**	58.52	59.43	**69.35**	**+4.30**

(b)

Table 11: Results (%) on two heterogeneous multi-task scenarios. We experiment with the RoBERTa backbone. Best results are highlighted in **bold**, and second-best results are underlined.

Methods	Emobank			EI-Reg		Avg.	$\boldsymbol{\Delta}\mathbf{p}\uparrow$
	A	V	D	Pear	Spear		
EW	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
MT-VIB	79.92	54.83	47.39	60.09	60.74	60.57	+1.88
InfoMTL	80.93	56.17	50.46	59.71	60.32	61.27	+3.24
MTRA	81.28	56.04	**50.86**	61.38	**62.35**	**62.29**	**+5.00**

(a)

Methods	GoEmotions	EmotionEval	Avg.	$\boldsymbol{\Delta}\mathbf{p}\uparrow$
	M-F1	M-F1		
EW	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
MT-VIB	46.19	77.99	62.09	+0.08
InfoMTL	**51.49**	78.24	**64.87**	**+5.92**
MTRA	50.20	**79.55**	**64.87**	+5.38

(b)

Table 12: Results (%) on two homogeneous multi-task scenarios. We experiment with the RoBERTa backbone. Best results are highlighted in **bold**, and second-best results are underlined.

B.4 Implementation Details

We experiment using an epoch number of 20, a total batch size of 128, and a maximum token length of 256. The maximum patience for early stopping is set to 3 epochs. The learning rate is set to $5e^{-5}$. The dropout rate is set to 0.2 for TweetEval and 0 for AffectEval. The perturbation radius ϵ is searched from $\{1, 5\}$. The dimension in MINE estimator is set to 128. The trade-off parameter β is searched from $\{0.1, 1, 2\}$. We report the detailed hyperparameter settings of MTRA with RoBERTa and BERT backbone models on two benchmarks

Methods	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**	**Δp ↑**
MTRA	**76.97**±1.54	**55.48**±4.24	**68.11**±2.59	80.10±0.10	72.05±0.81	**67.02**±2.61	**69.96**±1.62	**+7.06**
w/o TA	75.00±1.89	45.59±1.09	65.33±6.08	**80.69**±0.79	**72.28**±0.40	63.71±2.14	67.10±1.86	+1.48
w/o SA	75.45±1.15	47.80±1.74	66.56±6.16	77.64±2.72	70.78±0.91	65.07±1.14	67.22±0.96	+2.09
w/o TA & SA	75.81±1.05	46.19±6.01	66.17±5.81	78.58±2.00	71.00±1.80	65.24±2.31	67.16±0.96	+1.75
w/o TA & SA & SI	74.37±0.56	44.08±5.26	65.32±1.84	79.04±1.43	70.64±1.71	63.59±2.43	66.17±0.43	0.00

(a) Fine-grained results (%) of ablation studies on TweetEval.

Methods	GoEmotions	EmotionEval	Emobank			EI-Reg		**Avg.**	**Δp ↑**
	M-F1	M-F1	V	A	D	Pear	Spear		
MTRA	48.48±0.87	78.90±1.39	**79.59**±1.21	**55.40**±0.65	**44.45**±0.93	**57.38**±1.39	**57.81**±1.05	**61.20**±0.65	**+7.18**
w/o TA	48.40±1.13	**79.36**±1.78	76.92±0.82	51.83±2.08	39.21±1.53	53.95±1.69	54.54±1.72	59.50±0.40	+3.57
w/o SA	**49.15**±1.62	78.90±0.75	78.24±0.54	53.56±2.12	43.65±2.76	55.49±2.44	56.28±2.37	60.60±0.50	+6.06
w/o TA & SA	47.08±0.72	78.22±0.49	75.61±1.39	50.35±5.02	37.26±4.78	51.55±3.99	52.60±3.82	57.94±2.02	+0.61
w/o TA & SA & SI	47.13±0.33	77.97±0.63	75.62±0.79	49.44±4.70	36.47±4.02	51.01±4.62	52.23±4.68	57.64±2.12	0.00

(b) Fine-grained results (%) of ablation studies on AffectEval.

Table 13: Fine-grained ablation study of MTRA. We experiment with the RoBERTa backbone.

Methods	Data per	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**	**Δp ↑**
EW	20%	66.01	53.12	56.09	75.57	70.20	53.60	62.43	0.00
SI	20%	64.35	52.96	55.84	76.32	70.07	53.84	62.23	-0.34
UW	20%	65.75	43.59	55.24	79.15	70.77	56.16	61.78	-1.59
MT-VIB	20%	56.09	47.29	59.11	77.30	70.31	49.90	60.00	-4.18
InfoMTL	20%	68.85	53.42	57.78	77.92	70.40	60.63	**64.83**	**+4.07**
MTRA	20%	68.01	53.27	58.87	77.43	70.76	59.23	64.59	+3.67
EW	60%	73.59	47.68	63.45	77.85	71.76	63.95	66.38	0.00
SI	60%	73.23	46.45	64.03	78.45	71.74	63.93	66.31	-0.24
UW	60%	74.20	47.24	62.98	79.25	71.91	61.47	66.17	-0.45
MT-VIB	60%	69.61	47.87	67.42	78.42	71.55	63.02	66.31	+0.04
InfoMTL	60%	75.12	48.48	63.50	79.27	70.71	63.17	66.71	+0.50
MTRA	60%	75.43	49.18	65.89	78.20	70.97	63.17	**67.14**	**+1.27**
EW	80%	73.59	44.52	66.27	78.50	71.90	63.23	66.34	0.00
SI	80%	74.34	48.31	67.71	78.51	71.15	63.97	67.33	+1.98
UW	80%	75.06	47.98	65.87	79.00	70.28	63.39	66.93	+1.30
MT-VIB	80%	72.75	43.60	64.55	78.59	70.18	62.36	65.34	-1.57
InfoMTL	80%	77.10	46.49	68.32	78.94	70.62	65.83	67.88	+2.53
MTRA	80%	76.17	50.77	69.42	79.46	70.57	64.36	**68.46**	**+3.91**

Table 14: Fine-grained results (%) against different training data size on TweetEval.

Methods	Data per	GoEmotions M-F1	EmotionEval M-F1	Emobank V	Emobank A	Emobank D	EI-Reg Pear	EI-Reg Spear	**Avg.**	**Δp ↑**
EW	20%	37.85	67.47	58.48	17.09	11.41	40.98	42.28	43.99	0.00
SI	20%	37.86	67.52	57.14	17.81	11.48	37.67	38.60	43.08	-1.86
UW	20%	40.27	72.28	65.21	34.37	18.36	43.30	44.40	48.93	+19.17
MT-VIB	20%	36.31	63.35	61.25	14.62	17.92	45.72	47.21	44.35	+4.30
InfoMTL	20%	39.01	71.93	64.87	32.44	26.27	45.47	45.81	**49.44**	**+24.07**
MTRA	20%	40.90	70.84	62.33	28.32	18.27	46.93	47.90	48.86	+17.77
EW	60%	45.12	77.26	72.74	43.36	32.08	47.85	48.82	55.03	0.00
SI	60%	44.64	75.54	71.71	44.83	29.08	47.45	48.16	54.13	-1.71
UW	60%	44.86	75.56	73.83	44.56	34.68	49.22	50.02	55.27	+1.00
MT-VIB	60%	43.21	71.47	73.34	40.53	27.53	49.01	50.17	52.85	-3.94
InfoMTL	60%	44.93	76.50	74.01	39.14	37.80	44.35	45.17	54.13	-1.39
MTRA	60%	45.65	77.62	76.91	52.48	39.17	55.26	56.19	**58.80**	**+8.31**
EW	80%	46.12	77.61	74.39	47.79	31.14	51.82	52.51	56.75	0.00
SI	80%	46.19	78.15	74.26	47.88	31.58	48.97	49.24	56.17	-1.13
UW	80%	47.69	78.19	75.39	53.11	37.67	53.12	54.00	58.71	+4.49
MT-VIB	80%	45.73	73.47	73.89	47.56	28.30	49.65	50.55	54.80	-3.39
InfoMTL	80%	45.78	77.59	74.56	49.48	37.50	50.42	50.86	56.96	+1.09
MTRA	80%	49.01	79.88	77.71	53.09	39.99	52.91	53.94	**59.81**	**+6.57**

Table 15: Fine-grained results (%) against different training data size on AffectEval.

Methods	Label Noise	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**	**Δp $\uparrow$**
EW	20%	76.58	49.15	78.29	65.84	67.29	71.05	68.03	0.00
SI	20%	75.75	45.55	79.12	63.91	67.01	69.83	66.86	-2.07
UW	20%	74.62	47.90	80.30	67.49	65.10	70.30	67.62	-0.72
MT-VIB	20%	73.55	48.62	78.43	66.46	64.59	70.03	66.95	-1.56
InfoMTL	20%	76.85	49.18	78.52	65.63	66.39	70.38	67.82	-0.31
MTRA	20%	77.46	50.83	79.38	71.03	68.72	70.58	**69.67**	**+2.55**
EW	40%	76.52	44.90	78.79	64.23	67.17	70.69	67.05	0.00
SI	40%	73.89	47.03	79.81	56.80	60.98	69.98	64.75	-3.20
UW	40%	75.62	47.09	78.26	67.22	65.64	69.20	67.17	+0.55
MT-VIB	40%	75.19	48.43	78.69	66.36	66.74	70.01	67.57	+1.29
InfoMTL	40%	75.98	48.85	79.74	63.21	66.49	70.57	67.47	+1.09
MTRA	40%	76.97	50.92	80.53	70.14	67.15	69.75	**69.24**	**+4.01**
EW	60%	74.57	47.79	78.92	63.89	63.57	71.45	66.70	0.00
SI	60%	74.07	46.74	79.88	56.41	64.52	69.69	65.22	-2.39
UW	60%	71.89	41.73	79.23	61.52	62.20	69.58	64.36	-4.06
MT-VIB	60%	73.14	44.88	79.00	63.03	63.17	70.67	65.65	-1.83
InfoMTL	60%	75.22	42.77	78.94	62.38	63.81	71.50	65.77	-1.92
MTRA	60%	77.36	47.19	80.68	69.04	67.89	71.05	**68.87**	**+3.17**

Table 16: Fine-grained results (%) against different strengths of label noises on TweetEval. RoBERTa is the default backbone model.

in Table 10. 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 across Different Pair-wise Task Combinations

We further evaluate across different pair-wise task combinations. We compare model performance under 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*).

The fine-grained results across pair-wise heterogeneous and homogeneous MTL scenarios are shown in Table 11 and Table 12, respectively. MTRA outperforms comparison methods in terms of Avg. and Δp on pair-wise MTL scenarios. This demonstrates the effectiveness of MTRA in heterogeneous and homogeneous MTL settings.

C.2 Fine-grained Results of Ablation Studies

Table 13(a) and Table 13(b) show the fine-grained ablation results of the ablation studies on TweetEval and AffectEval. RoBERTa is the model backbone. w/o TA refers to removing information maximization for task-specific alignment, i.e., removing $\ell_{t,\text{MI}}$ and preserve $\ell_{t,\text{Task}}$ in Eq(3) for task prediction. w/o SA indicates removing conditional information minimization for self-alignment. w/o SI represents ablating the logarithmic operation.

C.3 Fine-grained Results under Data-constrained Conditions

We evaluate MTRA and 5 representative MTL methods when training with limited data by adjusting different ratios of the training set. Following Hu et al. (2025b), all methods are trained on randomly sampled subsets from the original training set, and we report the average results on the test set.

Table 14 and Table 15 show fine-grained results of MTRA and comparison baselines against different training data sizes on TweetEval and AffectEval, respectively. We experiment with the RoBERTa backbone model.

C.4 Fine-grained Results under Label-noisy Conditions

We evaluate the robustness against noisy labels during training. Specifically, we randomly choose 20%, 40%, 60% of training data and flip their labels to any category randomly with equal probability. For a task with K classes, only $1-\frac{1}{K}$ of the flipped labels are erroneous.

Table 16 shows fine-grained results of MTRA and comparison baselines against different ratios of label noises where RoBERTa is the model backbone.

C.5 Sensitivity Analysis of Hyperparameters

Our method MTRA involves two hyperparameters: the trade-off parameter β in Eq (3) and the perturba-

Methods	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**	$\mathbf{\Delta p} \uparrow$
EW (baseline)	74.37	44.08	65.32	79.04	70.64	63.59	66.17	0.00
MTRA								
$\beta = 1$	76.97	55.48	68.11	80.10	72.05	67.02	**69.96**	**+7.06**
$\beta = 2$	77.24	45.16	61.99	79.91	70.55	65.56	66.74	+0.88
$\beta = 0.5$	77.10	48.15	64.90	80.64	72.34	65.07	68.03	+3.17

(a) Fine-grained results (%) of parameter analysis on TweetEval.

Methods	GoEmotions	EmotionEval	Emobank			EI-Reg		**Avg.**	$\mathbf{\Delta p} \uparrow$
	M-F1	M-F1	V	A	D	Pear	Spear		
EW (baseline)	47.13	77.97	75.62	49.44	36.47	51.01	52.23	57.64	0.00
MTRA									
$\beta = 1$	48.48	78.90	79.59	55.40	44.45	57.38	57.81	**61.20**	**+7.18**
$\beta = 2$	49.57	79.53	78.59	53.74	43.25	54.68	55.11	60.63	+5.99
$\beta = 0.5$	49.48	79.02	78.74	52.52	40.70	52.91	53.48	59.75	+4.18

(b) Fine-grained results (%) of parameter analysis on AffectEval.

Table 17: Fine-grained performance against different values of the trade-off parameter β. We experiment with the RoBERTa backbone.

Methods	EmotionEval M-F1	HatEval M-F1	IronyEval F1(i.)	OffensEval M-F1	SentiEval M-Recall	StanceEval M-F1 (a. & f.)	**Avg.**	$\mathbf{\Delta p} \uparrow$
EW (baseline)	74.37	44.08	65.32	79.04	70.64	63.59	66.17	0.00
MTRA								
$\epsilon = 5$	76.97	55.48	68.11	80.10	72.05	67.02	**69.96**	**+7.06**
$\epsilon = 1$	76.25	48.74	70.32	80.03	71.54	68.13	69.17	+5.07
$\epsilon = 0.1$	76.48	47.39	68.94	80.27	71.80	65.61	68.42	+3.71

(a) Fine-grained results (%) of parameter analysis on TweetEval.

Methods	GoEmotions	EmotionEval	Emobank			EI-Reg		**Avg.**	$\mathbf{\Delta p} \uparrow$
	M-F1	M-F1	V	A	D	Pear	Spear		
EW (baseline)	47.13	77.97	75.62	49.44	36.47	51.01	52.23	57.64	0.00
MTRA									
$\epsilon = 5$	48.48	78.90	79.59	55.40	44.45	57.38	57.81	**61.20**	**+7.18**
$\epsilon = 1$	48.26	78.54	79.82	55.61	43.42	56.80	57.34	60.87	+6.52
$\epsilon = 0.1$	49.13	79.32	77.36	52.82	39.70	53.35	54.34	59.73	+4.07

(b) Fine-grained results (%) of parameter analysis on AffectEval.

Table 18: Fine-grained performance against different values of the perturbation radius ϵ. We experiment with the RoBERTa backbone.

tion radius ϵ. We evaluate the effect of β by varying its value in $\{0.5, 1, 2\}$, and study the impact of ϵ using $\{0.1, 1, 5\}$.

The results of EW baseline and our MTRA against different β and ϵ on both datasets are shown in Table 17 and Table 18, respectively. We observe that $\beta = 1$ and $\epsilon = 5$ consistently achieve the best performance on both benchmarks, confirming the reliability of the default parameter settings.

C.6 Computational Efficiency Analysis

Computational Overhead Analysis This paper adopts a hard parameter-sharing pattern and adopt the same architecture for MTL. Compared to the soft parameter-sharing, the hard pattern leads to lower training and inference costs. Compared to other MTL methods, MTRA has advantages in terms of computations and memory costs: 1) loss-based methods (e.g., GLS, DWA, UW, IMTL-L, and RLW) require computing all task-specific losses per batch to update weighting strategies, often demanding large batch sizes or task-balanced sampling, which increases memory usage. Our MTRA allows one or multiple tasks in a batch sample, making it more suitable for memory-limited settings. 2) Gradient-based methods (e.g., PCGrad, and GradNorm) need to operate gradients across tasks, introducing high computations and memory costs during training. MTRA allows for task-wise gradient updates, which can mitigate the overhead to some extent, although self-alignment via adversarial estimator introduces additional gradient computations. 3) Probabilistic methods (e.g., MT-VIB, VMTL, Hierarchical MTL, and InfoMTL) typically rely on high-cost variational inference and distribution sampling procedures. Our method avoids such

Methods	# Param for Training	# Param for Testing	# Memory for Training	# Memory for Testing
EW (baseline)	110M	110M	×1	×1
PCGrad	110M	110M	×3	×1
InfoMTL	110M + 576k	110M	×2.5	×1
MTRA	110M + 92k	110M	×1.5	×1

Table 19: Results of computational efficiency analysis. The default settings are batch size 128 and max length 256, with the memory usage of the EW baseline recorded as 1 GPU memory unit (approximately 20GB).

probabilistic operations and employs a lightweight two-layer MLP to learn the MINE estimator for task-specific alignment, leading to lower computational overhead.

Efficiency Analysis during Training and Testing Table 19 compares the number of model parameters (excluding the task-specific linear head) and GPU memory usage of MTRA, InfoMTL (the best loss-based and probabilistic method in Table 1), PCGrad (gradient-based method), and the EW baseline under the TweetEval benchmark and the RoBERTa backbone. During training, loss-based and gradient-based methods (e.g., PCGrad) require storing task-wise gradients, leading to substantially higher memory consumption. Probabilistic methods (e.g., InfoMTL) further increase both parameters and memory usage due to variational inference and distribution sampling. MTRA introduces a lightweight auxiliary module for task alignment, resulting in a moderate training memory increase. All methods use the same backbone, maintaining the same cost during testing. Overall, our MTRA has advantages in terms of computations and memory costs.