Thank all reviewers for the valuable comments and suggestions. Please find responses (R) to specific comments (C).

To Reviewer #1

C1: *Misleading comparisons to ELECTRA in RTE, STS-B and MRPC.*

R1: In our submission version, we adopt MNLI-initialization for RTE, STS-B, and MRPC to be consistent with the fine-tuning setting of RoBERTa. Actually, we have carried experiments on RTE, STS-B, and MRPC without MNLI-initialization to make a fair comparison with ELECTRA. The results are shown in Table 1. Removing MNLI-initialization only slightly hurts the performance on RTE and MRPC, but still outperforms ELECTRA on average score. We will add this comparison into our paper in the next version.

C2: *Other suggestions and minor typos.*

R2: Thanks for your suggestions. We will refer the ablation study in the appendix somewhere in the main paper, add pseudo code to describe our attention mechanism and mask strategy, and also fix the typos in the later version.

To Reviewer #2

C1: *The results are mainly reported on a base-level model.*

R1: Pre-training on large-level model requires huge amount of computation resource. Therefore, in this version, we initialize our large-level model from the RoBERTa model and continue to pre-train only 100K steps for a quick verification, which cannot fully demonstrate the advantages of our method. We are pre-training the large-level model from scratch, and will update the results when it is finished.

C2: *Do you submit test scores based on the best performing model on dev set.*

R2: Yes. Following the previous practice [1], we submit the best performing model on the dev set to evaluate the test scores.

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

C3: *Minor issues.*

R3: Thanks for your valuable suggestions. We will refine these issues in the later version.

To Reviewer #3

C1: *Providing a discussion about the comparisons between the proposed method and standard language modelings.*

R1: Thanks for the suggestion. As mentioned in Section 2.3-2.4, we have discussed the comparisons between MPNet and masked/permuted language modeling. We will add the discussion about our method and standard language modeling in the later version.

To Reviewer #4

C1: *Improvements in the large-level model.*

R1: Pre-training on large-level model requires huge amount of computation resource. Therefore, in this version, we initialize our large-level model from the pre-trained RoBERTa model and continue to pre-train only 100K steps for a quick verification, which cannot fully demonstrate the advantages of our method. We are pre-training the large-level model from scratch, and will update the results when it is finished.

C2: *SuperGLUE is recommended.*

R2: Thanks for your recommendation. We will prepare experiments on SuperGLUE and report it in the later version.

C3: *It will be better to have a discussion about the recent advances of researches and significance of this work.*

R3: Thanks for your advice. We will add a related work section to discuss the recent research about pre-trained language modelings to analyze the recent advances and significance of our work.

C4: *Why not choose the SOTA methods as backbone (e.g., ALBERT)?*

R4: We adopt BERT based structure as our backbone since it is one of the most popular architectures used in this field. We will conduct experiments based on other SOTA models (e.g., ALBERT) to manifest the generality of our method in the future.

C5: *What about training efficiency?*

R5: The training efficiency are reported in Table 2. When compared to XLNet/RoBERTa, we found our method can achieve better performance, but fewer computations. We will add the comparisons of training efficiency in the new version.

Model	RTE	STS-B	MRPC	GLUE
ELECTRA	75.2	91.0	88.1	85.8
MPNet	81.0	90.7	89.1	86.5
- MNLI-init	79.8	90.7	88.7	86.3

Table 1: Results of RTE, STS-B and MRPC on the test set without MNLI-initialization. "- MNLI-init" means disabling MNLI-initialization in MPNet. "GLUE" means the average score on all GLUE tasks.

Model	Train FLOPS	GLUE
BERT	6.4e19 (0.06×)	83.1
XLNet	1.3e21 (1.10×)	84.5
RoBERTa	1.1e21 (0.92×)	86.4
MPNet-300K	7.1e20 (0.60×)	87.7
MPNet-500K	1.2e21 (1.00×)	87.9

Table 2: Comparisons of training flops in different methods under $\text{BERT}_{\text{BASE}}$ setting. BERT is trained on 16GB data and others are on 160GB data.