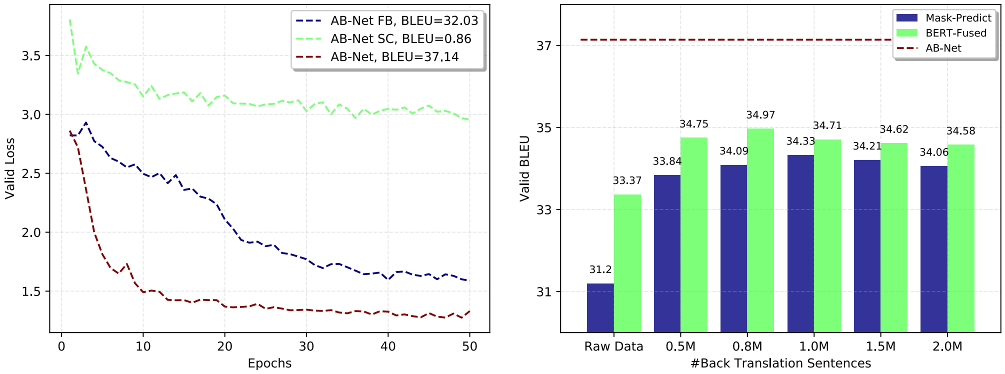

Figure 1: Left: results of different fine-tuning baselines. Right: Results of utilizing back-translation on baselines.

Common Response: We would like to thank all reviewers very much for the detailed feedback and valuable suggestions! We will follow the suggestions on writing and related works and revise accordingly. We agree with the reviewers' concerns about the settings of baselines, therefore we provide a common response here, which will be added to the main paper in the revision. We add a new investigation, where we consider the variant of fine-tuning the whole model in AB-Net (AB-Net FB), the variant that trains AB-Net from scratch (AB-Net SC) and baselines trained with back-translation. We train all settings for 50 epochs and test on the validation set of IWSLT14 De-En. For back-translation (BT), we first train an AT model on IWSLT14 En-De (BLEU score 28.96), and then use it to generate additional training pairs on the English Wikipedia data, which is a subset of the training corpus of BERT. Results are shown in Figure 1. From the left figure, AB-Net converges significantly faster than AB-Net FB, and AB-Net SC does not converge. When fine-tuning the whole model, more GPU memory is required because more gradient states need to be stored, therefore we have to halve the batchsize to fit the model into GPUs, which also slows down the training process. With the same batchsize, AB-Net saves 29% GPU memory and 26% wallclock training time compared with AB-Net FB. Moreover, we find that directly fine-tuning BERT is very unstable and sensitive to the learning rate, while only tuning the adapters can avoid this problem. In the right figure, the gains brought by BT are limited as adding over 1M monolingual data brings a performance drop. Also, BT requires to train another model and decode a large amount of monolingual data, which is time consuming. And our method is orthogonal with BT as shown by the Ro-En results in Table 3(a) of the main paper.

To Reviewer 1: We thank a lot for your detailed feedback!

R1Q1: The speedup of adapters and ablate more pre-training sizes. **R1A1:** Please refer to the common response for our analyses about the first point. For the second point, we have not explored the performance of our method on other pre-training models yet, but we do plan to consider small pre-training models such as DistilBert to explore more choices of adapters. We will also try to find appropriate case studies to provide intuitive illustrations.

R1Q2: Regarding the experimental settings. **R1A2:** We follow Zhu et.al. and use multi-bleu.perl to evaluate results for all models. The data in IWSLT14 tasks is lower cased in most previous works, so does Zhu et.al. according to their released code. Therefore we use uncased BERT for English and report the uncased BLEU scores in IWSLT14 tasks.

To Reviewer 2: We thank a lot for your insightful feedback! For your second question in "Additional feedback", We have tried different architecture variants of adapters and we find the results are similar.

R2Q1: For the suggestions on additional baselines. **R2A1:** Please refer to the common response for the comparisons between our method and additional baselines. For low-resource scenarios, as the multi-lingual BERT is trained on 104 languages, we believe our method can be directly applied to these languages. For other low-resource languages, if there exists monolingual data, then we can first pre-train BERT on it and then apply our method. If not, then it is naturally a very hard problem in MT which requires more ad-hoc techniques, and we will consider them as future works.

To Reviewer 3: We thank a lot for your positive comments about our work! Our method is model agnostic and can be applied to other pre-trained models with less memory and computation cost such as DistilBert.

To Reviewer 4: We thank a lot for your careful check of our manuscript! Regarding your questions in "Weaknesses":

R4A1&R4A3: Please refer to the common response for the additional comparisons with baselines. Our method efficiently brings more performance gains than back-translation, and is orthogonal with it. In addition, we can actually save GPU memory by only tuning the adapters rather than tuning the whole framework. **R4A2:** Please refer to R2A1 and Table 2 in the main paper where we have discussed about low-resource scenarios.

R4A3: We will revise our writing w.r.t your questions Q1-Q4 in "Additional feedback". For Q6&Q7, please refer to our common response and R4A2. For Q5, our method can additionally achieves a promotion of 0.2 BLEU scores on WMT14 En-De when initialized with a pre-trained Transformer decoder. We will update the results in the revision.