DATA-CUBE: Data Curriculum for Instruction-based Sentence Representation Learning

Yingqian Min [1]*, **Kun Zhou** [1]*, **Dawei Gao** [3], **Wayne Xin Zhao**[2]†, **He Hu**[1], and **Yaliang Li**[3]
[1]School of Information, Renmin University of China
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Alibaba Group
{yingqianm,hehu}@ruc.edu.cn, francis_kun_zhou@163.com
batmanfly@gmail.com, {gaodawei.gdw,yaliang.li}@alibaba-inc.com

Abstract

Recently, multi-task instruction tuning has been utilized to improve sentence representation learning (SRL). It enables SRL models to generate task-specific representations with the guidance of task instruction, thus exhibiting strong generalization ability on unseen tasks. However, these methods mostly neglect the potential interference problems across different tasks and instances, which may affect the training of the model. To address this issue, we propose a data curriculum method, namely **Data-CUBE**, that arranges the order of all the multi-task data for training, to minimize the interference risks from two aspects. At the task level, we aim to find the optimal task order to minimize the total cross-task interference risk and formulate this problem as the traveling salesman problem, which is further solved by a specially designed simulated annealing algorithm. At the instance level, we propose a measurement method to quantify the difficulty of all instances per task, and then arrange instances in an easy-to-difficult order for training. Experimental results show that our approach can boost the performance of state-of-the-art methods. Our code and data will be publicly released.

1 Introduction

Sentence representation learning (SRL) (Reimers and Gurevych, 2019; Gao et al., 2021) is a fundamental task in the NLP field, which focuses on encoding the semantic information of sentences into low-dimensional vectors. Typically, existing work (Karpukhin et al., 2020; Zhou et al., 2022) collects a set of sentence pairs (or augmented in an unsupervised way), and then learns the model parameters by maximizing and minimizing the similarity scores of relevant and irrelevant sentences, respectively. Previous SRL methods based on advanced language models and learning objectives (Reimers and Gurevych, 2019; Ni et al., 2022a), are capable of producing high-quality representations that perform well on various downstream tasks.

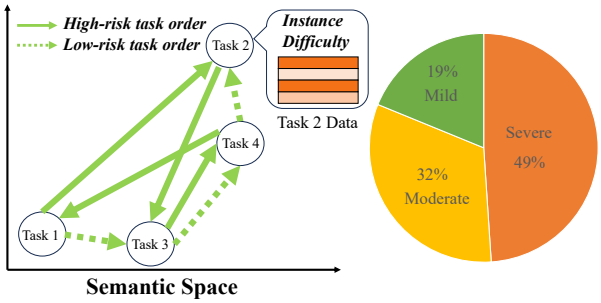

Figure 1: **(left) Example of task- and instance-level interference**. The distance reflects task similarity, and the shades of oranges represent the difficulty level. **(right) The underfitting degrees of all training tasks**. According to the ratio of instances whose positives and negatives are not clearly distinguished (margin<0.05), we categorize all tasks into three degrees: severe (>80%), moderate (>50% but <80%), and mild (<50%).

Despite the success, recent studies (Neelakantan et al., 2022) have revealed that it is challenging to directly transfer the learned sentence representations into new tasks, even causing significant performance degradation. To alleviate this problem, instruction tuning (Wei et al., 2022; Wang et al., 2022b) has been applied to sentence representation learning, which collects a diverse set of sentence-pair datasets with task-specific natural language instructions (Su et al., 2023; Xiao et al., 2023). Before training, each of the collected datasets is generally divided into multiple mini-batches at random (Su et al., 2023), and the SRL model will be trained over the mini-batches of all datasets in a random order. After multi-task training on the dataset collection, the model would become capable of generating task-specific sentence representations with the guidance of the task instruction, exhibiting improved generalization ability on unseen tasks.

However, as the collected datasets vary in data distributions, a random order for instruction data

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics ACL 2024, pages 13748–13761

scheduling would lead to potential *cross-task interference* risk for model optimization. As depicted in Figure 1 (left), when the neighboring mini-batches are from very different tasks, the successive learning of them would lead to conflict in the optimization objective, affecting the final performance of both tasks. In addition, the instances for a given task might be with varied difficulty levels. Randomly assigning them into mini-batches may result in potential *cross-instance interference*, which is also likely to cause the performance degradation. As shown in Figure 1 (right), the competitive model INSTRUCTOR (Su et al., 2023) struggles with distinguishing more than 80% positive and negative examples in almost half of the training datasets, indicating the severe underfitting problem.

To address these issues, in this paper, we propose **DATA-CUBE**, a Data CUrriculum method for instruction-Based sentencE representation learning. The core idea of our approach is to design a proper data curriculum that arranges the orders of all tasks and instances for minimizing the potential interference risks. Concretely, for the *cross-task interference*, we focus on finding the optimal task order where all the neighboring two tasks are as similar as possible, to minimize the total interference risks derived from task divergence. We formulate this task as the the travelling salesman problem (TSP) (Hoffman et al., 2013): all the tasks are regarded as the nodes in a fully connected graph with task similarity as the edge weights, and the optimal order search problem is essentially to find the longest route that visits all nodes. To solve the TSP, we employ the widely-studied simulated annealing algorithm to efficiently find its suboptimal solution. For the *cross-instance interference*, we measure the discriminability of positive and negative, as the estimated difficulty for sorting all the instances. Then, we divide them into *easy-to-difficult* mini-batches for training, to minimize the interference risks caused by varied instance difficulty.

To integrate the two strategies, we first find the optimal *task order* with the simulated annealing algorithm, then sort the instances within each task according to their difficulty to obtain the *instance order* for mini-batch arrangement, and finally divide the sorted instances from all the tasks into a sequence of mini-batches for training SRL models.

Our contributions are summarized as follows:

(1) To our knowledge, the proposed Data-CUBE is the first attempt of data curriculum in instruction-based SRL. It is a model- and data-agnostic approach for improving the training of SRL models.

(2) We reveal the interference problem in training instruction-based SRL, and propose to address cross-task interference by formulating it as a TSP and address cross-instance interference by employing an easy-to-difficult data curriculum.

(3) Extensive experiments on downstream tasks show the effectiveness of our approach, outperforming a number of competitive SRL models.

2 Related Work

Sentence Representation Learning. A robust sentence representation plays a pivotal role in diverse downstream tasks. Previously, most sentence representation models concentrate on a singular task or domain, weak in transferring to other downstream tasks without further fine-tuning. For instance, SimCSE (Gao et al., 2021), SBERT (Reimers and Gurevych, 2019), and DCLR (Zhou et al., 2022) are trained to address sentence similarity and classification tasks, while models like DPR (Karpukhin et al., 2020), Contriever (Izacard et al., 2022a), Master (Zhou et al., 2023), and GTR (Ni et al., 2022b) are applied to information retrieval. In response to this challenge, recent efforts have emerged to develop instruction-based sentence representation models through multi-task contrastive learning. Exemplars include INSTRUCTOR (Su et al., 2023), BGE (Xiao et al., 2023), and GTE (Li et al., 2023), which aim to enhance the adaptability and generalization capabilities of sentence representations across diverse tasks and domains. Existing studies primarily focus on aspects such as the training objective, model architecture, or training scale, while paying limited attention to the challenges posed by interference during the multi-task training process.

Instruction Tuning. Instruction tuning (Ouyang et al., 2022; Zhao et al., 2023) involves supervised fine-tuning pre-trained language models by integrating well-formatted natural language instructions into the input. This process is closely connected to multi-task learning and is believed to enhance the generalization capability of language models across a range of tasks (Wei et al., 2022). Previous studies have demonstrated that increasing the number and diversity of tasks associated with instructions can improve performance. Considering the effectiveness of instruction tuning, it has

been applied to various NLP tasks, such as sentence representation learning (Su et al., 2023; Xiao et al., 2023). However, with the increasing diversity of tasks, there is a potential for interference across different tasks, which may lead to performance degradation (Mueller et al., 2022). Hence, we propose to leverage data curriculum to alleviate the interference in the multi-task instruction tuning.

Traveling Salesman Problem (TSP). Traveling salesman problem is a classic combinatorial optimization problem in computer science and operations research. It is to find the shortest route for a salesman to visit a given set of cities exactly once and return to the start (Hoffman et al., 2013; Cheikhrouhou and Khoufi, 2021). Recognized as an NP-hard problem, TSP is costly to solve for large numbers of cities. Consequently, numerous heuristic methods have been developed to find near-optimal solutions efficiently (Helsgaun, 2006; Matai et al., 2010). A notable method is **S**imulated **A**nnealing (SA) (Bertsimas and Tsitsiklis, 1993), an algorithm inspired by metallurgy's annealing process. SA is effectively used in various combinatorial problems like TSP, Job Shop Scheduling Problem (Chakraborty and Bhowmik, 2015), and Graph Coloring Problem (Pal et al., 2012).

3 Preliminary

3.1 Task Definition

Sentence representation learning (SRL) is to train a capable text encoder that can map a sentence into a latent vector for downstream tasks. To enhance the generalization ability on unseen tasks of the SRL model, instruction based SRL models (Su et al., 2023) take as input the sentence s with a natural language instruction I, to obtain the task-aware sentence representation $\mathbf{v}$. To train the SRL model, we are given m instruction formatted sentence-pair datasets $\mathcal{D} = \{d_i\}_{i=1}^{m}$, where d_i denotes the i-th dataset and corresponds to the task o_i. Each dataset typically consists of n queries $\{q_j\}_{j=1}^{n}$ and their relevant sentences $\{s_j^{(+)}\}_{j=1}^{n}$ and irrelevant sentences $\{s_j^{(-)}\}_{j=1}^{n}$, with specific instructions $\langle I^{(q)}, I^{(+)}, I^{(-)}\rangle$ for the three text types.

During training, the model follows a certain task order $\mathcal{O} = \{o_i\}$ and instance order, typically random orders, to learn the model parameters. However, random data training would potentially lead to potential learning interference issues as discussed in Section 1. In this work, we aim to devise a data curriculum approach to improve the multi-task training for SRL, to reduce the interference risk.

3.2 Travelling Salesman Problem

Considering the interference problem across the m datasets, we first estimate the mutual interference risks between every two tasks $r(i, j)$, and then find the optimal order for all the tasks $\mathcal{O} = \{o_i\}_{i=1}^{m}$, to minimize the accumulated interference risk as:

$$\arg\min \sum_{i=1}^{m-1} r(o_i, o_{i+1}) + r(o_m, o_1) \qquad (1)$$

where o_i is the corresponding task of the dataset d_i from $\mathcal{D}$. Such an optimal order search problem can be converted as the traveling salesman problem (TSP) that finds the shortest route to visit all cities (*i.e.,* tasks) exactly once.

As TSP is proved to be an NP-hard problem, heuristic algorithms (Helsgaun, 2006; Matai et al., 2010) have been widely studied to find suboptimal solutions in a reasonable time. Simulated annealing (SA) (Bertsimas and Tsitsiklis, 1993) is a commonly used algorithm for TSP, and its basic idea is to start with an initial solution and then search by randomly perturbing the solution. In each iteration, the algorithm evaluates the quality of the new solution by computing the change in the objective function. If the new solution is better, it will replace the current one. Otherwise, the update would occur according to a probability calculated based on the temperature and the change in the objective function. As the iterations progress, the temperature decreases, gradually reducing the likelihood of accepting worse solutions and guiding the algorithm towards converging.

4 Approach

In this section, we present the proposed **DATACUBE**, a Data CUrriculum method for instruction-Based sentencE representation learning. Following previous work, we develop the approach based on a multi-task contrastive learning framework, and while introduce novel data curriculum methods for SRL, considering both task-level and instance-level arrangement, which can significantly reduce the learning interference issue.

4.1 Multi-task Contrastive Learning

We employ multi-task contrastive learning to train a pre-trained language model (*e.g.,* T5 (Raffel et al., 2020)) for producing sentence representations. In

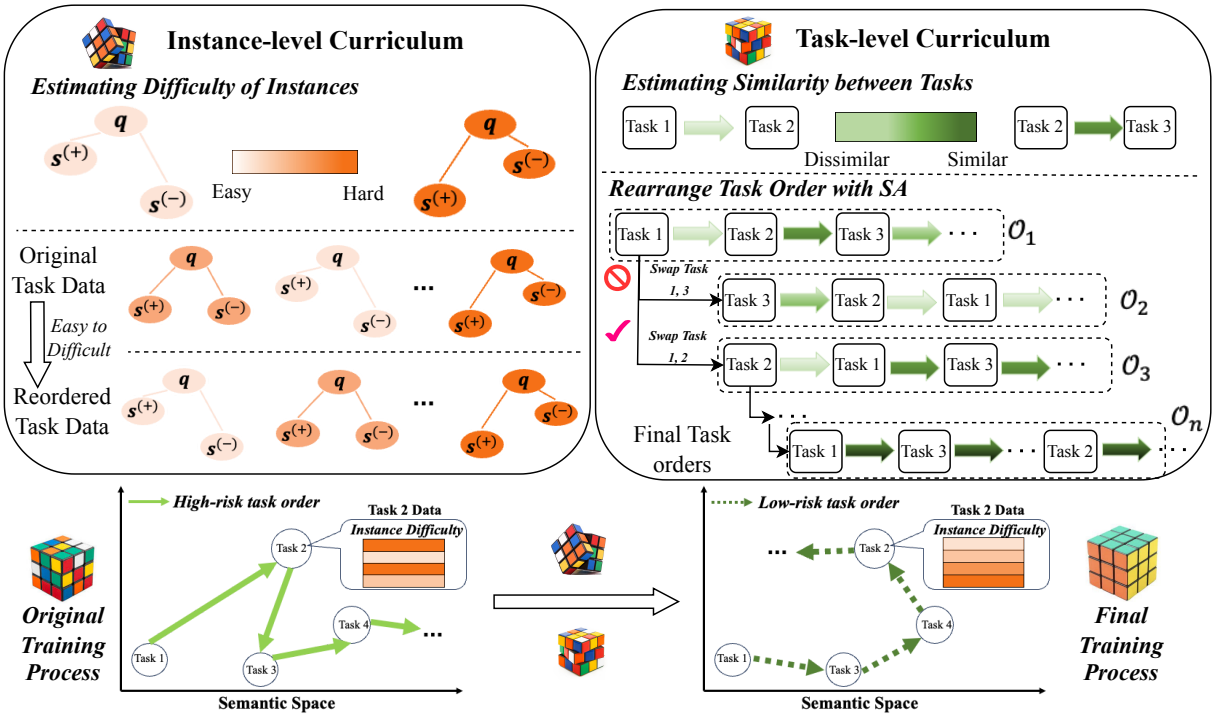

Figure 2: An overall illustration of Data-CUBE: the task-level curriculum rearranges the task orders from similar to dissimilar using the simulated annealing algorithm, and the instance-level curriculum reorganizes the instances within each task based on an easy-to-difficult order.

general, it maximizes the similarity of positive pairs $\langle q, s^{(+)} \rangle$ and minimizes the one of negative pairs $\langle q, s^{(-)} \rangle$, based on specific task instructions $\langle I^{(q)}, I^{(+)}, I^{(-)} \rangle$. Concretely, we first preprocess the collected datasets into multiple mini-batches with specific instructions and then optimize the model parameters via a multi-task learning loss.

For each dataset, we concatenate its contained queries, positive and negative sentences with corresponding instructions, to compose new instances:

$$\tilde{q} = [I^{(q)}; q], \tilde{s}^{(+)} = [I^{(+)}; s^{(+)}], \tilde{s}^{(-)} = [I^{(-)}; s^{(-)}], \quad (2)$$

where the instruction contains the description that specifies the task, *e.g.,* "*Represent the example for the following task: Given a scientific question, generate a correct answer to it*". Next, we perform the mini-batch splitting, and guarantee that all the in-batch instances come from the same task. Such a way avoids the possible cross-task interference when using in-batch negatives for contrastive learning. Thus, we leverage the following loss function:

$$\mathcal{L} = \sum_{i=1}^{m} \sum_{\mathcal{B} \in d_i} \sum_{j=1}^{|\mathcal{B}|} \frac{e^{\text{sim}(\mathbf{v}_{\tilde{q}_j}, \mathbf{v}_{\tilde{s}_j}^{(+)})/\tau}}{\sum_{k=1}^{|\mathcal{B}|} e^{\text{sim}(\mathbf{v}_{\tilde{q}_j}, \mathbf{v}_{\tilde{s}_k})/\tau}} \quad (3)$$

where $\mathcal{B} = \{\langle \tilde{q}_j, \tilde{s}_j^{(+)}, \tilde{s}_j^{(-)} \rangle\}_{j=1}^{|\mathcal{B}|}$ denotes the mini-batch of $|\mathcal{B}|$ instances from dataset d_i, $\mathbf{v}_{\tilde{q}_j}$ and $\mathbf{v}_{\tilde{s}_j}^{(+)}$ refer to the representations of the j-th query $\tilde{q}_j$ and positive sentence $\tilde{s}_j^{(+)}$ respectively, m denotes the number of datasets and τ denotes the temperature, and $\text{sim}(\cdot, \cdot)$ is the cosine similarity function.

Here, we adopt a similar setting in loss function (Eq. (3)) as previous study (Su et al., 2023), while our focus is to design a suitable data curriculum approach for scheduling the mini-batches from all the task datasets. In what follows, we will introduce the proposed task-level (Section 4.2) and instance-level (Section 4.3) curriculum methods in detail.

4.2 Task-level Curriculum Arrangement: From Similar to Different

As the divergence of data distributions between neighboring tasks may affect the learning of both (Ding et al., 2023), we expect that the training order can be a *"smooth"* transition across tasks, to minimize the accumulated cross-task interference risk. Thus, we estimate the cross-task interference risk based on task similarity, and then search the optimal order by the simulated annealing algorithm.

4.2.1 Cross-task Interference Risk Estimation

To estimate the cross-task interference risk, we adopt the similarity of text representations for measuring the divergence in data distribution. Con-

cretely, we randomly sample n_t queries per dataset to compose a representative subset, then compute the mean representation as the task representation:

$$\mathbf{v}^{(t)} = \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbf{v}_{\tilde{q}_j}. \quad (4)$$

Here, we use a pre-learned model (*i.e.,* Instructor (Su et al., 2023)) to produce the query representation. Based on it, the task similarity can be measured using the cosine similarity of task representations. As similar tasks typically have lower interference risk (Mueller et al., 2022), we can roughly estimate the cross-task interference risk as:

$$r(i,j) \propto -\text{sim}(\mathbf{v}_i^{(t)}, \mathbf{v}_j^{(t)}), \quad (5)$$

where $\text{sim}(\mathbf{v}_i^{(t)}, \mathbf{v}_j^{(t)})$ is the cosine similarity between the representations of the i-th and j-th tasks. With these estimated interference scores, we next study how to schedule different tasks to reduce the entire risk across the data curriculum.

4.2.2 Optimal Order Search

As discussed in Section 3.2, we can formulate optimal order search as TSP over in a fully connected undirected graph, in which tasks are considered as nodes and the estimated interference risk (Eq. 5) between two linked tasks are considered as edge weight. In this way, our goal becomes how to find the shortest route that visits each node exactly once, with the objective function as Eq. 1.

According to the negative correlation between interference risk and task similarity as Eq. 5, the risk minimization objective is equivalent to maximizing the sum of neighboring task similarity:

$$\arg\max \sum_{i=1}^{m} \text{sim}(\mathbf{v}_i^{(t)}, \mathbf{v}_{i+1}^{(t)}) + \text{sim}(\mathbf{v}_m^{(t)}, \mathbf{v}_1^{(t)}). \quad (6)$$

Therefore, our goal is to find the most smooth transition path for all the tasks, to avoid the drastic distribution shift of the neighboring tasks.

To solve TSP, we adopt the simulated annealing algorithm to find a suboptimal solution within a reasonable amount of time. Specially, simulated annealing iteratively perturbs the current solution to explore the solution space, and accepts the new solution based on the objective in Eq. 6 and a gradually decaying temperature τ_s. Concretely, we first initialize a task order $\mathcal{O}'$ by random shuffling. Next, we repeat the perturb-then-check process until convergence. In each iteration, we randomly choose a pair of tasks in the current order $\mathcal{O}'$, swap their positions to obtain the new order $\tilde{\mathcal{O}}'$, and check whether the total neighboring task similarity will increase. If increased, the new order will replace the current one. Otherwise, the new order will be accepted in a probability as:

$$p(\mathcal{O}', \tilde{\mathcal{O}}', \tau_s) = \exp(-\frac{\Delta(\mathcal{O}', \tilde{\mathcal{O}}')}{\tau_s}) \quad (7)$$

where $\Delta(\mathcal{O}', \tilde{\mathcal{O}}')$ denotes the difference of the total neighboring task similarity between the current and new orders using Eq. 6. Such a way prevents the solution from being stuck at a local minimum, and incorporating "Δ" also reduces the likelihood of accepting worse solutions. it also reduces the instability close to the converged suboptimal point.

4.3 Instance-level Curriculum Arrangement: From Easy to Difficult

In addition to the task-level curriculum, we also devise the instance-level curriculum, to reduce the *cross-instance* interference risk. The basic idea is to first estimate the varying difficulty of instances and then reorder the instances in each task from easy to difficult. Next, we detail the two steps.

4.3.1 Instance Difficulty Estimation

As the tasks for SRL mainly focus on distinguishing the relevant sentence $\tilde{s}^{(+)}$ and irrelevant sentence $\tilde{s}^{(-)}$ according to the query $\tilde{q}$, we leverage the discriminability of $\tilde{s}^{(+)}$ and $\tilde{s}^{(-)}$ to measure the instance difficulty.

Specially, the positive and negative sentences of easy instances would be clearly distinguished by an SRL model trained on the data, while the ones of difficult instances would pose more challenges for successful discrimination. We employ a pre-learned model (*i.e.,* Instructor) to encode the representations of the positive and negative pairs, then estimate the instance difficulty by computing the similarity difference as:

$$\phi(\tilde{q}, \tilde{s}^{(+)}, \tilde{s}^{(-)}) = \text{sim}(\mathbf{v}_{\tilde{q}}, \mathbf{v}_{\tilde{s}^{(+)}}) - \text{sim}(\mathbf{v}_{\tilde{q}}, \mathbf{v}_{\tilde{s}^{(-)}}). \quad (8)$$

The smaller the difference is, the more likely the model struggles with distinguishing the positive and negative, which indicates a difficult instance.

4.3.2 Instance Curriculum Arrangement

According to the difficulty measurement in Eq. 8, we can assign the estimated scores to the instances for a given task. Then, we sort all the instances

Model	BIO	S-R	S12	S13	S14	S15	S16	S17	S22	S-B	Avg.
Sentence Representation APIs											
OpenAI-TE	86.35	80.60	69.80	83.27	76.09	86.12	85.96	90.25	68.12	83.17	80.97
Voyage	84.85	79.71	77.09	88.91	82.08	89.21	84.74	90.73	62.10	**89.86**	82.93
Cohere	85.01	82.18	77.62	85.16	80.02	88.92	**86.92**	90.09	66.81	88.79	83.15
Ember	85.81	81.75	78.51	86.62	83.06	88.39	86.82	87.90	66.76	87.77	83.34
No-Instruction Sentence Representation Models											
GloVe	44.93	55.43	54.64	69.16	60.81	72.31	65.34	77.95	56.35	61.54	61.85
USE	78.19	74.43	72.58	72.22	69.98	82.22	76.91	85.22	61.90	80.28	75.39
Contriever	83.32	70.20	64.34	80.03	74.51	83.30	79.67	86.32	64.64	78.81	76.51
GTR	81.91	74.29	70.12	82.72	78.24	86.26	81.61	85.18	65.76	77.73	78.38
SimCSE	68.38	80.77	75.30	84.67	80.19	85.40	80.82	89.44	61.96	84.25	79.12
SGPT	79.50	79.59	74.29	85.35	79.21	85.52	82.54	90.44	63.20	85.67	80.53
E5	84.73	80.49	75.93	85.22	80.54	88.81	85.28	89.37	62.99	87.21	82.06
SentenceT5	80.43	80.47	78.85	**88.94**	84.86	89.32	84.67	89.46	65.33	84.01	82.63
Instruction-based Sentence Representation Models											
Jina	84.43	79.2	74.52	83.16	78.09	86.91	83.65	90.16	64.88	84.60	80.96
Udever	85.52	81.41	77.47	86.38	81.17	88.23	86.29	90.62	65.01	88.02	83.01
Stella	85.94	81.06	78.72	84.88	83.11	88.74	86.35	87.71	66.28	87.45	83.02
BGE	84.65	81.68	**79.05**	86.37	82.78	88.03	86.49	87.5	67.05	87.52	83.11
GTE	88.65	79.81	76.81	88.11	82.66	88.93	84.25	88.47	**69.71**	86.07	83.35
INS	84.39	81.27	76.28	88.18	81.92	89.01	85.49	90.30	67.74	86.88	83.15
+Data-Cube	**89.37**	**82.52**	78.46	88.39	**83.06**	**89.46**	85.87	**91.08**	68.28	87.61	**84.41**

Table 1: Sentence representation performance on 10 STS tasks (Spearman's correlation on the English test set) in MTEB (Muennighoff et al., 2023). We choose diverse models as baselines, including traditional no-instruction sentence representation models, instruction-based sentence representation models, and sentence representation APIs. In the case of models with multiple versions (*e.g.,* varying parameter scales), we opt for the version that demonstrates superior performance. All reported results are derived from the MTEB Leaderboard. We employ bold numbers to emphasize the best results obtained on each dataset.

in each task descendingly, and further divide them into multiple mini-batches $\mathcal{B}$. In this way, we can obtain the easy-to-difficult mini-batches per task. Compared with randomly sampling, our method can alleviate the interference caused by varying instance difficulty within each mini-batch and the difficulty divergence in neighboring mini-batches.

In addition, we find that too difficult instances may not always be useful, as they could potentially introduce noise into data. Therefore, following existing work (Zhou et al., 2022), we design a binary mask α_i using a threshold δ to reduce its influence as:

$$\alpha_i = \begin{cases} 0, & \phi(\tilde{q}_i, \tilde{s}_i^{(+)}, \tilde{s}_i^{(-)}) \geq \delta \\ 1, & \phi(\tilde{q}_i, \tilde{s}_i^{(+)}, \tilde{s}_i^{(-)}) < \delta \end{cases} . \quad (9)$$

Then, we apply the mask to the contrastive loss of each instance in Eq. 3. This ensures that noisy instances cannot be directly learned but serve as in-batch negatives for other instances.

5 Experiments

In this section, we train with DATA-CUBE based on INSTRUCTOR and conduct evaluations on four task categories within MTEB (Muennighoff et al., 2023), encompassing a total of 28 downstream tasks. Furthermore, we continue to investigate the effectiveness and robustness of DATA-CUBE. For detailed settings, see Appendix B and C.

5.1 Main Results

We present the main experiment results in Table 1. Based on the results, it is evident that instruction-based sentence representation models generally perform better than no-instruction models, although some no-instruction models are much larger than instruction-based models in terms of scale (*e.g.,* Sentence-T5 XXL has 11B parameters while the largest version of BGE is about 300M). A potential reason is that instruction tuning enhances the models' understanding of tasks by integrating natural language instructions. This integration assists

Task Type	Datasets	INS	+Data-CUBE
Reranking	AUDQ	64.30	**64.74**
	SODQ	52.17	51.96
	SDRR	82.00	**82.82**
	MSR	31.68	**31.73**
	Avg.	57.53	**57.81**
Clustering	ACP2P	43.16	**43.76**
	ACS2S	32.56	**33.25**
	BCP2P	37.62	**37.63**
	BCS2S	31.33	31.06
	MCP2P	34.22	33.98
	MCS2S	32.00	30.89
	RC	64.65	63.56
	RCP2P	64.63	**65.31**
	SEC	68.78	**70.23**
	SECP2P	36.15	35.59
	TNC	54.13	**55.82**
	Avg.	45.29	**45.55**
Pair Classification	SDQ	93.07	**93.32**
	TSE	77.42	**78.69**
	TUC	87.18	86.73
	Avg.	85.89	**86.25**

Table 2: Sentence representation performance on reranking, clustering, and pair classification tasks.

Settings	BIO	S12	S14	S22	Avg.
Data-CUBE	**89.37**	**78.46**	**83.06**	**68.28**	**84.41**
w/o Inst	87.53	77.64	82.85	66.95	83.94
w/o Task	86.56	78.15	82.53	67.48	83.94
Vanilla	88.16	77.10	82.31	64.47	83.42

Table 3: Ablations of the two-level curriculum.

the models in encoding sentences into task-aware representations, thereby providing a significant advantage in downstream tasks.

Among all the compared models, our method achieves the highest average performance across STS tasks. Although it may not always rank first in certain tasks like STS-12, STS-13, STS-16, STS-22, and STSBenchmark, it maintains competitive results. Notably, models achieving the best results in these tasks tend to excel in only one task but underperform in others. In contrast, our approach consistently demonstrates effectiveness across all tasks. When compared to our backbone model (*i.e.,* INSTRUCTOR), our method significantly improves the performance on all STS tasks. In addition, we also evaluate our method on other task categories (*e.g.,* Reranking, Clustering, and PairClassifiacation). As Table 2 shows, our method achieves average performance gains on these diverse task categories. This implies that our approach plays a significant role in reducing the interference among

Settings	BIO	S12	S14	S22	Avg.
Ours	**89.37**	**78.46**	**83.06**	**68.28**	**84.41**
800K	88.30	77.78	83.00	67.17	83.99
3M	88.40	77.71	83.06	68.12	84.10
5M	87.83	78.01	83.14	66.46	84.03

Table 4: Variation studies of the iterations of Simulated Annealing algorithm on the test set of STS tasks.

Settings	BIO	S12	S14	S22	Avg.
Ours	**89.37**	**78.46**	**83.06**	**68.28**	**84.41**
8	84.82	76.44	82.23	66.61	83.23
16	87.37	78.23	83.35	68.76	84.21
32	86.70	77.50	83.00	68.38	84.03

Table 5: Variation studies of different batch sizes on the test set of STS tasks.

tasks, which enhances not only the performance in specific tasks but also the overall versatility.

It is noteworthy that both the volume of training data and the mini-batch size utilized in our approach are considerably smaller compared to other robust instruction-based sentence representation models. Specifically, our model is trained with only 1 million sentence pairs and a batch size of 64, in stark contrast to models like BGE which use 300 million sentence pairs and a batch size of 32768. This indicates that our approach has the potential to enhance the model's ability to learn more effectively from multi-task data in the context of data interference and achieve comparable or even superior performance despite limited data and computational resources.

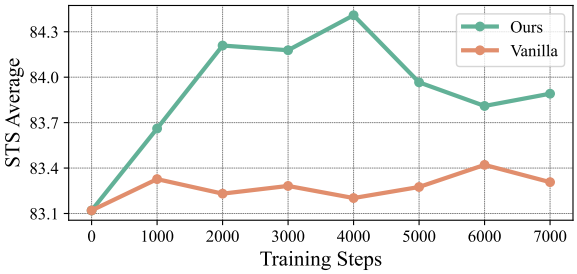

Figure 3: Performance fluctuation curve on the STS tasks during the training process.

5.2 Further Analysis

Next, we continue to investigate the effectiveness and robustness of Data-Cube. This involves conducting ablation studies on the two-level curriculums, assessing the impact of iterations in Simulated Annealing, analyzing the influence of mini-batch size during training, and thoroughly exam-

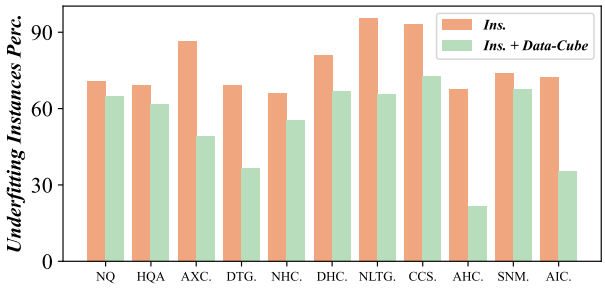

Figure 4: The percentage of underfitting instances within different tasks. We show the comparison between INSTRUCTOR and fine-tuned with Data-CUBE.

ining the training convergence. To gain a better understanding of the variations between different settings, we carefully select several tasks in STS, such as BIOSSES (Soganciogl u et al., 2017), STS-12, 14, and 22 (Agirre et al., 2012, 2014; Chen et al., 2022), where our method demonstrates more noticeable improvements.

Ablations of Two-level Curriculum. To explore the influence of task-level curriculum, we exclusively implement the instance-level curriculum to reorganize the training data, sorting instances within each task by difficulty using Eq. 8 while randomly shuffling task orders. Conversely, we employ the task orders generated by the task-level curriculum but randomly shuffle the instances within each task to assess the effectiveness of the instance-level curriculum. Furthermore, we train our backbone model without extra operations as the vanilla baseline. As Table 3 shows, both two-level curriculums contribute to alleviating the interference of multi-task data and enhancing the performance of the sentence representation model.

Different Task-level Curriculum Methods. In the task-level curriculum, we calculate the similarity between tasks by meaning the representations of the sampled queries per task using Eq. 4 and Eq. 5 and utilize the SA algorithm to search the suboptimal results. To validate the effectiveness of our methods, we conduct experiments using other variations. Concretely, we change the way of task-level similarity measurement and optimization objectives. The detailed settings are as follows: **(1) Mean-Q-ArgMax (MQMax, Ours):** We use the task vector with Eq. 4 and estimate the cross-task interference risk using Eq. 5. When arranging the task order, we maximize the total task similarity. **(2) Tf-idf-Argmax (TIMax)**: We use the Tf-idf (Salton and Buckley, 1988) vector of each task and calculate the cosine similarity to estimate the cross-task interference risk. When arranging the task order, we maximize the total task similarity. **(3) Mean-Q-Argmin (MQMin):** We use the task vector with Eq. 4 and estimate the cross-task interference risk using Eq. 5. When arranging the task order, we minimize the total task similarity. **(4) Mean-Q-Random (MQRand):** We use the task vector with Eq. 4 and estimate the cross-task interference risk using Eq. 5. When arranging the task order, we shuffle the task order at random. As shown in Table 6, our approach can also perform better than all the other variations, indicating the effectiveness of using mean query representation for measuring the task similarity and maximizing the total task similarity.

Settings	BIO	S12	S14	S22	Avg.
MQMax*	**89.37**	**78.46**	**83.06**	**68.28**	**84.41**
TIMax	88.21	77.98	82.83	68.16	84.09
MQMin	87.28	76.81	81.89	67.53	83.45
MQRand	86.56	78.15	82.53	67.48	83.94

Table 6: Variation studies of different task-level curriculum methods. "*" indicates our method.

Settings	BIO	S12	S14	S22	Avg.
E2D*	**89.37**	**78.46**	**83.06**	**68.28**	**84.41**
D2E	87.68	76.41	81.93	68.06	83.73
Random	87.53	77.64	82.85	66.95	83.94

Table 7: Variation studies of different instance-level curriculum methods. "*" indicates our method.

Different Instance-level Curriculum Methods. In the instance-level curriculum, we leverage the Eq. 8 for instance difficulty estimation and reorder instances per task from easy to difficult along the thought of curriculum learning. To validate the effectiveness of our ordering method, we rearrange the order of instances to create two variations of our methods for comparison. The detailed settings are as follows: **(1) Easy-to-Difficult (E2D, Ours):** We arrange instances per task in an easy-to-difficult order. **(2) Difficult-to-Easy (D2E):** We arrange instances per task in a difficult-to-easy order. **(3) Random:** We randomly shuffle instances per task using seed 42. As shown in Table 7, our approach consistently outperforms all the variations, indicating the effectiveness of our designed Easy-to-Difficult order.

Iterations of Simulated Annealing. In the task-level curriculum, we employ Simulated Annealing

algorithm that gradually obtains an approximate solution as the iterations progress. In broad terms, a higher number of iterations typically leads to a more optimal solution. Consequently, we undertook experiments on task orders generated through varying iteration counts to illustrate the adequacy of the specific iteration count we employed. We opt to compare the results of using task orders generated by SA at 800K, 2M (Ours), 3M, and 5M iterations. As Table 3 shows, these performances are comparable, indicating that the chosen iteration step of 2M is adequate for alleviating interference.

Size of Mini-batch. During multi-task contrastive learning, we leverage the in-batch negatives to extend the positive-negative ratio, which has been shown to enhance the uniformity of the sentence representation model and thereby improve overall performance (Karpukhin et al., 2020) In the instance-level curriculum, we propose to alleviate the interference between instances with different difficulty. However, when using a larger mini-batch, the variability in difficulty within the batch increases, seemingly conflicting with our curriculum design. To address this, we conduct experiments to evaluate how the instance-level curriculum affects model performance across different mini-batch sizes. As shown in Table 5, while larger mini-batches generally lead to better performance, our Data-CUBE results remain comparable even with smaller batch sizes. Notably, the model performs better with a batch size of 16 compared to 32, indicating that our approach is particularly effective in resource-constrained scenarios.

Analysis of Training Convergence. To validate the efficacy of our approach in mitigating data interference, we assess the model's performance on STS tasks throughout the training process (See Figure 3). In comparison to directly continuing training, employing Data-CUBE leads to significantly improved performance in fewer training steps. Furthermore, we compare the ratio of underfitting instances within tasks before and after training with Data-CUBE (See Figure 4). It is evident that the ratio of underfitting instances consistently decreases across various tasks. It indicates the effectiveness of our method in alleviating interference for multi-task contrastive learning and results in a more powerful and robust sentence representation model.

6 Conclusion

In this work, we proposed Data-CUBE, a data curriculum method for multi-task instruction based sentence representation learning. The key idea of our approach is to reduce the cross-task and cross-instance interference risks by using a more suitable data curriculum of instances for training. To achieve this, we employed a simulated annealing algorithm to find the optimal task order to minimize the cross-task interference, and assigned all instances per task into easy-to-difficult mini-batches to reduce the cross-instance interference. Experimental results on MTEB sentence representation evaluation tasks have shown that our approach can boost the performance of state-of-the-art baselines.

Limitations

Although Data-CUBE is a model-agnostic and data-agnostic approach to enhancing instruction-based sentence representation models, due to the lack of experimental details, we have yet not employed it on the state-of-the-art models. Actually, the improvement of integrating our approach on Instructor is able to indicate its effectiveness. We will conduct more experiments using our approach on other strong baselines in the future. Furthermore, due to our limited computational resources, we have not explored our method on larger models, *e.g.,* 3B and 7B LLaMA. We also leave it in our future work, and explore more efficient and effective data curriculum methodology for large-scale datasets and models.

Acknowledgement

This work was partially supported by National Natural Science Foundation of China under Grant No. 62222215, Beijing Natural Science Foundation under Grant No. 4222027 and L233008. Xin Zhao is the corresponding author.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 252–263. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 81–91. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 497–511. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 385–393. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 32–43. Association for Computational Linguistics.

Dimitris Bertsimas and John Tsitsiklis. 1993. Simulated annealing. *Statistical science*, 8(1):10–15.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14. Association for Computational Linguistics.

Shouvik Chakraborty and Sandeep Bhowmik. 2015. An efficient approach to job shop scheduling problem using simulated annealing. *International Journal of Hybrid Information Technology*, 8(11):273–284.

Omar Cheikhrouhou and Ines Khoufi. 2021. A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy. *Comput. Sci. Rev.*, 40:100369.

Xi Chen, Ali Zeynali, Chico Q. Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw A. Grabowicz, Scott A. Hale, David Jurgens, and Mattia Samory. 2022. Semeval-2022 task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation, SemEval@NAACL 2022, Seattle, Washington, United States, July 14-15, 2022*, pages 1094–1106. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2270–2282. Association for Computational Linguistics.

Chuntao Ding, Zhichao Lu, Shangguang Wang, Ran Cheng, and Vishnu Naresh Boddeti. 2023. Mitigating task interference in multi-task learning via explicit task routing with non-learnable primitives. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 7756–7765. IEEE.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. 2021. TWEAC: transformer with extendable QA agent classifiers. *CoRR*, abs/2104.07081.

Michael Günther, Louis Milliken, Jonathan Geuter, Georgios Mastrapas, Bo Wang, and Han Xiao. 2023. Jina embeddings: A novel set of high-performance sentence embedding models. *CoRR*, abs/2307.11224.

Keld Helsgaun. 2006. *An effective implementation of K-opt moves for the Lin-Kernighan TSP heuristic*. Ph.D. thesis, Roskilde University. Department of Computer Science.

Karla L Hoffman, Manfred Padberg, Giovanni Rinaldi, et al. 2013. Traveling salesman problem. *Encyclopedia of operations research and management science*, 1:1573–1578.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin,

and Edouard Grave. 2022b. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1224–1234. Association for Computational Linguistics.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi S. Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1279–1289. The Association for Computational Linguistics.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *CoRR*, abs/2308.03281.

Xueqing Liu, Chi Wang, Yue Leng, and ChengXiang Zhai. 2018. Linkso: a dataset for learning to retrieve similar question answer pairs on software development forums. In *Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering, NL4SE@ESEC/SIGSOFT FSE 2018, Lake Buena Vista, FL, USA, November 4, 2018*, pages 2–5. ACM.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Rajesh Matai, Surya Prakash Singh, and Murari Lal Mittal. 2010. Traveling salesman problem: an overview of applications, formulations, and solution approaches. *Traveling salesman problem, theory and applications*, 1(1):1–25.

David Mueller, Nicholas Andrews, and Mark Dredze. 2022. Do text-to-text multi-task learners suffer from task conflict? In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2843–2858. Association for Computational Linguistics.

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *CoRR*, abs/2202.08904.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2006–2029. Association for Computational Linguistics.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training. *CoRR*, abs/2201.10005.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022a. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1864–1874. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022b. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9844–9855. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Anindya Jyoti Pal, Biman Ray, Nordin Zakaria, and Samar Sen Sarma. 2012. Comparative performance of modified simulated annealing with simple simulated annealing for graph coloring problem. *Procedia Computer Science*, 9:321–327.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.*, 24(5):513–523.

Darsh J. Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1056–1063. Association for Computational Linguistics.

Gizem Soganciogl u, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinform.*, 33(14):i49–i58.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1102–1121. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022a. Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5085–5109. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3597–3606. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 1–11. The Association for Computer Linguistics.

Xin Zhang, Zehan Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, and Min Zhang. 2023. Language models are universal embedders. *CoRR*, abs/2310.08232.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Kun Zhou, Xiao Liu, Yeyun Gong, Wayne Xin Zhao, Daxin Jiang, Nan Duan, and Ji-Rong Wen. 2023. MASTER: multi-task pre-trained bottlenecked masked autoencoders are better dense retrievers. In *Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, Turin, Italy, September 18-22, 2023, Proceedings, Part II*, volume 14170 of *Lecture Notes in Computer Science*, pages 630–647. Springer.

Kun Zhou, Beichen Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Debiased contrastive learning of unsupervised sentence representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6120–6130. Association for Computational Linguistics.

Appendix

A The process of training with DATA-CUBE

Algorithm 1: Training with Data-CUBE

Input : Original Data $\mathcal{D} = \{d_i\}$, the temperature τ, cooling rate α, and max iterations N of SA, and the backbone model

Output : Instruction-tuned model M'

// Task Curriculum
Initialize a random order $\mathcal{O}' = \{o_i\}$
for i *in* $range(N)$ **do**
  $\tilde{\mathcal{O}}' \leftarrow$ Swap a random pair of tasks in $\mathcal{O}'$
  Calculate $\Delta(\mathcal{O}, \mathcal{O}')$ using Eq. 6
  if $\Delta(\mathcal{O}', \tilde{\mathcal{O}}') > 0$ **then**
    $\mathcal{O}' \leftarrow \tilde{\mathcal{O}}'$
  else if *rand()* $< p(\mathcal{O}', \tilde{\mathcal{O}}', \tau_s)$ **then**
    $\mathcal{O}' \leftarrow \tilde{\mathcal{O}}'$
  end
  $\tau_s \leftarrow \alpha \cdot \tau_s$
end
// Instance Curriculum
for d_i *in* $\mathcal{D}$ **do**
  Calculate $\phi(\tilde{q}, \tilde{s}^{(+)}, \tilde{s}^{(-)})$ of each instance in d_i using Eq. 8
  Arrange d_i to d_i' in descending order based on ϕ
end
// Combine Two-level Curriculum
Initialize an empty $\mathcal{D}'$
for o_i *in* $\mathcal{O}$ **do**
  Choose dataset d_i' corresponding to o_i
  Select the first batch $\mathcal{B}$ of d_i'
  $\mathcal{D}'.append(\mathcal{B})$
  $d_i'.remove(\mathcal{B})$
end
// Multi-task Contrastive training
Use $\mathcal{D}'$ to train the backbone model with Eq. 3
Get the final instruction-tuned model

B Training Settings

Training Dataset. We opt for the multi-task sentence-pair dataset MEDI (Su et al., 2023), comprising 330 sub-datasets spanning various tasks and domains, with a total of 1.4 million instances for training. Each sub-task is accompanied by corresponding natural language instructions elucidating its detailed goal or description.

Training Details To arrange the task-level curriculum, we utilize the simulated annealing algorithm and early stop at approximately 2 million steps to obtain a suboptimal task order $\mathcal{O}$. To rearrange the instance-level curriculum, we calculate the difficulty of instances (ϕ in Eq. 8) in advance and sort all the instances in each task by descending. After pre-reassigning the training data following Data-Cube, we start training from the checkpoint of INSTRUCTOR-large (335M parameters) (Su et al., 2023) with a batch size of 64. We use a softmax temperature τ of 0.01 and optimize the model with the AdamW optimizer. The warmup ratio is set to 0.1 and the learning rate is 2×10^{-5}.

C Evaluation Settings

Evaluation Dataset We conduct evaluations on four task categories within the MTEB dataset (Muennighoff et al., 2023), encompassing a total of 28 downstream tasks. For STS tasks, we choose 10 datasets (*e.g.,* BIOSSES (Soganciogl et al., 2017), STS12-77 (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017), and SICK-Relatedness (Marelli et al., 2014)), to assess the performance. We employ Spearman's correlation of the English test set as the evaluation metric. Reranking tasks include AskUbuntuDupQuestions (Lei et al., 2016), MindSmall (Wu et al., 2020), SciDocsRR (Cohan et al., 2020), and StackOverflowDupQuestions (Liu et al., 2018). Mean Average Precision (MAP) of the test set is utilized to measure performance. Clustering tasks encompass 11 datasets, such as ArxivClusteringS2S, ArxivClusteringP2P, BiorxivClusteringS2S, BiorxivClusteringP2P, MedrxivClusteringP2P, MedrxivClusteringS2S (Muennighoff et al., 2023), RedditClustering, RedditClusteringP2P, StackExchangeClustering, StackExchangeClusteringP2P (Geigle et al., 2021), and TwentyNewsgroupsClustering*. In these clustering tasks, we use v-measure of the test set as the evaluation metric. Pair Classification tasks consist of SprintDuplicateQuestions (Shah et al., 2018), TwitterSemEval2015 (Xu et al., 2015), and TwitterURLCorpus (Lan et al., 2017). Performance is assessed using accuracy on the test set.

*https://scikit-learn.org/0.19/datasets/twenty$_n ewsgroups.html$

Baseline Models We select several sentence representation methods that have achieved state-of-the-art performance on STS tasks, including publicly available models scaling from 100M to 11B parameters, and APIs without exact parameter amounts. Concretely, we select traditional no-instruction sentence representation models (*e.g.,* GloVe (Pennington et al., 2014), USE (Cer et al., 2018), Contriever (Izacard et al., 2022b), GTR (Ni et al., 2022b), SimCSE (Gao et al., 2021), SGPT (Muennighoff, 2022), E5 (Wang et al., 2022a), and SentenceT5 (Ni et al., 2022a)), instruction-based models (*e.g.,* Jina (Günther et al., 2023), Udever (Zhang et al., 2023), Stella [†], BGE (Xiao et al., 2023), GTE (Li et al., 2023), and INSTRUCTOR), and APIs (*e.g.,* OpenAI Text Embedding (Neelakantan et al., 2022), Voyage [‡], Cohere [§], and Ember [¶]).

[†]https://huggingface.co/infgrad/stella-base-en-v2
[‡]https://docs.voyageai.com/
[§]https://txt.cohere.com/introducing-embed-v3/
[¶]https://docs.llmrails.com/embedding/embed-text