Gated Position-based Attention for Neural Machine Translation

Anonymous ACL submission

Abstract

Attention is a key component of modern neural machine translation architectures. Its effectiveness was attributed to its capability of modeling word dependencies based on their representation similarity. However, recent work shows that word dependency can be replaced with position dependency with only minor degradation. In this paper, we propose position-based attention as a variant of multi-head attention where the attention weights are computed from position representations. A naive replacement of token vectors with position vectors in self-attention results in a significant loss in translation quality, which can be recovered by using relative position representations and a gating mechanism. We show analytically that this gating mechanism introduces some form of word dependency and validate its effectiveness experimentally under various conditions. The resulting network, rPosNet, outperforms all existing position-based approaches and matches Transformer quality while requiring more than 20% fewer attention parameters after training.

1 Introduction

The Transformer (Vaswani et al., 2017) revolutionized the field of neural machine translation before its wide adoption in numerous other tasks (Dong et al., 2018; Devlin et al., 2019; Dosovitskiy et al., 2021). Using self-attention (Vaswani et al., 2017), the Transformer computes high-level representations for each token as a weighted sum of the entire sequence, where the weights depend on the pairwise content interactions.

Recent work argues that results similar to the Transformer can also be achieved by modeling self-attention weights based on positional instead of content information (Wu et al., 2019; You et al., 2020; Tolstikhin et al., 2021; Liu et al., 2021; Lee-Thorp et al., 2022). Often, these position-based methods are used with some form of gating mechanism that precedes or wraps the token-mixing operation (Wu et al., 2019; Liu et al., 2021; Kim et al., 2023). While it is known that gating is important to guide the information flow of neural networks (Srivastava et al., 2015; Dauphin et al., 2017), its importance for position-based approaches is not covered in the literature. Thus, we compare existing position-based approaches as self-attention replacements for machine translation and observe a significant loss in quality for approaches without gating. Additionally, scoring with a diverse set of metrics shows that no existing approach can consistently match Transformer results, even with gating.

In this paper, we propose rPosNet, a network that leverages gating and computes self-attention weights based on the interactions of absolute and relative position representations. Additionally, we deliver insights into gating and its dependency on position information. In summary, we provide the following contributions:

- Analytically, we derive that wrapping the weighted sum of tokens with a gating mechanism introduces latent content-dependent token-mixing weights (Section 3).
- rPosNet outperforms existing position-based methods and slightly exceeds the Transformer while saving 20% of the self-attention parameters (Section 6).
- We show that increasing the expressiveness of token-mixing weights reduces the usefulness of gating (Section 7.2).
- We observe experimentally that rPosNet is less effective when used in cross-attention. Our gating reformulation suggests one probable reason, but we leave detailed investigations for future work (Section 7.3).

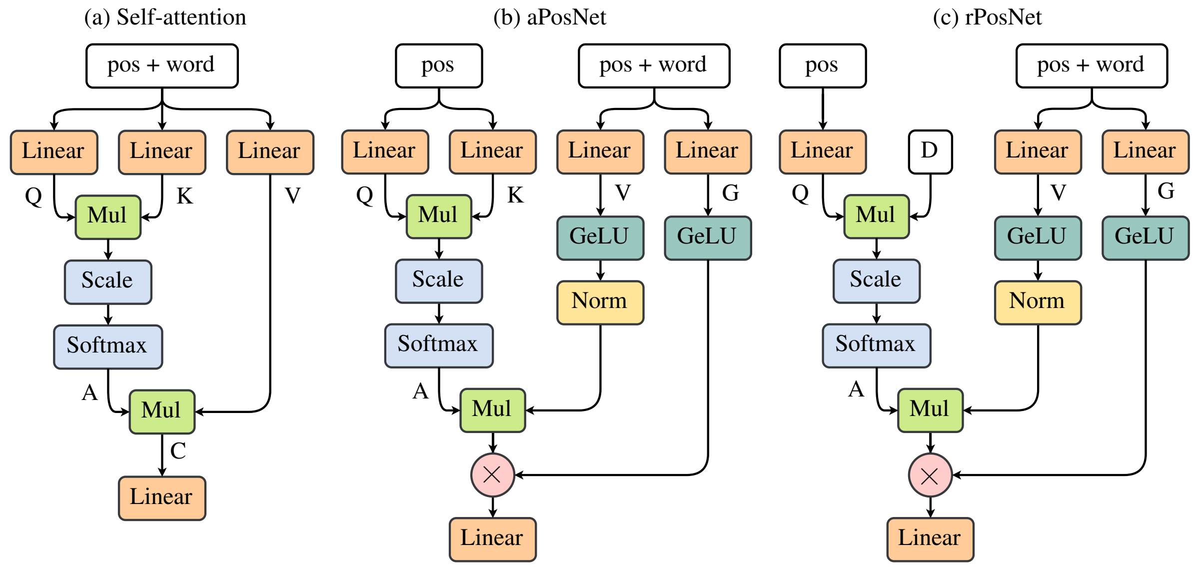

Figure 1: Flowchart representation of (a) self-attention, (b) gated absolute position-based attention (aPosNet), and (c) gated relative position-based attention (rPosNet). While self-attention provides word and position information to queries (Q) and keys (K), we omit word information to calculate the attention weights of PosNet. In rPosNet, we model relative positions using relative position representations (D). Additionally, we employ the gating mechanism presented in Section 2.3, which applies a GeLU activation and layer normalization on the value vectors (V) and elementwise multiplies the context vector (C) with the GeLU activated gate (G).

2 Background

Neural machine translation is typically modeled with an encoder-decoder sequence-to-sequence (Sutskever et al., 2014) Transformer, which mainly consists of multi-head attention and feed-forward sub-layers. In the following, we introduce our notation, position-based token-mixing alternatives and the gating mechanism commonly used in modern architectures.

2.1 Multi-head Attention

Given a source sequence $\mathbf{x} \in \mathbb{R}^{M \times D}$ and target sequence $\mathbf{y} \in \mathbb{R}^{N \times D}$, the multi-head attention mechanism (Vaswani et al., 2017) mixes the elements in $\mathbf{x}$ for every element in $\mathbf{y}$. If $\mathbf{y}$ and $\mathbf{x}$ refer to the same sequence, it is called self-attention. The multi-head concept derives from performing the following operations on H parallel splits of the feature dimension D. In this work, we drop the head indices for simplicity of notation. To calculate the unnormalized mixing weight, referred to as attention energy, of y_n and x_m, those are projected into query and key and combined using the dot product:

$$\hat{\alpha}_{nm} := \frac{(W^Q y_n)(W^K x_m)^\top}{\sqrt{D}}. \qquad (1)$$

Since $\hat{\alpha}_{nm}$ is computed from token contents, we say that attention captures token-token interactions.

The attention weight is then calculated by the softmax normalization of the attention energy:

$$\alpha_{nm} := \frac{\exp \hat{\alpha}_{nm}}{\sum_{m'} \exp \hat{\alpha}_{nm'}}, \qquad (2)$$

and used as the token-mixing weight in the weighted sum over projected input tokens $\mathbf{x}$, denoted value vectors:

$$c_n := \sum_m \alpha_{nm} \cdot (W^V x_m). \qquad (3)$$

We will refer to the result c_n as context vector. Finally, the context vectors of each head are concatenated and mixed with a linear projection, called output projection.

2.2 Position-based token mixing

We briefly overview how existing position-based token-mixing approaches propose to modify the attention weights and provide the corresponding Equations in Appendix A for comparison.

FNet Proposed for language understanding, FNet (Lee-Thorp et al., 2022) applies a 2D Fourier transform over the spatial and feature dimension of $\mathbf{x}$. However, this formulation performed poorly in preliminary experiments, which is why our FNet implementation, denoted FourierNet, applies a 1D Fourier transform along the spatial dimension and employs value and output projections.

GaussianNet Proposed for machine translation, You et al. (2020) hardcode self-attention weights as a Gaussian distribution. They report similar performance to the Transformer when GaussianNet is applied for self-attention but a significant degradation if extended to cross-attention.

LinearNet Tolstikhin et al. (2021) propose mixing tokens with a learnable spatial projection, effectively representing α. It has been proposed, together with other architecture changes, for image classification and natural language understanding with minor degradations to the Transformer.

LightConv For machine translation and other tasks, Wu et al. (2019) introduce a lightweight form of depthwise convolution, which shares the kernel weights W across the feature dimension of a head and the outputs while additionally softmax normalizing them.

gLinearNet Liu et al. (2021) combine the spatial projection of LinearNet with the gating mechanism of Section 2.3. They propose their architecture for image classification and masked language modeling and report significant improvements over Tolstikhin et al. (2021).

2.3 Gating Mechanisms

Various formulations of gating mechanisms have been proposed to control the information flow in neural networks (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Srivastava et al., 2015; van den Oord et al., 2016; Dauphin et al., 2017). They all have in common an elementwise multiplication between two vectors where one, the gate, is bounded in the $[0, 1]$ interval. The gating mechanism we consider here has been proven effective with position-based token-mixing approaches (Liu et al., 2021; Kim et al., 2023) and differs from other gating mechanisms in that the gate is GeLU activated (Hendrycks and Gimpel, 2016) and thus only lower bounded. This gating mechanism modifies the weighted sum of Equation 3 by applying layer normalization (Ba et al., 2016) on the value vector $v_m = W^V x_m$ and elementwise multiplying the context vector with the gate $g_n = \sigma_g(W^G y_n)$:

$$c_n := \Big[\sum_m \alpha_{nm} \cdot \mathrm{Norm}\big(\sigma_g(v_m)\big) \Big] \odot g_n, \quad (4)$$

where σ_g refers to the GeLU function. In general, gating can be applied with any formulation of α. However, we will show experimentally in Section 7.2 that its benefits strongly depend on the information incorporated within α.

3 Reformulating the gating mechanism

To better understand the implications of gating, we reformulate Equation 4. We omit layer normalization for simplicity and will show in the Appendix B that the general reformulation is unaffected if we apply layer normalization on v_m. Additionally, we leverage the GeLU approximation $\sigma_g(v_m) \approx v_m \sigma_s(1.702 v_m)$, where σ_s refers to the Sigmoid function, and rewrite Equation 4 as

$$c_n \approx \sum_m \alpha_{nm} \beta_{nm} \odot v_m. \quad (5)$$

Equation 5 shows that gating the context vector introduces the latent weights $\beta_{nm} \in \mathbb{R}^D$:

$$\beta_{nm} = g_n \odot \sigma_s(1.702 v_m), \quad (6)$$

which consists of the two independent factors $\beta'_n = g_n$ and $\beta'_m = \sigma_s(1.702 v_m)$. While the multiplication of β'_n and β'_m, in general, allows for token-token interactions, the independence of these factors poses a limitation: for a given query token y_n, the ratio between the weights assigned to x_m and to $x_{m'}$ is independent of y_n:

$$\frac{\beta_{nm}}{\beta_{nm'}} = \frac{\beta'_m}{\beta'_{m'}}. \quad (7)$$

In other words, the ratios of token-mixing weights for a query y_n as computed by β are predetermined by the ratios across $\beta'_{1 \ldots M}$. While we show in Section 6 that this limitation is not problematic for self-attention, it may be part of the reason gating and relative position-based attention are not effective in cross-attention (see Section 7.3).

4 Position-based attention

In this Section, we propose position-based attention, which determines the token-mixing weight connecting tokens x_m and y_n solely based on the position-position interactions between n and m. We pair position-based attention with the gating mechanism of Section 2.3.

4.1 Absolute position-based attention

In absolute position-based attention we compute the attention energy as the dot product between the two projected position embeddings $\tilde{n}$ and $\tilde{m}$:

$$\hat{\alpha}_{nm} := \frac{(W^Q \tilde{n})(W^K \tilde{m})^\top}{\sqrt{D_h}}. \quad (8)$$

Note that while $\tilde{n}$ and $\tilde{m}$ are shared across layers, W^Q and W^K are layer-specific. We refer to the combination of Equation 8 and the gating mechanism of Section 2.3 with aPosNet.

Theoretical Complexity Except for the gating overhead, aPosNet has a similar theoretical complexity as attention. However, after training, the position representations $\tilde{n}$ and $\tilde{m}$ are fixed and do not depend on the input. It allows us to pre-compute $\hat{\alpha}$ to obtain a matrix of shape $(H \times N \times N)$. Thus, with projections, one layer accounts for $HN^2 + 3D^2$ parameters. Additionally, this pre-computation reduces the number of operations from $2N^2D + 5ND^2$ to $N^2D + 3ND^2$ by omitting the dot product and key query projections. We compare theoretical complexities in Appendix C.

Relation to gLinearNet With the pre-computed attention energy matrix, aPosNet becomes similar to gLinearNet except that α of gLinearNet is not normalized and has been trained directly.

4.2 Relative position-based attention

To model position interactions with relative position-based attention, we borrow the relative position representations $\tilde{d}_{nm}$ from Shaw et al. (2018), which we use in the dot product with the projected position embedding $\tilde{n}$:

$$\hat{\alpha}_{nm} := \frac{(W^Q\tilde{n})(\tilde{d}_{nm})^\top}{\sqrt{D_h}}. \tag{9}$$

Similarly to Shaw et al. (2018), the distance embedding $\tilde{d}$ is clipped to a maximum unidirectional context size K:

$$\tilde{d}_{nm} := \text{Embedding}_{\text{rel}}\Big(\text{clip}\big(\lfloor \gamma n \rfloor - m, K\big)\Big). \tag{10}$$

However, in contrast to Shaw et al. (2018), we extend relative position-based self-attention to be compatible with cross-attention by multiplying n with the length ratio $\gamma := \frac{M}{N}$ which we determine similar to You et al. (2020) by measuring the average length ratio on the training set. We refer to the combination of Equation 9 and the gating mechanism of Section 2.3 with rPosNet. In Figure 1, we illustrate the operations performed by aPosNet and rPosNet in comparison to multi-head self-attention.

Theoretical Complexity Similar to aPosNet, we can pre-compute $\hat{\alpha}$ of rPosNet after training, which summarizes the interactions between query and relative position representations into a matrix of shape $(H \times \hat{K} \times N)$, where $\hat{K} = 2K + 1$. Thus, pre-computing $\hat{\alpha}$ after training reduces the number of parameters from $\hat{K}D + 4D^2$ to $H\hat{K}N + 3D^2$ and operations from $\hat{K}ND + N^2D + 4ND^2$ to $N^2D + 3ND^2$. With the hyperparameters presented in Section 5, the pre-computation saves 23% of attention parameters in the Base model configuration and 24% in the Big configuration.

Relation to LightConv After pre-computing $\hat{\alpha}$, rPosNet differs from LightConv in that rPosNet does not share token-mixing weights across y_n and has global context. We provide an ablation study in Appendix E to understand the importance of these differences.

5 Experimental Setup

5.1 Datasets & Evaluation

We perform our comparison on four datasets of varying sizes: IWSLT14 German-English (Federico et al., 2014), WMT14 English-{German, French} (Bojar et al., 2014), and WMT18 English-Chinese (Bojar et al., 2018). We evaluate DE→EN models on the test sets TED-{dev10,dev12, test10, tst11, tst12}, EN-{DE, FR} models on newstest14 and EN→ZH models on newstest17. An overview of the dataset statistics is shown in Table 1. We preprocess all datasets using Byte Pair Encoding (BPE) (Sennrich et al., 2016) and lowercase the text for the DE→EN direction.

We report BLEU (Papineni et al., 2002), BLEURT (Sellam et al., 2020), and COMET (Rei et al., 2020) for each evaluation. All scores are calculated on detokenized text. To calculate BLEU scores, we use sacreBLEU[1] and its internal tokenizations[2,3]. For BLEURT and COMET, we use the official implementations[4,5] and the models *BLEURT-20* and *wmt20-comet-da*, respectively. To summarize results, we will refer to the translation quality difference between two approaches as their relative difference averaged across all metrics and datasets.

[1] https://github.com/mjpost/sacrebleu

[2] SacreBLEU signature for EN, FR, DE: nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0

[3] SacreBLEU signature for ZH: nrefs:1|case:mixed|eff:no|tok:zh|smooth:exp|version:2.0.0

[4] https://github.com/google-research/bleurt

[5] https://github.com/Unbabel/COMET

5.2 Model Architectures

Our Base and Big Transformer architectures follow the implementation of Vaswani et al. (2017), whereas, for the Small models, we halve the feed-forward dimension to 1024 and increase dropout to 0.3. We compare position-based token-mixing approaches by leveraging the respective formulations instead of encoder/decoder self-attention while leaving the rest of the Transformer architecture unchanged. We make an exception for FourierNet, which cannot be straightforwardly extended to the decoder because it has an explicit dependency on the sequence length. Instead, FourierNet uses multi-head attention within decoder self-attention.

In preliminary experiments, we found that aPosNet works best with sinusoidal positional embeddings (Vaswani et al., 2017) and rPosNet with learnable embeddings (Gehring et al., 2017). All other position-based token-mixing approaches use sinusoidal positional embeddings. Similar to Shaw et al. (2018), our implementation of rPosNet and LightConv use a unidirectional context window $K = 16$ for the Base and $K = 8$ for the Big model.

5.3 Training Setup

Our training setup closely follows the configuration of Vaswani et al. (2017). Similarly, we use the Adam optimizer and a warmup learning rate schedule with 4000 steps. We group batches by sentence length and train the Small models for 30k steps, the Base models for 150k, and the Big models for 300k. The final model is an average over the best checkpoint and its following if there are enough checkpoints to average, or else we take an average over the last checkpoints. We determine the best checkpoint by its perplexity on the validation set. For De→En, we consistently average 30 checkpoints with a checkpoint period of 300 steps; for the Base models, we average 7 checkpoints with 1000 steps each; for the Big models, 20 checkpoints with 600 steps each. The Small model uses an effective batch size of approximately 16000 target tokens while the Base and Big models accumulate approximately 27000 target tokens per step. We use beam search with a beam size of 12 for all models. All models in this work are implemented in PyTorch (Paszke et al., 2019) and are available in the supplementary material. The Small models are trained on a single 2080 TI graphics card, the Base models on two, and the Big models on four.

Table 1: Dataset statistics.

Dataset	Vocabulary Size		Train Pairs	Test Pairs
	Source	Target		
De→En	10k		160k	6750
En→De	44k		4M	3003
En→Fr	46k		36M	3003
En→Zh	32k	45k	17M	2001

6 Results

We compare translation quality of the Base model configurations in Table 2, and Small and Big model configurations in Table 3.

Gated position-based attention In all experiments, we observe rPosNet performing as well or slightly better than the Transformer with an average translation quality increase of 0.7%. It shows that the self-attention weights of rPosNet, consisting of content-dependent β and position-dependent α, achieve sufficient expressiveness for machine translation. aPosNet cannot match this expressiveness and underperforms the Transformer with an average relative degradation of 1.8%. In the Small setup on De→En, this reaches an absolute degradation of 2.9 points in Comet and 0.8 points in Bleurt. The significant difference between aPosNet and rPosNet highlights the importance of relative position information in α.

The results of gLinearNet and LightConv further emphasize the strong modeling capabilities of absolute (query) and relative position (key) interactions in rPosNet. In comparison, token-mixing weights in gLinearNet solely model absolute position interactions and LightConv relative position interactions. Both cannot match rPosNet's translation quality, with gLinearNet on average lacking behind by 1.3% relative and LightConv by 2.4%. Note that in contrast to Wu et al. (2019), we do not match parameters between LightConv and the Transformer. Most prominent in the Base setting on En→De, rPosNet outperforms gLinearNet by 0.6 Bleurt and 1.9 Comet points. While aPosNet cannot match Transformer results, rPosNet outperforms other position-based methods and is on par with Shaw et al. (2018) and the Transformer across all model sizes and data conditions.

Hard-coded token-mixing weights Our results show that hard coding encoder self-attention

Table 2: Base model results on En→De, En→Fr, and En→Zh. Note that this scoring differs from Vaswani et al. (2017) in that they split German compound words, which usually increases the Bleu score, and from You et al. (2020) in that we use sacreBLEU's default tokenizer, not 'intl'. We ensured that our baseline system and reimplementation of You et al. (2020) match in Bleu when evaluating similarly.

Model	Params	En→De			En→Fr			En→Zh		
	(En→De)	Bleu	Bleurt	Comet	Bleu	Bleurt	Comet	Bleu	Bleurt	Comet
Transformer	66.5M	26.3	71.1	47.6	37.8	69.0	61.1	33.8	64.3	42.5
Shaw et al. (2018)	66.7M	26.3	71.4	48.6	37.8	69.2	61.6	34.0	64.6	43.5
FourierNet	63.4M	22.8	66.0	31.8	34.9	64.2	49.3	31.5	61.6	34.9
GaussianNet	60.2M	25.3	68.1	39.5	36.7	66.9	55.7	32.6	62.6	36.8
LinearNet	61.8M	25.3	69.8	44.3	37.0	67.7	58.2	33.1	63.3	40.2
LightConv	63.4M	26.0	70.6	46.7	37.4	68.6	60.3	33.0	63.5	41.1
gLinearNet	65.0M	26.1	70.8	46.7	37.8	69.1	61.3	33.5	64.0	42.4
aPosNet	65.0M	25.9	70.6	46.1	37.7	69.0	61.4	33.6	63.7	42.2
rPosNet	63.9M	26.6	71.4	48.6	37.9	69.4	61.8	33.8	64.2	43.1

Table 3: Big model results on En→De and Small model results on De→En.

Model	En→De				De→En			
	Params	Bleu	Bleurt	Comet	Params	Bleu	Bleurt	Comet
Transformer	221M	27.1	72.3	50.4	36.8M	35.0	69.3	37.6
Shaw et al. (2018)	221M	27.3	72.7	51.5	37.0M	35.4	69.7	38.8
FourierNet	208M	24.0	67.6	36.5	33.6M	32.5	66.9	28.2
GaussianNet	196M	26.3	69.4	42.3	30.4M	34.3	68.4	34.1
LinearNet	199M	26.6	71.3	48.0	32.0M	34.0	68.3	33.9
LightConv	209M	26.8	71.7	49.1	33.6M	34.4	68.9	35.5
gLinearNet	212M	27.1	72.2	49.9	35.2M	34.5	69.0	36.3
aPosNet	212M	26.8	71.4	47.7	35.2M	34.2	68.5	34.7
rPosNet	210M	27.3	72.2	50.4	34.1M	35.1	69.5	38.2

weights as the twiddle factors of the Fourier transform (FourierNet) leads to poor results for machine translation and, on average across all datasets and metrics, degrades translation quality relative to the Transformer by 13.2%. In GaussianNet, weights are manually designed to follow the normal distribution of Transformer self-attention patterns, which significantly reduces the degradation to 6.3%. However, the translation quality is still considerably worse than LinearNet's, the weakest model with trainable self-attention weights. The difference between LinearNet and GaussianNet is negligible in Bleu but made visible with Bleurt and Comet, which correlate better with human judgment (Kocmi et al., 2021). In particular, we confirmed by manually analyzing a sample of translations (see Appendix F) that the semantic metrics discriminate better between translation hypotheses when they all have little overlap with the references or changing a single word alters the meaning of the sentence. Thus, approaches with learnable token-mixing weights, such as rPosNet, are considerably better than hard-coded approaches.

7 Analysis

7.1 Decoding speed

Table 4 shows the average CPU decoding speed and variance on En→De measured across ten runs. Despite the lower theoretical complexity of position-based attention, we observe similar performance to the Transformer and minor improvements to Shaw et al. (2018) in practice. Comparing profiling results between rPosNet and the Transformer showed that while rPosNet reduces the time spent in linear projections and matrix multiplications from 73% to 65% (with similar total runtime), the overall time

Table 4: Comparing average CPU decoding speeds and variances measured across ten runs on En→De.

Model	Decoding Speed [Tokens \ s]
Transformer	16.8 ± 0.61
Shaw et al. (2018)	16.5 ± 0.60
aPosNet	17.0 ± 0.69
rPosNet	16.8 ± 0.61

spent within self-attention layers stays the same due to the overhead introduced from indexing the attention energies and gating.

7.2 The impact of gating and query-key information

The gating mechanism is known to guide the learning of cross-token patterns (Tu et al., 2017; Dauphin et al., 2017). In Section 3, we mathematically showed that by gating the context vector, these patterns are captured within the latent token-mixing weights β. Since the products of β and α form the actual token-mixing weights, we analyze in this Section how content information in α impacts the usefulness of gating. For that, we compare the utilization of position versus content information in the query and key input of self-attention, with and without gating. The results are visualized in Figure 2, where we depict Comet scores on the y-axis and the query and key input on the x-axis.

The formulation of position-based attention without gating primarily[6] differs from the Transformer in the provided information within queries and keys. Relative attention uses the relative position representations of Shaw et al. (2018)'s approach but without the query-key dot product of multi-head attention. Thus, relative attention differs from rPosNet in that content information is provided to the queries and is equal to dynamic convolutions (Wu et al., 2019) with global context (Chang et al., 2021). In total, the x-axis of Figure 2 depicts position-position interactions for aPosNet and rPosNet, token-position interactions for relative attention, token-token interactions for the Transformer, and token-token + token-position interactions for Shaw et al. (2018). We sort these approaches on the x-axis in order of their attention weight expressiveness.

[6]The position embeddings may also differ between approaches.

Figure 2 shows that gating in position-based attention approaches increases Comet by 3.5 points for aPosNet and rPosNet. While relative attention can leverage gating with a lower but significant absolute Comet increase of 2.6 points, the increment for the Transformer is only 2 points and 0.9 points for Shaw et al. (2018). Thus, gating is less helpful if α can capture content-dependent patterns, and increasing the expressiveness of those patterns diminishes the usefulness of gating. Since gating introduces an additional projection matrix of size D^2 per self-attention layer, content-based mixing approaches may just leverage the additional parameter. In contrast, approaches that do not incorporate content information within the attention weights can benefit from token-token interactions captured in β. Additionally, the comparable performance of rPosNet and relative attention with gating suggests that gating makes the content information within relative attention redundant for translation quality.

7.3 Comparing the usage of rPosNet across attention layers

While the aforementioned experiments concentrated on self-attention, we also consider cross-attention in this Section and analyze how the usage of rPosNet affects translation quality compared to multi-head attention. In Table 5, we depict the translation quality on En→De when combinations of encoder self-attention (enc-self), decoder self-attention (dec-self), and decoder cross-attention (dec-cross) employ multi-head attention (✗) or rPosNet (✓). The model using rPosNet only for cross-attention while all other layers employ multi-head attention (row 4) significantly decreases translation quality by 5.7% relative to the Transformer. The result suggests that content-dependent patterns incorporated by β cannot sufficiently capture source-target token interactions. We hypothesize that part of the reason is the inability of β to express varying relations across source tokens (see Section 3). While this may be a significant limitation of gating, we leave the exploration of this and other possible reasons to future work.

However, utilizing rPosNet within all self-attention layers (row 8), so that rPosNet is the only token-mixing method, does not lead to further degradation of translation quality with a relative degradation to the Transformer of 5.5% (5.3% relative in Bleu). Although the loss is substantial, rPosNet improves upon You et al. (2020)'s

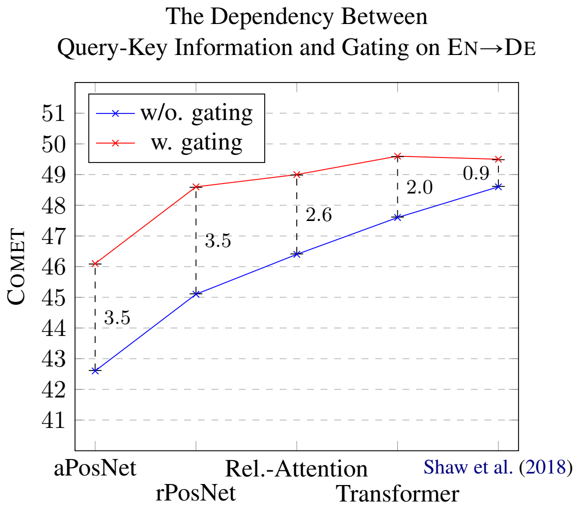

Figure 2: Approaches depicted on the x-axis differ in the provided information to queries and keys. On the y-axis we depict COMET, which is the most accurate metric according to Kocmi et al. (2021), and provide the full Table showing BLEU, BLEURT, and COMET in Appendix D. If position information is provided to queries and keys, gating has a significant positive impact on translation quality that diminishes with the usage of content information.

relative BLEU degradation of 12.3%[7]. Additionally, Table 5 shows that using rPosNet within decoder self-attention is only beneficial if encoder self-attention leverages rPosNet, whereas the usage within encoder self-attention always positively impacts translation quality.

8 Related Work

The question of how to represent position and integrate it into the Transformer architecture has been a vast research field that we briefly want to overview and connect to our approach. An extensive line of research focuses on improving position embeddings (Kitaev et al., 2020; Liu et al., 2020; Kiyono et al., 2021) and their integration into the word vectors (Neishi and Yoshinaga, 2019; Wang et al., 2020). This direction is mainly orthogonal to our approach, and many ideas and methods can be leveraged with position-based attention. We leave these investigations for future work and restricted to learnable (Gehring et al., 2017) and sinusoidal (Vaswani et al., 2017) embeddings.

A different line of research focuses on integrating position within the attention mechanism (Shaw et al., 2018; Dai et al., 2019; Dufter et al., 2020; Huang et al., 2020; Raffel et al., 2020; Ke et al., 2020; He et al., 2021; Wu et al., 2021). They all improve over Transformer models for various tasks by modifying word and position interactions within the attention matrix and introducing relative position representations as a scalar or vector. While they still rely on content-dependent attention weights, they showed the importance of relative position representations, which we also used in rPosNet. However, we are interested in studying purely position-based self-attention approaches and how they can perform at least on par with the (content-based) Transformer. Additionally, we compare with Shaw et al. (2018) as an upper bound since it leverages token-token interactions and was proposed for machine translation.

[7] As reported by You et al. (2020)

Table 5: A translation quality comparison of all combinations in which encoder self-attention (enc-self), decoder self-attention (dec-self), and/or decoder cross-attention (dec-cross) use either multi-head attention (✗) or rPosNet (✓). We conduct the experiments on EN→DE and report BLEU, BLEURT, and COMET.

rPosNet Layers			EN→DE		
enc-self	dec-self	dec-cross	BLEU	BLEURT	COMET
✗	✗	✗	26.3	71.1	47.6
✓	✗	✗	26.4	71.2	48.1
✗	✓	✗	26.1	71.1	47.2
✗	✗	✓	24.6	69.2	43.8
✓	✓	✗	26.6	71.4	48.6
✓	✗	✓	24.8	69.8	45.2
✗	✓	✓	24.3	69.0	42.8
✓	✓	✓	24.9	69.3	43.5

9 Conclusion

We have introduced the gated token-mixing approaches aPosNet and rPosNet for machine translation. Although their token-mixing weights are position-based, the gating mechanism introduces content dependency in the form of latent weights β. These weights capture token-token interactions and are crucial for the results of rPosNet. We have effectively used rPosNet as a self-attention replacement and saved more than 20% of the self-attention parameters without loss in translation quality.

Limitations

The goal of this paper is to find alternatives as good as Transformer that do not require word-word interactions in attention computation. We have compared numerous approaches across many data conditions and model sizes to show the validity of our results. However, we can identify the following limitations in our work:

- rPosNet's decoding is not faster than Transformer despite the better theoretical computation given by pre-computing weights;
- rPosNet's position-based attention is an effective replacement of Transformer's self-attention, but its usage in cross-attention leads to quality loss;
- We did not have enough computational resources to run all our numerous experiments multiple times, so we rely on the consistent results we obtain across different conditions and metrics.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, and Lucia Specia, editors. 2014. *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Baltimore, Maryland, USA.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Tyler Chang, Yifan Xu, Weijian Xu, and Zhuowen Tu. 2021. Convolutions and self-attention: Re-interpreting relative positions in pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4322–4333, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 933–941. JMLR.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2020. Increasing learning efficiency of self-attention networks through direct position interactions, learnable temperature, and convoluted attention. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3630–3636, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marcello Federico, Sebastian Stüker, and François Yvon, editors. 2014. *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*. Lake Tahoe, California.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. 2020. Improve transformer models with better relative position embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3327–3335, Online. Association for Computational Linguistics.

Guolin Ke, Di He, and Tie-Yan Liu. 2020. Rethinking positional encoding in language pre-training. *CoRR*, abs/2006.15595.

Kwangyoun Kim, Felix Wu, Yifan Peng, Jing Pan, Prashant Sridhar, Kyu J. Han, and Shinji Watanabe. 2023. E-branchformer: Branchformer with enhanced merging for speech recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 84–91.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. 2021. SHAPE: Shifted absolute position embedding for transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3309–3321, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to MLPs. In *Advances in Neural Information Processing Systems*.

Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. 2020. Learning to encode position for transformer with continuous dynamical model. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6327–6335. PMLR.

Masato Neishi and Naoki Yoshinaga. 2019. On the relation between position information and sentence length in neural machine translation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338, Hong Kong, China. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations.

In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. 2021. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems*, volume 34, pages 24261–24272. Curran Associates, Inc.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*, 5:87–99.

Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. 2016. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. 2020. Encoding word order in complex embeddings. In *International Conference on Learning Representations*.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. DA-transformer: Distance-aware transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2059–2068, Online. Association for Computational Linguistics.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Weiqiu You, Simeng Sun, and Mohit Iyyer. 2020. Hard-coded Gaussian attention for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7689–7700, Online. Association for Computational Linguistics.

A Formulas describing related position-based token-mixing approaches

In the following, we provide the formulas describing how the position-based token-mixing approaches from Section 2.2 formulate the context vector.

FNet

$$c_n := \mathcal{R}\Big(\sum_m \exp\big[-2\pi j \frac{n \cdot m}{M}\big] \cdot \mathcal{F}_h(x_m)\Big) \tag{11}$$

GaussianNet

$$c_n := \frac{1}{\sigma\sqrt{2\pi}} \sum_m \exp\left[\frac{-(m-\mu(n))^2}{2\sigma^2}\right] \cdot (W^V x_m) \tag{12}$$

LinearNet

$$c_n := \sum_m W_{nm} \cdot (W^V x_m) \tag{13}$$

LightConv

$$c_n := \sum_{k=0}^{2K} \frac{\exp W_k}{\sum_{k'=0}^{2K} \exp W_{k'}} \cdot \sigma_{\text{GLU}}(W^V x_{n+k-K}) \tag{14}$$

B Reformulating the gating mechanism with layer normalization

Substituting $z_m = \sigma_g(v_m)$ we rewrite the gating mechanism of Equation 4 as

$$c_n := \Big[\sum_m \alpha_{nm} \cdot \text{Norm}(z_m)\Big] \odot g_n. \tag{15}$$

Similar to Section 3, we aim to rediscover the weighted sum over v_m. For this, we utilize the definition of layer normalization:

$$\text{Norm}(x) := a \odot [f_1(x)x - f_2(x)] + b, \tag{16}$$

with gain $a \in \mathbb{R}^D$, bias $b \in \mathbb{R}^D$, $f_1(x) = \frac{1}{\sqrt{\sigma_x}}$ and $f_2(x) = \frac{\mu_x}{\sqrt{\sigma(x)}}$. The insertion into Equation 15 gives us:

$$\begin{aligned} c_n \approx\ & a \odot \sum_m \alpha_{nm} \underbrace{f_1(z_m) \cdot g_n \odot \sigma_s(v_m)}_{\beta_{nm} \in \mathbb{R}^D} \odot v_m \\ & - a \odot \sum_m \alpha_{nm} \cdot f_2(z_m) \cdot g_n + \sum_m \alpha_{nm} b \odot g_n. \end{aligned} \tag{17}$$

Table 6: Comparing how different attention approaches leverage gating.

Model	Gating	Params	EN→DE BLEU	EN→DE BLEURT	EN→DE COMET
Transformer	✗	66.5M	26.3	71.1	47.6
	✓	69.7M	26.6	71.6	49.6
Shaw et al. (2018)	✗	66.7M	26.3	71.4	48.6
	✓	69.9M	26.7	71.8	49.5
Rel. Self-Attention	✗	63.6M	25.7	70.4	46.4
	✓	66.7M	26.5	71.3	49.0
aPosNet	✗	61.8M	25.4	69.4	42.6
	✓	65.0M	25.9	70.6	46.1
rPosNet	✗	60.8M	25.3	70.1	45.1
	✓	63.9M	26.6	71.4	48.6

Utilizing the normalization property $\sum_m \alpha_{nm} = 1$ we can simplify Equation 17 to:

$$c_n \approx a \odot \sum_m \alpha_{nm}\beta_{nm} \odot v_m + g_n \odot \Big[b - a\sum_m \alpha_{nm} f_2(z_m)\Big], \quad (18)$$

with

$$\beta_{nm} = f_1(z_m) \cdot g_n \odot \sigma_s(1.702 v_m). \quad (19)$$

Although Equation 18 assumes α to be normalized, not normalizing α does not affect β and only adds a context-dependent scale in front of b. All in all, the Equations show that with or without layer normalization, gating introduces the token-mixing weights β.

C Theoretical complexity comparison

We compare theoretical complexities across position-based token-mixing approaches, the Transformer, and Shaw et al. (2018) concerning the number of operations and parameters in Table 8.

D Table: The impact of gating and query-key information

By depicting COMET scores in Figure 2, we visualized how the effectiveness of gating decreases with increased token-mixing weight expressiveness. In Table 6, we provide the full results with the number of parameters, BLEU, BLEURT, and COMET.

E Ablation analysis: From LightConv to rPosNet

With the similarities between LightConv and rPosNet, we want to understand what features of rPosNet are responsible for its better translation quality.

Table 7: Starting from LightConv and progressively implementing the features of rPosNet.

Model	Params	EN→ZH BLEU	EN→ZH BLEURT	EN→ZH COMET
Transformer	107M	33.8	64.3	42.5
Shaw et al. (2018)	107M	34.0	64.5	43.8
Light Convolution	101M	33.2	63.4	40.6
+ GLU [LightConv]	104M	33.0	63.5	41.1
+ GeLU Gating	104M	33.5	63.7	42.4
+ Global Context	104M	33.6	63.9	42.4
rPosNet	104M	33.8	64.2	43.1

While Wu et al. (2019) propose LightConv initially with the GLU mechanism (Dauphin et al., 2017) (see Equation 14), we differentiate between LightConv with and without GLU since the effect of gating is a central component of our analysis. We start with LightConv without GLU, denoted Light Convolution, and progressively implement the features of rPosNet. In Table 7, we show the translation quality on EN→ZH of the models leveraging the respective position-based approach instead of self-attention. Light Convolution (row 3) shows similar translation quality to LightConv (row 4). Replacing GLU gating with the gating mechanism of Section 2.3, denoted GeLU gating (row 5), increases translation quality noticeably by 0.5 points in BLEU, 0.2 points in BLEURT, and 1.3 points in COMET. Additionally, adding global context (row 6) by spreading the outer kernel weights across the whole sequence increases translation quality slightly by 0.1 BLEU and 0.2 BLEURT (no improvement in COMET). The remaining difference to rPosNet (row 7) is the different training scheme and rPosNet's unshared kernel weights across query positions. Together they add additional 0.2 points in BLEU, 0.3 in BLEURT, and 0.7 in COMET. The results show that all differences between LightConv and rPosNet are responsible for their translation quality difference. While the global context seems negligible for machine translation, GeLU gating, training scheme, and unshared token-mixing weights are the most important.

F Example failure cases of BLEU

Throughout our analysis, we observed that BLEU often disagrees with the semantic metrics BLEURT and COMET. For example, the translation quality in the Base configuration on EN→DE, of GaussianNet, LinearNet, (see Table 2) aPosNet without

Table 8: We compare the theoretical complexity and number of parameters per attention layer. $\hat{K}$ refers to the bidirectional context size. With the formulation of position-based attention, the attention energies can be pre-computed after training, resulting in different complexities between training and search.

Model	Parameters		Operations	
	Train	Search	Train	Search
Transformer	$4D^2$		$2N^2D + 4ND^2$	
Shaw et al. (2018)	$\hat{K}D + 4D^2$		$\hat{K}ND + 2N^2D + 4ND^2$	
FNet	$2D^2$		$N\log(N)D + D\log(D)N$	
GaussianNet	$2D^2$		$\hat{K}ND + 2ND^2$	
LinearNet	$HN^2 + 2D^2$		$N^2D + 2ND^2$	
LightConv	$H\hat{K} + 3D^2$		$\hat{K}ND + 3ND^2$	
gLinearNet	$HN^2 + 3D^2$		$N^2D + 3ND^2$	
aPosNet	$5D^2$	$HN^2 + 3D^2$	$2N^2D + 5ND^2$	$N^2D + 3ND^2$
rPosNet	$\hat{K}D + 4D^2$	$H\hat{K}N + 3D^2$	$\hat{K}ND + N^2D + 4ND^2$	$N^2D + 3ND^2$

gating, and rPosNet without gating (see Table 6) is similarly measured by BLEU but varies significantly in BLEURT and COMET. We analyzed translation samples of GaussianNet and LinearNet and observed that BLEU often falsely depicts translation quality when hypotheses have little overlap with the reference or changing a single word alters the meaning of the sentence. While the inaccuracies of BLEU are already known (Kocmi et al., 2021), we want to show exemplarily how BLEU would have misled our analysis. Without using BLEURT and COMET, we would have concluded that aPosNet and rPosNet would be equally good without gating and that the hard-coded weights of GaussianNet are as good as the learnable weights of LinearNet.

Table 9: Example failure cases on EN→DE in which BLEU depicts a misleading score. These inaccurate BLEU scores are best visualized when comparing GaussianNet and LinearNet. Both models achieve the same corpus-level BLEU score but differ significantly in BLEURT and COMET (see Table 2). The translations show that measuring the syntactical overlap between the hypothesis and reference translation is not an accurate measure of translation quality.

		BLEU	BLEURT	COMET
Source	Haigerloch: Focus on the Abendmahlskirche			
Reference	Haigerloch: Abendmahlskirche rückt in den Blickpunkt			
LinearNet	Haigerloch: Fokus auf die Abendmahlskirche	15.2	84.0	72.3
GaussianNet	Haigerloch: Focus on the Abendmahlskirche	15.2	35.6	−15.0
Source	Does he know about phone hacking?			
Reference	Weiß er über das Telefon-Hacking Bescheid?			
LinearNet	Weiß er von Telefonhacking?	15.8	80.2	72.5
GaussianNet	Kennt er über Telefon-Hacking?	17.0	38.2	8.8
Source	The new season in the Falkenberg "Blue Velvet" club has begun.			
Reference	Die neue Saison in der Falkenberger Discothek "Blue Velvet" hat begonnen.			
LinearNet	Die neue Saison im Falkenberg "Blue Velvet" Club hat begonnen.	33.1	75.3	85.7
GaussianNet	Die neue Saison im Falkenberg "Blue Velvet" hat begonnen.	53.7	72.2	74.5
Source	Finally, let's talk pumpkins.			
Reference	Aber kommen wir endlich zu den Kürbissen.			
LinearNet	Abschließend möchte ich noch auf die Kürbisse eingehen.	4.8	71.4	41.0
GaussianNet	Schließlich, lassen Sie uns reden Kürbisse.	5.5	36.0	−60.3
Source	A combined English literature and language course will be scrapped.			
Reference	Der kombinierte Kurs aus englischer Literatur und Sprache wird abgeschafft.			
LinearNet	Eine kombinierte englische Literatur und Sprachkurs wird verschrottet.	9.6	60.8	44.0
GaussianNet	A combined German literature and language course will be scrapped.	3.7	19.8	−42.8
Source	However, there was no sigh of relief to be heard from Ludwigsburg.			
Reference	Ein erstes Aufatmen war aus Ludwigsburg dennoch nicht zu vernehmen.			
LinearNet	Von Ludwigsburg war jedoch kein Seufzer der Erleichterung zu hören.	5.3	76.7	46.7
GaussianNet	Es gab jedoch keinen Seufzer der Erleichterung, von Ludwigsburg gehört zu werden.	3.7	45.1	−30.3
Source	Sayings come from the Bible			
Reference	Sprichwörter kommen aus der Bibel			
LinearNet	Sprichwörter stammen aus der Bibel	42.7	90.3	108.0
GaussianNet	Sayings kommen aus der Bibel	66.9	60.7	3.7
Source	Uwe Link has an offer for anyone who wants to set off in a carriage.			
Reference	Wer dann mit der Kutsche vorfahren will, für den hat Uwe Link ein Angebot.			
LinearNet	Uwe Link hat ein Angebot für jeden, der in einer Kutsche starten will.	9.0	70.0	59.0
GaussianNet	Uwe Link hat ein Angebot für jeden, der einen Wagen starten möchte.	8.5	46.3	−10.0
Source	Solicitors should uphold the highest standards of integrity and should instil trust and confidence in the public.			
Reference	Anwälte müssen die höchsten Standards an Integrität aufrechterhalten und in der Öffentlichkeit für Vertrauen und Zuversicht sorgen.			
LinearNet	Die Staatsanwälte sollten die höchsten Standards der Integrität wahren und Vertrauen in die Öffentlichkeit schaffen.	10.9	77.9	67.4
GaussianNet	Die Umweltschützer sollten die höchsten Standards der Integrität einhalten und Vertrauen und Vertrauen in die Öffentlichkeit schaffen.	12.2	53.6	−0.7