A Appendices

A.1 Glossary

- AG: Attention Guidance
- AG Loss: Attention Guidance Loss
- AG Model: Attention Guided Model
- RoBERTa-X-AG: An X layer RoBERTa model trained with MLM and AG loss
- MLM: Masked Language Modeling
- RoBERTa-X-MLM: An X layer RoBERTa model trained with only MLM loss
- SOTA: State-of-the-art
- Head: Self-Attention Heads

A.2 Mathematical Specification of Patterns

Please refer to section 3 for definitions.

Let $CNT(\text{`token'})$ be the total number of occurences of `token` in an input $I$ of length $n$. $I[j]$ represents the $j^{th}$ token in $I$. Let `DELIM` represent the set of all delimiters added by the tokenizer.

$$\mathbf{P}_{\texttt{[Next]}}[p,q] = \begin{cases} 1 & \text{if } q = p+1 \\ \frac{1}{n} & \text{if } p = n \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{P}_{\texttt{[Prev]}}[p,q] = \begin{cases} 1 & \text{if } q = p-1 \\ \frac{1}{n} & \text{if } p = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{P}_{\texttt{[First]}}[p,q] = \begin{cases} 1 & \text{if } q = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{P}_{\texttt{[Period]}}[p,q] = \begin{cases} \frac{1}{CNT(\text{`.'})} & \text{if } I[q] = \text{`.'} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{P}_{\texttt{[Delim]}}[p,q] = \begin{cases} \frac{1}{CNT(\text{`DELIM'})} & \text{if } I[q] \in \texttt{DELIM} \\ 0 & \text{otherwise} \end{cases}$$

We add the patterns in figure 4 for reference.

A.3 Why is AG Loss Useful?

The AG loss converges within 0.2% of pre-training time. This fast convergence is because it is simple to attend to our patterns, which only require propagation of the positional embedding (for `[Next]`, `[Prev]`), or the non-contextual word embeddings in the input layer (for `[Delim]`, `[Period]`). In theory, this is particularly easy for Transformers because of the presence of residual connections (He et al., 2016). We observe from Figure 5 that as soon as AG loss converges, the MLM loss starts decreasing, and we hypothesize that the quick convergence of AG loss because of the reasons explained above is responsible for our method's advantages.

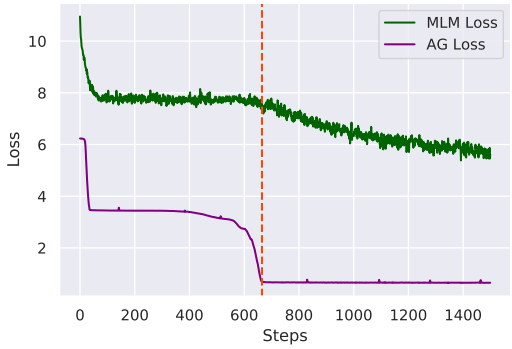


Figure 5: MLM loss ($\mathcal{L}_{MLM}$) and AG loss ($\mathcal{L}_{AG}$) for RoBERTa-12-AG. The MLM loss starts dropping as soon as AG loss converges.

A.4 Running English RoBERTa models on TPUs

To show that the trends mentioned in Table 4 hold even when specialized hardware is used, and models are run for longer, we run the RoBERTa-12-AG and RoBERTa-12-MLM models for 8 epochs (as opposed to 7) on TPUs. The results in Table 9 show that AG models to continue to have the advantage over MLM models.

A.5 Train Loss Curves for ELECTRA

Our method shows faster convergence with ELECTRA, as shown in Figure 6.

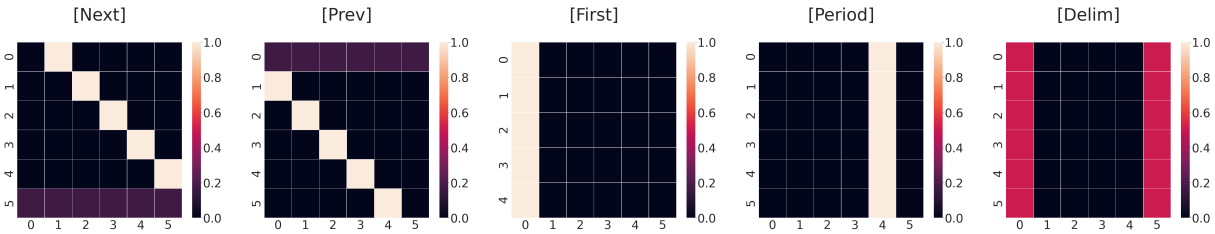


Figure 4: Example attention patterns for the sentence "<s> *Welcome to EMNLP* . </s>". Note that the first three patterns don't depend on the sentence, and can be considered fixed patterns, and the last two depend on the position of the period and delimiters respectively.

Task	RoBERTa-12-GPU		RoBERTa-12-TPU		SOTA	
	MLM	AG	MLM	AG	Model	Score
MNLI-m	78.9	**79.0**	80.3	**81.2**	$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2018a)	84.6⋆
MNLI-mm	77.6	**78.9**	80.8	**81.4**		83.4⋆
QNLI	86.1	**86.8**	88.7	**89.0**		90.5⋆
QQP	68.4	**68.9**	69.3	**69.7**		71.2⋆

Table 9: RoBERTa-12-AG continues to outperform RoBERTa-MLM-AG even when trained on TPUs. This shows that using AG loss provides performance improvements even when using specialized hardware. We also report BERT-Base's scores for reference. Note BERT's scores are not directly comparable because it is trained for 40 epochs and our models are trained for 8.

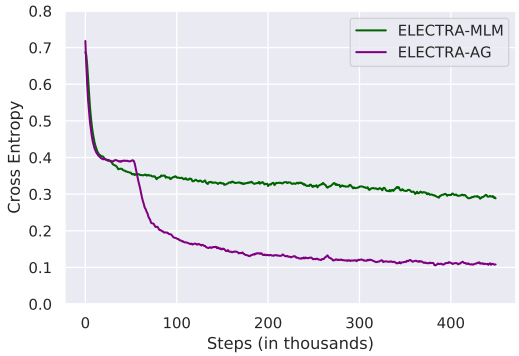


Figure 6: Loss Curves for ELECTRA. The MLM model has an extended plateau, whereas our AG model starts converging almost instantly.

A.6 Train Negative Log-Likelihood Curves for Machine Translation

We report the train loss curves for experiments on machine-translation (section 5.5) in Figure 7. Our AG model converges to the same loss as the BASE model, but the Hard-coded Gaussian model (You et al., 2020) converges to a slightly higher loss.

A.7 Model Configurations

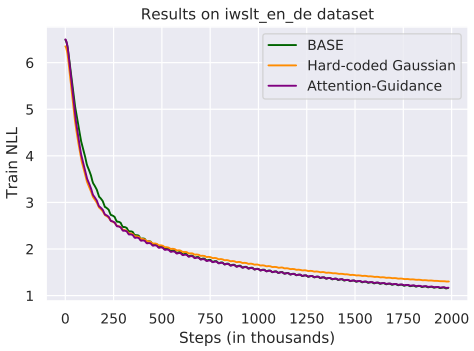
We follow Liu et al. (2019) for all design choices not mentioned in Table 10. The size of feed-forward layers is always $4 \times d_{model}$.

Model	Layers	Heads	Hidden Size
RoBERTa-8	8	12	768
RoBERTa-12	12	12	768
RoBERTa-16	16	16	768

Table 10: Model design choices. Hidden Size is d_{model} in Vaswani et al. (2017). Heads is the number of heads per layer.

A.8 Best performing hyperparameters

The best performing hyperparameters for each model are mentioned in Table 11. All design choices that are not mentioned (like dropout in the feed-forward layer) follow Liu et al. (2019).



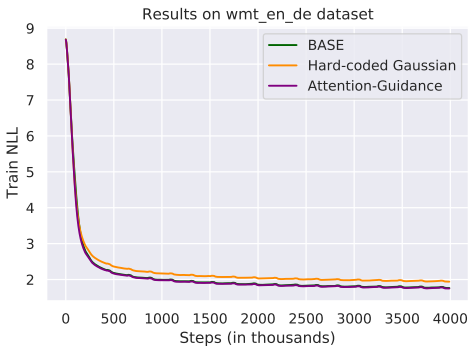


Figure 7: Train loss curves on IWSLT-en-de dataset (left) and WMT-en-de dataset (right). Our AG model converges to the same loss as the BASE model, but the Hard-coded Gaussian model (You et al., 2020) converges to a higher loss.

Language	Model	Learning Rate	Warmup Steps	α_0/λ	Batch Size
English	RoBERTa-8-MLM	1e-4	10000	-	120
	RoBERTa-12-MLM	5e-5	10000	-	84
	RoBERTa-16-MLM	1e-5	10000	-	48
	RoBERTa-8-AG	1e-4	0	100/0.5	120
	RoBERTa-12-AG	1e-4	0	10/0.5	84
	RoBERTa-16-AG	1e-4	0	10/0.5	48
Filipino	RoBERTa-8-MLM	1e-4	10000	-	40
	RoBERTa-12-MLM	5e-5	10000	-	28
	RoBERTa-16-MLM	1e-5	10000	-	16
	RoBERTa-8-AG	1e-4	0	100/0.5	40
	RoBERTa-12-AG	1e-4	0	10/0.5	28
	RoBERTa-16-AG	1e-4	0	10/0.5	16
Oromo	RoBERTa-8-MLM	1e-4	1000	-	40
	RoBERTa-12-MLM	5e-5	1000	-	40
	RoBERTa-16-MLM	1e-5	1000	-	32
	RoBERTa-8-AG	1e-4	0	100/0.5	40
	RoBERTa-12-AG	1e-4	0	10/0.5	40
	RoBERTa-16-AG	1e-4	0	10/0.5	32

Table 11: Best performing hyperparameters. α_0 is the relative weight placed on AG loss (equation 4) and λ is the fraction of heads being guided in each layer.