# PowerBERT: Improving BERT with a Power Set Ensemble of Fine-Tuned Single and Multitask Models

Stanford CS224N Default Project

**Eric Lee**
Department of Computer Science
Stanford University
ericlee7@stanford.edu

**Jeanette Han**
Department of Computer Science
Stanford University
jhan306@stanford.edu

**Kevin Song**
Department of Computer Science
Stanford University
kevsong@stanford.edu

## Abstract

In this paper, we explore a novel approach to multi-task model ensembling — the Power Set Ensemble, or **PowerBERT** — and its potential to optimize performance for three downstream NLP tasks: semantic textual similarity, paraphrase detection, and sentiment analysis. Ensembling the seven models within the power set of tasks with two different weighing schema — voting and weighted — achieved overall accuracy rates of 76.8% and 77.0% respectively, which is competitive with well-known models. As part of our implementation, we applied an array of techniques including cosine similarity fine-tuning, L2 regularization, and gradient surgery over the single and multi-task models that comprise PowerBERT. In doing so, we explored the potential for multiple techniques working in tandem to further enhance model performance.

## 1 Key Information

- Mentor: No
- External Collaborators: No
- Sharing project: No
- Team Contributions:
  - **Eric Lee** implemented cosine similarity fine-tuning with L2 regularization, trained fine-tuned single-task models, and assembled the final PowerBERT ensemble models.
  - **Jeanette Han** completed minBERT implementation, implemented gradient surgery for multitask training, and trained fine-tuned multitask models.
  - **Kevin Song** conducted research into implementation of extensions, experimented with ensemble schemes, and led writing and editing of the paper/poster.

## 2 Introduction

Powerful LMs, namely BERT (Devlin et al. (2018)), have enabled rich exploration of optimization techniques to improve performance on NLP tasks such as semantic textual similarity (STS), paraphrase detection (PD), and sentiment analysis (SA). Many recent proposals offer novel solutions to address limitations of such LM architectures. For example, multi-task learning with gradient surgery enables concurrent task learning and enhanced performance on multiple tasks. Another significant

improvement is cosine similarity fine-tuning for STS tasks, which more accurately captures semantic similarity by comparing the orientations of embedding vectors rather than their magnitudes. However, there is minimal research into the combination of such optimization techniques to further enhance model performance within an extensive ensemble to train intermediate combinations of tasks.

To address this gap, we began with a pre-trained BERT model (minBERT) and implemented various techniques including **cosine similarity fine-tuning** and **multi-task learning** with **gradient surgery** independently. Additionally, we tested a novel approach to **ensembling** by creating an individual model for each element of the power set of tasks and ensembling their results by both a voting system and a performance-based weighting scheme. **PowerBERT**, our final ensemble model, achieved the best performance rates of all the individual models and techniques experimented with.

## 3 Related Work

Vaswani et al. (2017) was the landmark paper that initially proposed the transformer architecture, inspiring development of more robust LMs such as BERT (Devlin et al. (2018)), which featured bidirectional text processing and popularized robust and unsupervised pre-training, with fine-tuning for specific downstream tasks. The BERT architecture has laid a crucial foundation for diverse experimentation due to the inherent flexibility of the model and its potential for extension. For example, additional layers can be added to test techniques such as regularization, or to train on specific tasks by simply replacing the top layer with a task-specific layer. A prominent example is RoBERTa (Liu et al. (2019)), a more robust transformer-based model that is more rigorously trained and utilizes a more flexible tokenization method (byte-pair encoding) as opposed to WordPiece embeddings. Notably, the RoBERTa paper also introduces the idea of **ensembling**: training on multiple instances of RoBERTa with different initializations and then combining their outputs yielded better results due to reduction in variance/noise and better generalization.

One such optimization technique built over the BERT model is the **multi-task learning** framework, proposed by Qiwei Bi (2022), to improve the quality and accuracy of news recommendations. This framework, named MTRec, employs BERT as the news encoder and additive attention as the user encoder to address the main task of news recommendation, as well as two auxiliary tasks — category classification and named entity recognition (NER) — to capture category and entity information. MTRec simultaneously optimizes three tasks that capture holistic information about user preferences for and engagement with news content: (1) *click prediction* (will a user click on a news article?) (2) *reading duration prediction* (how long will a user read this article for?), and (3) *news category classification* (what category of news should this article be classified as?). An ablation study conducted within the paper affirms the equal contribution of both auxiliary tasks in optimizing the main task, which inspired our own power-set model design.

Additionally, Reimers and Gurevych (2019) introduce Sentence-BERT (SBERT), which is built over BERT and RoBERTa and is optimized specifically for NLP tasks such as sentence similarity, clustering, and semantic search. The motivation behind SBERT is to alleviate the computational overhead and training time of existing NLP structures: instead of requiring both input sentences to be fed into the model like BERT and RoBERTa, SBERT simply harnesses **cosine similarity** to determine STS, significantly decreasing computational overhead and reducing the time needed to determine the most similar sentences in a set of $n = 10,000$ sentences from $\tilde{65}$ hours on a modern V100 GPU to just 5 seconds.

## 4 Approach

Noting that techniques such as cosine similarity, multitask learning, and ensembling have been independently confirmed to improve pre-trained BERT model performance, our approach aims to create a robust BERT model by combining these strategies within a novel ensemble. Specifically, we aim to see whether this combination of techniques improves performance on three NLP tasks: sentiment analysis (SA), paraphrase detection (PD), and semantic textual similarity (STS). As our baselines, we implement the provided minBERT model which, without any additional frameworks or optimization, attained an overall dev score of 47.1%. This served as a baseline standard of comparison to gauge model performance.

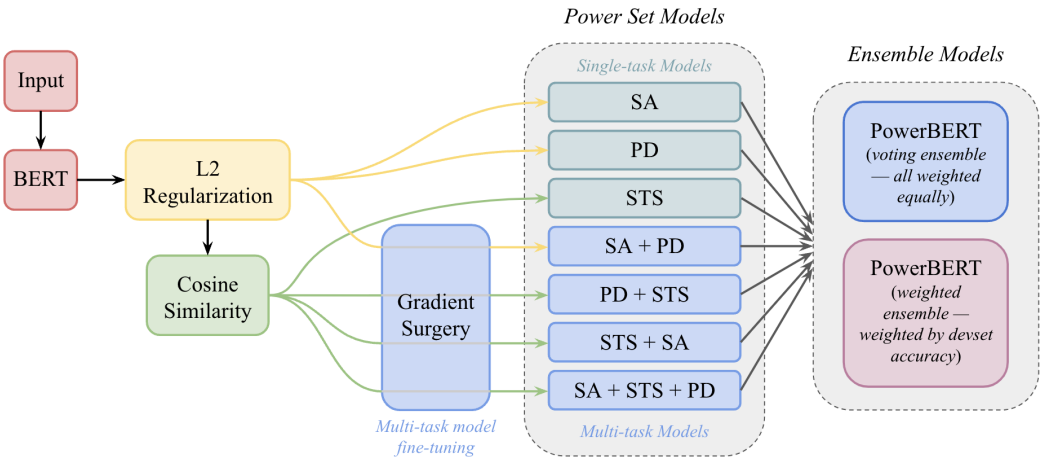


Figure 1: PowerBERT framework. Note that power set models for STS task additionally leverage cosine similarity fine-tuning.

Consider the three distinct tasks that the multi-task model must respond to: SA, PD, and STS. We take the power set of these tasks, which (excluding the resulting empty set) encompasses all possible combinations of these tasks that the model can simultaneously learn. This includes each individual task, all pairwise combinations, and the combination of all three tasks, resulting in 7 subsets: $\{SA\}, \{PD\}, \{STS\}, \{SA + PD\}, \{PD + STS\}, \{STS + SA\}$, and $\{SA + STS + PD\}$. Our critical novel contribution is training fine-tuned versions of these seven models and combining them in an ensembled model whose output is a weighted average of the individual model outputs. We tested two ensembling schema for PowerBERT. In the *voting ensemble*, the logits for each model relevant to the given task are evenly averaged (arithmetic mean), such that each model has an equal "vote" in the final output. In *weighted ensemble*, logits from models with higher accuracy are assigned greater weights. The motivation behind this power set ensemble is to combine the beneficial effects of specialization in the single-task models with insights about complex interactions learned in the multitask models for more robust predictions.

In order to maximize the performance of PowerBERT, we implemented an array of extensions to maximize the performance of each individual model within the power set:

**L2 Regularization.** To reduce model complexity, we utilize **L2 regularization**, a technique pioneered by Hoerl and Kennard (1970), to address overfitting, which we diagnosed via noticeable drops between dev and test set performance. We implemented this by adding an L2 regularization term to the loss, which captures the model complexity by summing the squares of all *n* feature weights:

$$\text{L2 regularization term} = ||w||_n^2 = w_1^2 + w_2^2 + ... + w_n^2 \tag{1}$$

This regularization method penalizes significantly higher weights, enabling us to minimize loss and model complexity concurrently.

**Cosine Similarity Fine-Tuning.** Because STS, one of our tasks of interest, involves comparing embeddings to determine how similar they are, we can measure the semantic distance between embeddings with **cosine similarity** as a fitting metric. Cosine similarity simply utilizes the cosine of the angle between two embeddings as a metric for their similarity:

$$\text{cosine-sim}(u, v) \text{ for embeddings } u, v = \frac{u \cdot v}{||u||||v||} \tag{2}$$

In particular, during training for STS, we return the cosine similarity of the embeddings after the forward pass, with absolute value and scalar multiplication applied to transform the prediction to

match the range of the label (shifting the range from [-1, 1] to [0, ..., 5]). In training, we take the mean squared error between this transformed prediction and the label.

**Multitask Fine-Tuning.** Multitask fine-tuning allows us to train models that generalize to multiple tasks simultaneously. In our approach, we cycled through the each dataset in alternating loops for each epoch.

However, we noticed in our early development that increased accuracy in one task was often paired with lack of progress in another. To counter this issue, implemented **gradient surgery** in each of our multitask learning models, a technique introduced by Yu et al. (2020) that projects the gradient of the $i$-th task, $g_i$, onto the normal plane of another conflicting task's gradient, $g_j$:

$$g_i = g_i - \frac{g_i \cdot g_j}{||g_j||^2} * g_j \tag{3}$$

This eliminates potential gradient direction conflicts between tasks during backpropagation and allows more streamlined progress toward high accuracy among all tasks.

# 5 Experiments

## 5.1 Data

We utilize four different datasets to train the three associated tasks:

- Sentiment analysis: Stanford Sentiment Treebank dataset (11,855 sentences with sentiment labels), CFIMDB dataset (2,434 polar movie reviews with positive/negative labels)
- Paraphrase analysis: Quora dataset (404,298 question pairs with boolean labels)
- STS: SemEval STS Benchmark Dataset (8,628 sentence pairs labeled 0 to 5 on the STS scale)

We use the given train/dev/test splits for these datasets, with no additional preprocessing.

## 5.2 Evaluation Method

We used the dev and test set accuracy metrics detailed in Section 4.4 of the Default Project specifications to evaluate model performance: label accuracy for sentiment analysis and paraphrase detection, and Pearson correlation for semantic textual similarity. Additionally, given that the original paper achieved 80.78% accuracy on STS with cosine-similarity, we treat this as a baseline for performance for PowerBERT.

## 5.3 Experimental Details

For every model, we trained using an Adam optimizer with weight decay 1e-4, L2 regularization, and a learning rate 1e-5 for 10 epochs on Google Cloud GPUs. For models trained for the STS task, cosine similarity fine-tuning was implemented with the transformation of logits to match label ranges as described in Section 5. For all multitask training models (trained on either two or three tasks), gradient surgery was applied after each alternating loop cycling through training examples in the relevant datasets.

After creating the seven individual models each training on a different subset of tasks, the individual models were loaded into the ensemble model, which provided predictions by aggregating logits by either the *voting ensemble* or *weighted ensemble* weighing schemes. Note that in the ensemble models, weights are only given to relevant models for each task. Thus, for each task in the *voting ensemble* model, the logits from each relevant individual model each had weight $\frac{1}{4}$ (ex. for the SA task, the logits from each of the {**SA**}, {**SA**, PD}, {**SA**, STS} and {**SA**, PD, STS} models is $\frac{1}{4}$). Then, for each task in the *weighted ensemble* model, the logits from the relevant models each had weight proportional to that model's accuracy on the dev set (ex. for the SA task, the weight for the {SA} model is $\frac{\text{\{SA\} Dev Accuracy}}{\text{\{SA\} Dev Accuracy + \{SA, PD\} Dev Accuracy + \{SA, STS\} Dev Accuracy + \{SA, PD, Dev STS\} Accuracy}}$).

### 5.4 Results

The dev set results of our intermediate and final models are as follows:

| Model (CS = Cosine Similarity Fine-Tuning, GS = Gradient Surgery, L2 = L2 Regularization) | Sentiment Analysis Accuracy | Paraphrase Detection Accuracy | STS Pearson Correlation |
|---|---|---|---|
| minBERT | 0.318 | 0.632 | -0.070 |
| Naive SA, PD, STS | 0.337 | 0.683 | 0.204 |
| SA, PD, STS (CS) | 0.473 | 0.847 | 0.454 |
| Fine-Tuned SA (L2) | 0.515 | 0.385 | 0.259 |
| Fine-Tuned PD (L2) | 0.203 | 0.854 | 0.519 |
| Fine-Tuned STS (CS, L2) | 0.202 | 0.382 | 0.772 |
| Fine-Tuned SA, PD (GS, L2) | 0.478 | 0.857 | 0.534 |
| Fine-Tuned PD, STS (CS, GS, L2) | 0.219 | 0.854 | 0.730 |
| Fine-Tuned SA, STS (CS, GS, L2) | 0.499 | 0.474 | 0.718 |
| Fine-Tuned SA, PD, STS (CS, GS, L2) | 0.470 | 0.843 | 0.705 |
| Single-Model Ensemble | 0.515 | 0.854 | 0.772 |
| **Voting PowerBERT Ensemble** | 0.529 | 0.869 | **0.832** |
| **Weighted PowerBERT Ensemble** | **0.537** | **0.873** | 0.830 |

We observe that each extension (namely cosine similarity fine-tuning, gradient surgery, and L2 regularization) contributed to increased performance across all three tasks in early development. Further, each of the fine-tuned subset models clearly demonstrated strong performance on the tasks they were fine-tuned on and weak performances on the tasks they were not fine-tuned on—showing the effectiveness of the fine-tuning. A significant insight is that while the single-task models achieved higher scores for the task they focused on, the multitask models achieved scores that were lower but still strong across tasks. This observation lends support to the power set framework to balance the strengths and weaknesses of each model.

As a point of comparison, we trained a *single-model ensemble* model, serving as a naive ensemble of the three single-task fine-tuned model using the direct logits from each individual model for the corresponding task. Evidently, this approach is relatively successful (achieving greater accuracy on all tasks in comparison to the multitask model trained on all tasks), validating the effectiveness of ensembling models. Simultaneously, the performance of the *single-model ensemble* model validates the necessity of the power set ensemble framework, as the two **PowerBERT** models incorporating all seven individual models demonstrate superior results across all tasks.

These results are echoed on the test set as well:

| Model | Sentiment Analysis Accuracy | Paraphrase Detection Accuracy | STS Pearson Correlation | Overall Accuracy |
|---|---|---|---|---|
| Single-Model Ensemble | 0.524 | 0.858 | 0.796 | 0.760 |
| Voting PowerBERT Ensemble | 0.527 | **0.872** | **0.809** | 0.768 |
| **Weighted PowerBERT Ensemble** | **0.534** | **0.872** | 0.806 | **0.770** |

On the test set, the success of the powerBERT models in comparison to the single-model ensemble validate the necessity of incorporating the individual multitask models as a power set configuration in the ensembled model. Further, the slight edge in the *weighted ensemble* schema suggests that the choice of weighting scheme within the power set ensemble is a factor in performance, and that weighing by dev set accuracy shows promise as a method. Overall, the overall peak accuracy we achieved was 77.0% with *weighted ensemble* PowerBERT, which seems comparable with top models (top 30 on leaderboard) and meets the benchmark 0.808 STS score from the original cosine similarity fine-tuning paper (Reimers and Gurevych (2019)).

## 6 Analysis

Through the various multitask finetuning extensions we implemented within our novel ensemble structure, we were able to achieve increases across the various NLP tasks. We further explored visualized distributions of our model's results for the similarity and sentiment datasets to identify performance trends and pinpoint potential areas of improvement.

Figure 2 (Left) shows that the vast majority (87%) of our model's errors deviated by only one rating value away from the correct labels. Inspecting these results combined with the counts of rating values in Figure 2 (Right) indicates that the model accurately identifies general positive or negative sentiment but struggled to discern between differences in severity. We see in Figure 2 (Left) that most of the model's outputs are clustered around the middle values (2-4), which reveals a lack of ability to distinguish highly positive or negative sentiments. We discover that the issue arises most commonly in cases of negative reviews with advanced semantics like colloquial sayings or sarcasm. For example the sentence: "If the first Man in Black was money, the second is small change" received a rating of 2 (Neutral) from our model when the actual rating was 0 (Very Negative). Inability to capture the meaning of the negative relationship between "small change" and "money" is most likely what led to a neutral rating. Across the results, these nuances in semantics demonstrate the contextual challenges we faced in sentiment level prediction. Additionally, we examined that some of the dataset labels could have been ambiguously determined. For example the review: "A delightful coming-of-age story." received a rating of 4 (Very Positive) from our model when the actual rating was 3 (Positive). We believe there can be arguments to support why PowerBERT gave a more positive rating, suggesting that there could be inconsistencies in the Dev or training set leading to these small differences.

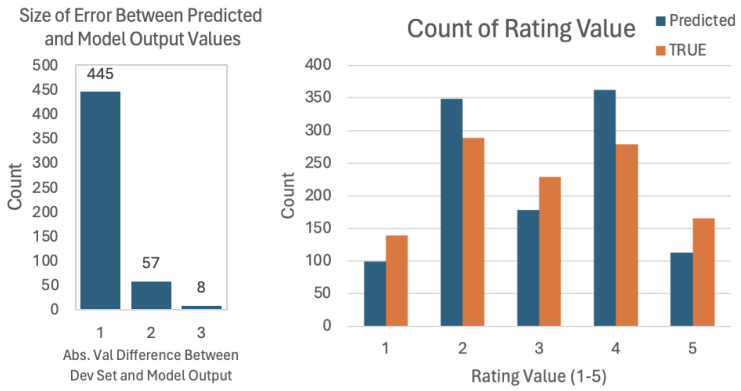


Figure 2: (Left) Distribution of difference between SST Dev set and model ratings for all occurrences of error. (Right) Counts of ratings for SST Dev set and model across score range. Scores ranges from 0 (Very Negative) to 4 (Very Positive)

Looking at the STS development dataset, we noticed that our final model generally predicted scores within a range of one from the accurate rating as seen in Figure 3 (Left), but issues arose in the lower ranges of scores. The model had much greater difficulty identifying scores in the lowest bucket (0-0.5) from Figure 3 (Right) but was quite accurate for the middle range. Our model appears to assign higher similarity scores when sentences have the same word tokens, even if their meaning and use is different. The sentence pair "Work into it slowly." and "It seems to work." received a rating of 2.7 (Neutral) from our model when the actual rating was 0 (Not at all related). Even if this example sentence pair is quite short in length, there should be enough context for our model to accurately determine that the word "work" is used differently in each. These discrepancies could reveal that our model might rely too heavily on token similarities rather than evaluating semantic embeddings.

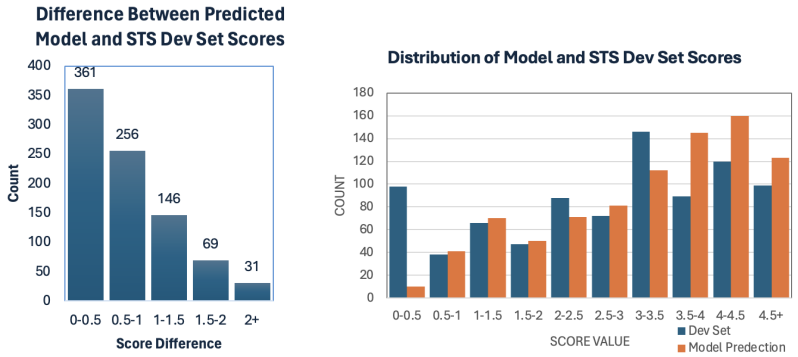


Figure 3: (Left) Distribution of score difference between STS Dev set and model ratings. (Right) Counts of ratings outputted by STS Dev set and model across score range. Scores ranges from 0 (not at all related) to 5 (same meaning)

## 7 Conclusion

PowerBERT was able to achieve significant improvements on top of the basic BERT model through our exploration of advanced fine-uning techniques such as Cosine Similarity Fine-Tuning, Gradient Surgery, L2 Regularization and a novel power set ensemble structure. Our final model generated competitive results, yielding an overall test set result of 0.77 and an STS score matching the original Reimers and Gurevych (2019) paper. Our ensemble model combined the strengths of various models in the power set to create a robust model demonstrating proficiency on multiple tasks.

We acknowledge limitations in our model in the diversity of our training data and difficulty identifying advanced linguistic phrases - seen particularly in the STS development set. Since we only trained and evaluated on the provided datasets, we are unsure of the capabilities of our model on other NLP tasks as well as more diverse cases.

To improve PowerBERT in the future, we would want to continue developing the ensemble model and focus on exploring different methods for weighing the models in the power set. Currently, we have a strong baseline weighing schema by development set accuracy, but we strive to improve this method and determine the most optimal weights. We plan to continue investigation with future work training another neural network to learn the optimal weights to attribute to each model. Given the improvement with the implementation of the *weighted ensemble* schema, we are optimistic that this change will yield further improvements in accuracy.

## 8 Ethics Statement

There are several potential ethical challenges posed by our work. While cosine similarity fine-tuning clearly optimizes model performance, this is predicated on the assumption that the embeddings used to calculate cosine similarity do not contain bias themselves. However, this is difficult to confirm given the sheer size of the datasets, as well as the usage of corpora that are open to public contribution such as IMDB and Quora. If embeddings encode bias — for example, associating certain genders with certain professions, placing them closer semantically — the cosine similarity will simply reinforce the proximity of the embeddings, further propagating this encoded bias. An ideal mitigation measure would be to parse datasets and remove clearly biased datapoints, but this may be impractical and is ultimately still a subjective process. A more practical measure could involve applying techniques that combat overfitting (e.g. smoothness-indusing adversarial regularization) and debias parameters to reduce the severity of the issue.

Furthermore, training multi-task and ensemble models will be computationally expensive and resource-intensive, which can have non-negligible environmental impacts (Strubell et al. (2019)); in our approach alone, our final PowerBERT ensemble models comprise 7 unique models, each requiring its own training. Furthermore, our results may validate the development of models with increasingly significant training resource expenditure, especially since the number of individual "power set" models will increase exponentially with the number of tasks. However, our findings

underline that multi-task learning still enhances the performance of BERT models, even when trained on comparatively smaller datasets than those of the original papers (Qiwei Bi (2022), Reimers and Gurevych (2019)); thus, it is important to assert that smaller LMs can still match the performance of massive LMs. Furthermore, we made conscious design choices within our framework to minimize overhead complexity where possible; for example, we utilize L2 Regularization over SMART Loss to more effectively deploy computational resources.

## References


Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Arthur E. Hoerl and Robert W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*.

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

Lifeng Shang Xin Jiang Qun Liu Hanfang Yang Qiwei Bi, Jian Li. 2022. Mtrec: Multi-task learning over bert for news recommendation. *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *CoRR*, abs/2001.06782.