General Response Thanks for all reviewers for your insightful comments. We appreciate the reviewers for your commendation for the simplicity, intuitiveness and effectiveness of our method. We will carefully address your suggestions on typos, writing style and missing citations in the revision.

About theoretical analysis: The main contribution of our paper is a novel early exiting approach that empirically performs well. The theoretical analysis has never been claimed to be the main contribution, instead it serves as a supplementary analysis for trying to interpret the approach in a different aspect from the empirical analysis. Although it is not with a tight constraint, as Reviewer 2 and 3 commented, we believe it is still meaningful to present the interesting theoretical insights that help explain the observation of the empirical results (e.g., performance improvement), which has been rarely discussed and studied by prior related work.

Reviewer 1 Thanks for your endorsement and the pointer to an interesting neuroscience finding! As for the speed improvement, our approach can improve the speed by around 50% while improving the accuracy on ALBERT-based models. It can also improve the speed by around 150% or 250% with a moderate performance drop for a 12- or 24-layer pretrained language models by decreasing the patience hyperparameter. It is true that our method works better with deeper models. Considering recently released large pretrained language models like Megatron-LM and GPT-3 that contain 72 and 96 layers respectively, we believe our approach will benefit future pretrained language models.

Reviewer 2 *Response to Weaknesses:* (1) The similarity between overfitting and overthinking is indeed the motivation behind our method. We acknowledge that we should better support this claim and will rewrite that part to make it more convincing. (2) Please refer to "General Response". (3) Yes, we've also discussed the ensemble effect in our paper (Ln. 113-114) and "different layers are good for different inputs" is an interesting view point that we would like to explore further.

Response to Additional Feedback: (1) It is indeed helpful and we will add this ensemble baseline in our revision. (2) Yes, it seems smaller than overfitting. We will add error bars. (3) Following DistilBERT [Sanh et al., 2019], we report the average of metrics (e.g., F1 and accuracy for MRPC). We will clarify this in the revision. (4) We are also surprised by the training time reduction. Our guess is that training by taking multiple losses can facilitate the training process, similar to multitask learning.

Reviewer 3 *Response to Weaknesses:* Please refer to "General Response".

Response to Correctness: (1) As we described in the caption of Table 2, all results shown in Table 2 are medians of 5 runs. For DeeBERT, we use the official code to obtain the results **on the development set**. Note that the results reported in the DeeBERT paper are on the test set, and are not median/average results. We have contacted the authors of DeeBERT and got their official numbers on development set (81.65 on MNLI, 91.06 on SST-2). We will update our paper with their results. The results of LayerDrop and HeadPrune are also reproduced with their officially released code. For BranchyNet and ShallowDeep, we implement them by closely following the original papers. We will open-source these two implementations along with PABEE. (2) Indeed, there is no contradiction between Table 1 and Figure 3. We control all baselines to have a target speed-up between the speed of 6-layer and 9-layer ALBERT and report the **highest accuracy** in Table 1. As shown in Figure 3, PABEE has a slightly lower acceleration ratio when achieving its performance peak and this is the reason why PABEE in Table 1 looks slower than BranchyNet and Shllow-Deep. Actually, Figure 3 shows that PABEE achieves a better speed-accuracy trade-off (better accuracy at a given speed-up ratio) compared to the baselines across a wide acceleration spectrum. (3) **Equation 6 is correct**. As described in Ln. 121, we allocate more weights to later classifiers following [Kaya et al., 2019]. As stated in Section 3.2 of [Kaya et al., 2019], this design choice is because: *"the earlier ICs have less learning capacity."*

Response to Related Work: (1) We would like to kindly point out that the proceedings of ACL are released **after the NeurIPS deadline**. Also, these papers (including their arXiv preprints) were released within two months before the NeurIPS deadline and thus considered concurrent work according to NeurIPS's policy. We have tried our best to cite their arXiv preprints and add some discussion about them but it is not possible to have a thorough comparison, especially given that two of them are **not evaluated on GLUE**. (2) [Schwartz et al., 2020] does use the same exiting criteria as in [Kaya et al., 2019] (max prediction score) but they differ in details. We will rephrase this sentence. (3) Thanks for the pointers to more related work! We will add them in the revision.

Reviewer 4 Thanks for your endorsement and we are glad that you like the simplicity of our method!

Response to Weaknesses: We will run an experiment on pretraining BERT with the random layer numbers (in the same way we fine-tune on downstream tasks) to verify our guess that the mismatch between pretraining and fine-tuning causes that.

Response to Related Work: Thanks for pointing out the missing citation. We categorized LayerDrop as a static approach because it specifies a predefined set of layers to be used during inference, so the number of layers to be used is not dynamically adjusted with respect to the input. We will reconsider the categorization of LayerDrop per your suggestion.