Supervised contrastive learning as multi-objective optimization for fine-tuning large pre-trained language models

Youness Moukafih, Mounir Ghogho, Kamel Smaïli

SUPERVISED CONTRASTIVE LEARNING AS MULTI-OBJECTIVE OPTIMIZATION FOR FINE-TUNING LARGE PRE-TRAINED LANGUAGE MODELS

Youness Moukafih[1,2], *Mounir Ghogho*[1] *and Kamel Smaili*[2]

[1]TICLab, College of Engineering and Architecture, Université Internationale de Rabat
[2]LORIA, Université de Lorraine
{youness.moukafih, kamel.smaili}@loria.fr {youness.moukafih, mounir.ghogho}@uir.ac.ma

ABSTRACT

Recently, Supervised Contrastive Learning (SCL) has been shown to significantly outperform the well-known cross-entropy loss-based learning on most classification tasks. In SCL, a neural network is trained to optimize two objectives: pull an anchor and positive samples together in the embedding space, and push the anchor apart from the negatives. These two different objectives may be conflicting with one another, thus requiring a trade-off between them during optimization. In this work, we formulate the SCL problem as a Multi-Objective Optimization problem for the fine-tuning phase of RoBERTa language model. Two methods are utilized to solve the optimization problem: (i) the linear scalarization (LS) method, which minimizes a weighted linear combination of per-task losses; and (ii) the Exact Pareto Optimal (EPO) method which finds the intersection of the Pareto front with a given preference vector. We evaluate our approach on several GLUE benchmark tasks, without using data augmentations, memory banks, or generating adversarial examples. The empirical results show that the proposed learning strategy significantly outperforms a strong competitive contrastive learning baseline.

1. INTRODUCTION

Recently, contrastive learning has achieved state-of-the-art performance in various artificial intelligent applications, including Natural Language Processing (NLP) (1; 2; 3), Computer Vision (CV) (4) and graph representation learning (5; 6). Many approaches have been proposed to learn high-quality representations by minimizing a contrastive loss. The main common idea behind these approaches is as follows: train an encoder, a neural network, to increase both intra-class compactness and inter-class separability in the embedding space. In other words, the goal is to train a model to optimize two objectives: embed examples belonging to the same class close to each other, and embed examples from different classes further apart.

Contrastive learning has been used in both self-supervised and supervised Learning settings. In the former, positive pairs are created by performing data augmentation methods, while the negatives are formed by the anchor and randomly chosen examples from the same mini-batch. In the latter, label information is leveraged by considering samples belonging to the same class as positive examples for each other, while negatives are samples from the remaining classes.

The training in both self-supervised and Supervised Contrastive Learning settings is usually done in two steps: a pretraining step where a contrastive loss is minimized using the encoder's normalized representations, followed by a conventional training step where a simple linear model having as input the learnt representation is trained using the cross-entropy loss. In (1), the authors proposed to combine the cross-entropy loss and a SCL (7) for fine-tuning a large language model. Their method improves the performance on several NLP classification tasks from the GLUE benchmark over fine-tuning RoBERTa-large model using cross-entropy loss, especially in the few-shot learning setting.

A variety of contrastive losses have been proposed for optimizing the two objectives mentioned above (8; 7; 2). The main objective of this work is to investigate the issue of conflicting objectives and devise new solutions using the Multi-Objective Optimization (MMO) framework.

Several algorithms exist for MOO. The most straightforward approach to MOO is the linear scalarization (LS) which minimizes the weighted sum of the objective functions given a preference vector r. The preference vector, consisting of the weighting parameters, represents a single trade-off between objectives. LS has a major limitation: the Pareto optimal solution cannot be obtained for all preference vectors when the objectives are non-convex. However, LS tends to work well in practice. In (9), the authors proposed the multiple-gradient descent algorithm (MGDA), which uses gradient-based optimization to find one solution on the Pareto front. However, different solutions are found for different initializations and the preference vector is not taken into account. In (10), the authors proposed a multi-task learning (MTL) algorithm from the MOO perspective that scales to high-dimensional problems. Instead of the uniform weight strategy, they used the MGDA algorithm to determine the optimal weights to obtain

a solution on the Pareto front. Their method, however, finds only one single arbitrary Pareto-optimal solution. In (11), the authors proposed Pareto multi-task learning (Pareto MTL), a method that splits the objective space into separate cones given a set of preference rays, and returns a solution per cone. Their approach is capable of finding several points on the Pareto front; However, it scales poorly with the number of cones and does not converge to the exact desired ray on the Pareto front. Recently, (12) proposed a new approach to find the Exact Pareto Optimal (EPO) solution in the objective space (the intersection of the Pareto front with a given preference ray).
The technical novelty of our work resides in the formulation of the SCL problem as a MOO problem. To the best of our knowledge, this is the first work on fine-tuning a large language model by minimizing two objective functions using MOO. The first objective function encourages the encoder to maximize the agreement between a cluster of points belonging to the same class in the latent space, while the second guides the encoder to represent sentences form different classes far away from each other in the latent space. We address the MOO problem using both LS and EPO methods.
We show the merit of our approach on multiple datasets form the GLUE natural language understanding benchmark. We evaluate the method on both few-shot (20, 50, 100 labeled examples) learning as well as the full dataset training settings. The experiments show the superiority of our approach over two very strong baselines that fine-tune RoBERTa-base language model using CE only and CE + SCL objective respectively.
Our contributions can be summarized as follows:

- we formulate the SCL problem as a MMO problem;
- we propose a method to fine-tune pre-trained language models by using SCL and MOO (here we focus on RoBERTa-base language models);
- we adapt two well-known MOO approaches to solve several downstream tasks from the GLUE benchmark.

2. THE PROPOSED APPROACH

We propose to fine-tune the pre-traind RoBERTa language model by minimizing an objective function that combines the well-known cross-entropy and a supervised contrastive loss (See Figure 1). We formulate the contrastive loss optimization as a MOO problem. To solve the latter, we investigate both linear scalarization and exact pareto optimal methods (12). Similar to (1), we fine-tune RoBERTa on single sentence and sentence-pair classification tasks. Following common practice for fine-tuning pre-trained language models for classification (13; 1), we consider the $[CLS]$ token to be the final representation of the input data.

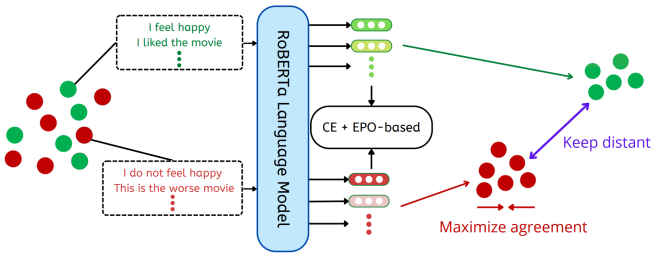


Fig. 1: The general framework of our proposed approach

2.1. Preliminaries

Let us denote a labeled dataset as $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_i$, where $\boldsymbol{x}_i \in \mathcal{X}$ represents the i^{th} instance of the dataset and $y_i \in \mathcal{Y} = \{1, ..., C\}$ is its label. We train an encoder function (a neural network) $f_{\boldsymbol{\theta}} : \mathcal{X} \longrightarrow \mathcal{Y}$ parameterized by $\boldsymbol{\theta}$.
The goal of MOO is to jointly minimize m non-negative objective functions. In our setting, each objective function is the expectation of the individual (or mini-batch) loss, $\tilde{\ell}_j$, over labeled instances of $\boldsymbol{x}$ and y, randomly sampled from the data distribution $\mathcal{P}_D$:

$$\ell_j(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_D}[\tilde{\ell}_j(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))] \tag{1}$$

where $j \in \{1, \ldots, m\}$. The goal of MOO is to find Pareto optimal solutions.
We define a partial ordering on the objective space by $\boldsymbol{\ell}(\boldsymbol{\theta}) \preceq \boldsymbol{\ell}(\boldsymbol{\theta}')$, where $\boldsymbol{\ell}(\boldsymbol{\theta}) = [\ell_1(\boldsymbol{\theta}), \ldots, \ell_m(\boldsymbol{\theta})]^T \in \mathbb{R}_+^m$, if for all $j \in \{1, \ldots, m\}$, $\ell_j(\boldsymbol{\theta}) \leq \ell_j(\boldsymbol{\theta}')$; for strict inequality, i.e. $\boldsymbol{\ell}(\boldsymbol{\theta}) \prec \boldsymbol{\ell}(\boldsymbol{\theta}')$, we have $\ell_j(\boldsymbol{\theta}) < \ell_j(\boldsymbol{\theta}')$ for some of the values of j.
Definition (Pareto dominance). A solution $\boldsymbol{\theta}_1$ dominates a solution $\boldsymbol{\theta}_2$ if $\boldsymbol{\ell}(\boldsymbol{\theta}_1) \prec \boldsymbol{\ell}(\boldsymbol{\theta}_2)$. In other words, $\boldsymbol{\theta}_1$ is not worse than $\boldsymbol{\theta}_2$ on any objective, and $\boldsymbol{\theta}_1$ is better than $\boldsymbol{\theta}_2$ on at least one objective, i.e.: $\exists q \in \{1, \ldots, m\}$ s.t $\ell_q(\boldsymbol{\theta}_1) < \ell_q(\boldsymbol{\theta}_2)$. A point that is not dominated by any other point is called Pareto optimal solution. The set of all Pareto optimal solutions is called Pareto set, denoted by, $\mathcal{P}_{\boldsymbol{\theta}}$, and its image is called the Pareto front $\mathcal{P}_{\boldsymbol{\ell}} = \{\boldsymbol{\ell}(\boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\mathcal{P}_{\boldsymbol{\theta}}}$.

2.2. The proposed training strategy

We first define some needed notations, formulate the learning problem of interest, and then provide details of the proposed solution. Let $\mathcal{S}^k = \{(\boldsymbol{x}_i, y_i) | y_i = k\}$ denotes the set of all samples belonging to class k within the corpus $\mathcal{D}$. Let $\mathcal{B}_k$ be a mini-batch of N_k examples randomly sampled from $\mathcal{S}^k$. Let $\mathcal{H}_k = f_{\boldsymbol{\theta}}(\mathcal{B}_k) \in \mathbb{R}^{N_k \times d}$ be the ℓ_2 normalization of the highest

level representation of the neural network:

$$\mathcal{H}_k = \begin{bmatrix} \boldsymbol{h}_1^\top \\ \boldsymbol{h}_2^\top \\ \vdots \\ \boldsymbol{h}_{|\mathcal{B}_k|}^\top \end{bmatrix} \in \mathbb{R}^{N_k \times d}$$

where $\boldsymbol{h}_j$ is the embedding vector corresponding to instance j, whose dimension is denoted by d, and $|\cdot|$ denotes the cardinality operator.
Similar to (1; 7), our main goal is to maximize the similarity between points that belong to the same class and minimizing the similarity between elements of different classes. We now proceed with the formulation of the two objective functions. First, we define the following matrices:

$$\mathcal{M}^{(k)} = \mathcal{H}_k \mathcal{H}_k^T \in \mathbb{R}^{N_k \times N_k},$$

$$\mathcal{N}^{(k)} = [\mathcal{H}_k \mathcal{H}_{k'}^T]_{k' \in \mathcal{Y}, k' \neq k} \in \mathbb{R}^{N_k \times \overline{N_k}}$$

where $\overline{N_k} = \sum_{k' \neq k} N_k$ and $[.]$ denotes the horizontal concatenation operator. Matrices $\mathcal{M}^{(k)}$, with $k = 1, \cdots, C$, contain the intra-class similarities, whereas matrices $\mathcal{N}^{(k)}$ contain inter-class similarities.

In this work, we aim to find an encoder which maximizes the intra-class similarities and minimizes the inter-class similarities. To this end, we define the following loss functions:

$$\tilde{\ell}_{pos} = -\frac{1}{C} \sum_{k=1}^{C} \frac{1}{N_k} \sum_{i=1}^{N_k} \log \left[\frac{1}{N_k - 1} \sum_{p=1, p \neq i}^{N_k} \exp\left(\mathcal{M}_{i,p}^{(k)} / \tau \right) \right] \tag{2}$$

$$\tilde{\ell}_{neg} = \frac{1}{C} \sum_{k=1}^{C} \frac{1}{N_k} \sum_{i=1}^{N_k} \log \left[\frac{1}{\overline{N_k}} \sum_{n=1}^{\overline{N_k}} \exp\left(\mathcal{N}_{i,n}^{(k)} / \tau \right) \right] \tag{3}$$

where C is the number of classes, $\tau \in \mathcal{R}_+$ is a temperature parameter, and $\mathcal{M}_{i,p}^{(k)}$ and $\mathcal{N}_{i,n}^{(k)}$ denotes the (i,p)th element of $\mathcal{M}^{(k)}$ and the (i,n)th element of $\mathcal{N}^{(k)}$ respectively. The objective functions are defined as the expectation of the above (mini-batch) loss functions over the distribution of the data.

The overall (single mini-batch) loss that we propose in this work is a weighted linear combination of the above conficting objectives and the CE loss. When including the latter, the two objectives to minimize become:

$$\ell_1 = \lambda \tilde{\ell}_{pos} + (1 - \lambda) \mathcal{L}_{CE} \tag{4}$$

$$\ell_2 = \lambda \tilde{\ell}_{neg} + (1 - \lambda) \mathcal{L}_{CE} \tag{5}$$

where $\mathcal{L}_{CE}$ is the conventional cross-entropy loss, and λ is a hyper-parameter that controls the weight of the SCL term in the objective function. The MOO is solved using both LS and exact Pareto optimal methods to fine-tuning the pretrained language model.

• **Linear Scalarization method**: this is the most straightforward approach to solve a MOO problem. It converts the latter to a single-objective optimization problem (Eq 6). Indeed, LS optimizes the weighted sum of the objectives, i.e.

$$\theta^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x}, y) \sim \mathcal{P}_D} \sum_{j=1}^{m} r_j \ell_j \tag{6}$$

where $\boldsymbol{r} \in \Omega^m$ is the preference vector, with

$$\Omega^m := \left\{ \boldsymbol{r} \in \mathbb{R}_+^m \;\middle|\; \sum_{j=1}^{m} r_j = 1, \text{and } r_j \geq 0 \; \forall j \right\}.$$

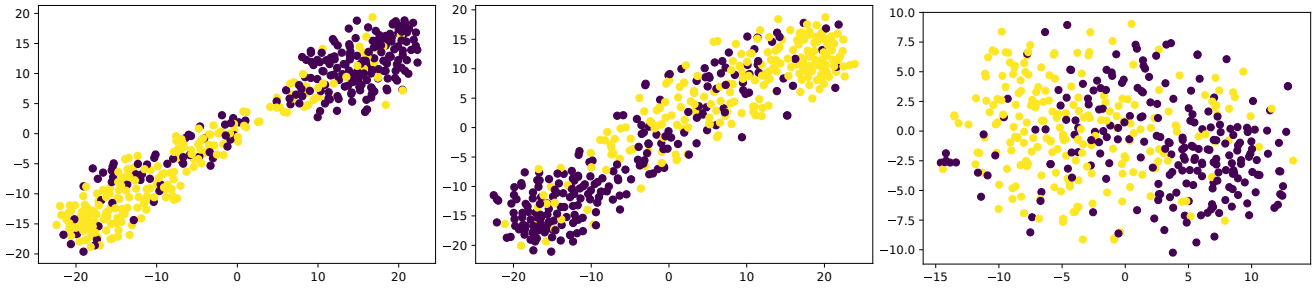
In our problem formulation $m = 2$. Although LS has some theoretical limitations, it has been shown to work well in practice.
• **Exact Pareto Optimal method**: it finds the intersection of the Pareto front with a given preference ray. This is achieved by considering the preference vector $\boldsymbol{r}$ as a ray in the loss space and training a neural network to reach a Pareto optimal solution on the inverse ray $\boldsymbol{r}^{-1}$. Thus, an Exact Pareto optimal (EPO) solution with respect to a preference vector $\boldsymbol{r}$ belongs to the set:

$$\mathcal{P}_{\boldsymbol{r}} = \left\{ \boldsymbol{\theta}^* \in \mathcal{P}_{\boldsymbol{\theta}} \middle| r_1 \ell_1^* = \cdots = r_m \ell_m^* \right\} \tag{7}$$

where $\ell_j^* = \ell_j(\boldsymbol{\theta}^*)$. This is achieved by balancing two goals: finding a descent direction towards the Pareto front and approaching the desired ray (12).

3. TRAINING DETAILS

We evaluate our approach by measuring top-1 accuracy on multiple tasks of the GLUE natural language understanding benchmark namely, SST-2, QNLI and MNLI datasets (14). In our experiments, for few-shot learning setting, similar to (1), we sample 500 examples from the original validation set of each dataset to build our validation set, and half of the validation to build the test set. Note that, we evaluate our proposed approach only on SST2, QNLI and MNLI tasks to make a fair comparison with the baselines, however our method can be applied on other tasks. We run each experiment with 10 different seeds, and report the average test accuracy and the standard deviation along with the baselines. Best hyperparameters combination are picked based on the average validation accuracy. In this paper, we use RoBERTa-base due to the GPU RAM constraint. During all the fine-tuning runs, we use AdamW optimizer with a learning rate of 1e-5, batch size of 16, and dropout rate of 0.1. We optimized the temperature hyper-parameter on the validation set by sweeping for $\tau \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and $r_1 \in \{0.1, 0.3, 0.5\}$. Simulations show that models with best test accuracies across all experimental settings overwhelmingly use the hyperparameter combination $\tau = 0.3$, $\lambda = 0.3$, and $\boldsymbol{r} = [0.1, 0.9]$



(a) CE + LS-based (b) CE + EPO-based (c) CE + SCL

Fig. 2: T-SNE plot of the learned embeddings on the SST-2 test set where we have 20 annotated examples to fine-tune RoBERTa-base language model fine-tuned with CE + LS-based (left), with CE + EPO-based (mid) and with CE + SCL objective (right).

4. RESULTS & ANALYSIS

Here, we report the obtained results of our approach in the few-shot learning setting (20, 50, 100 annotated examples) and compare them with those obtained by the baselines that fine-tune the RoBERTa-base with CE and CE + SCL loss, respectively. Performance is measured in terms of the Accuracy metric on the test set. We run each experiment with 10 different seeds (details of the experimental setup are explained in Experiments section). For both CE + LS-based and CE + EPO-based losses, experiments were carried out using different preference vectors. As shown in Table 1, we observe that our approach obtains significantly better performance than the baselines. For instance, the CE + EPO-based loss achieves, on SST-2, an accuracy of 68.96%, which is a 9.24 and 8.6 percentage points improvement over CE and CE + SCL respectively when the number of annotated examples is 20. Similarly, CE + LS-based objective achieves better results than the baselines when the RoBERTa-base is fine-tuned using 20 labeled examples, with 1.85 percentage points improvement on QNLI and 1.9 percentage points improvement on MNLI. This shows that our approach *generalizes* better than the baselines. We believe that this is due to the fact that good generalization (high-quality representations) requires capturing well the similarity between examples in one class and contrasting them with examples from other classes. Note that, as we increase the number of annotated examples, performance improvement over the baseline decreases, leading to 1.5 percentage points improvement on SST-2 with 50 examples and 0.23 percentage points improvement with 100 examples. However, our CE + EPO-based method achieves consistent improvements on MNLI dataset across all data regimes. On QNLI dataset, we see that our method is outperformed by the CE + SCL when the number of labeled examples is 50. However, overall, our method performs better than the baselines in the few-shot learning setting. Figure 2 shows the tSNE plots of the learned sentence embeddings on SST-2 test set when RoBERTa-base is fine-tuned using only 20 annotated examples with CE + LS-based, CE + EPO-based, and CE + SCL losses. As we can clearly see, our approach forces the encoder to better separate the classes in the embedding space, while forcing it to achieve more compact clustering.

Loss	N	SST-2	QNLI	MNLI
CE	20	59.72 ± 5.9	50.57 ± 1.8	34.07 ± 1.5
CE + SCL	20	60.34 ± 5.3	50.80 ± 1.8	33.25 ± 1.5
CE + LS-based (Ours)	20	**72.06 ± 5.4**	**51.89 ± 1.6**	**34.81 ± 1.1**
CE + EPO-based (Ours)	20	**68.96 ± 3.5**	**51.44 ± 1.2**	33.94 ± 0.9
CE	50	81.64 ± 1.6	61.20 ± 2.5	37.36 ± 2.1
CE + SCL	50	81.42 ± 4.8	**66.02 ± 2.4**	38.20 ± 1.1
CE + LS-based (Ours)	50	**81.92 ± 1.6**	62.06 ± 4.2	**39.55 ± 1.8**
CE + EPO-based (Ours)	50	**82.93 ± 0.9**	62.93 ± 1.6	**40.52 ± 1.5**
CE	100	85.32 ± 1.4	72.74 ± 1.4	44.87 ± 1.5
CE + SCL	100	85.41 ± 1.3	73.37 ± 1.1	43.79 ± 1.9
CE + LS-based (Ours)	100	**85.64 ± 1.4**	73.39 ± 1.2	**46.51 ± 1.7**
CE + EPO-based (Ours)	100	**85.43 ± 1.2**	**74.29 ± 0.8**	**46.89 ± 1.6**

Table 1: Few-shot classification accuracies on test sets of the GLUE benchmark where we have N=20, 50, 100 annotated examples for fine-tuning RoBERTa-base model. We report mean (and standard deviation) performance over 10 different seeds for each experiment.

5. CONCLUSION

In this paper, we propose a novel learning strategy for text classification tasks. We formulate the supervised contrastive learning problem as a Multi-Objective Optimization problem. The proposed loss function includes both supervised contrastive learning loss and the conventional cross-entropy loss. To solve the optimization problem, we employed two well-known approaches, namely the linear scalarization and the exact Pareto optimal solution search method. We evaluated the proposed method in few-shot learning (20, 50, 100 labeled examples) as well as the full dataset training on several datasets from GLUE benchmark. Empirically, we demonstrate the superior performance of our solution over two competing approaches for fine-tuning RoBERTa-base model. As a future work, we aim to adapt the proposed method for the self-supervised learning setting. We will also extend our approach to different application domains such as computer vision and graph representation learning.

References

[1] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," *arXiv preprint arXiv:2011.01403*, 2020.

[2] Youness Moukafih, Abdelghani Ghanem, Karima Abidi, Nada Sbihi, Mounir Ghogho, and Kamel Smaïli, "Sim-scl: A simple fully-supervised contrastive learning framework for text representation," in *AJCAI 2021-34th Australasian Joint Conference on Artificial Intelligence*, 2022.

[3] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader, "Declutr: Deep contrastive learning for unsupervised textual representations," *arXiv preprint arXiv:2006.03659*, 2020.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[5] Hakim Hafidi, Mounir Ghogho, Philippe Ciblat, and Ananthram Swami, "Graphcl: Contrastive self-supervised learning of graph representations," *arXiv preprint arXiv:2007.08025*, 2020.

[6] Hakim Hafidi, Mounir Ghogho, Philippe Ciblat, and Ananthram Swami, "Negative sampling strategies for contrastive self-supervised learning of graph representations," *Signal Processing*, vol. 190, pp. 108310, 2022.

[7] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18661–18673, 2020.

[8] Tianyu Gao, Xingcheng Yao, and Danqi Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[9] Jean-Antoine Désidéri, "Multiple-gradient descent algorithm (mgda) for multiobjective optimization," *Comptes Rendus Mathematique*, vol. 350, no. 5-6, pp. 313–318, 2012.

[10] Ozan Sener and Vladlen Koltun, "Multi-task learning as multi-objective optimization," *Advances in neural information processing systems*, vol. 31, 2018.

[11] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong, "Pareto multi-task learning," *Advances in neural information processing systems*, vol. 32, 2019.

[12] Debabrata Mahapatra and Vaibhav Rajan, "Exact pareto optimal search for multi-task learning: Touring the pareto front," *arXiv preprint arXiv:2108.00597*, 2021.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.