Highway Transformer: Self-Gating Enhanced Self-Attentive Networks

Yekun Chai[†] **Shuo Jin**[‡] **Xinwen Hou**[†]
[†]Institute of Automation, Chinese Academy of Sciences
[‡]University of Pittsburgh
chaiyekun@gmail.com shj42@pitt.edu

Abstract

Self-attention mechanisms have made striking state-of-the-art (SOTA) progress in various sequence learning tasks, standing on the multi-headed dot product attention by attending to all the global contexts at different locations. Through a *pseudo information highway*, we introduce a gated component *self-dependency units* (SDU) that incorporates LSTM-styled gating units to replenish internal semantic importance within the multi-dimensional latent space of individual representations. The subsidiary content-based SDU gates allow for the information flow of modulated latent embeddings through skipped connections, leading to a clear margin of convergence speed with gradient descent algorithms. We may unveil the role of gating mechanism to aid in the context-based Transformer modules, with hypothesizing that SDU gates, especially on shallow layers, could push it faster to step towards suboptimal points during the optimization process.

1 Introduction

Self-attention mechanism has lately attracted extensive interests due to its remarkable achievement on a wide range of sequence modeling applications, including natural language processing such as neural machine translation (Vaswani et al., 2017; Ott et al., 2018; Shaw et al., 2018), language modeling (LM) (Dai et al., 2019; Al-Rfou et al., 2019), self-supervised pretraining (Radford et al., 2018; Devlin et al., 2018; Lan et al., 2019); image generation (Parmar et al., 2018); deep reinforcement learning (Zambaldi et al., 2018; Vinyals et al., 2019), *etc*.

Holding the great promise of deep neural networks in language and images, Transformer capitalizes on the stacked multi-headed self-attention mechanism based on the conventional encoder-decoder architecture in a sequence-to-sequence (seq2seq) manner to learn the global soft signals without explicit recurrence mechanism. Multi-head dot product attention (MHDPA) not only underpins the parallel training of multiple heads but captures long-term dependencies across an arbitrarily long distance within the same context. In which separated multiple heads independently draw sub-level attentions within the latent semantic sub-space of a fixed dimension, where different heads are presumed to signal different meaning aspects implicitly (Vaswani et al., 2017). Additionally, residual connections between layers allow the deep tandem stack of multiple identical modules by impeding *degradation* problem during training (He et al., 2016). Thus Transformer architectures take the place of Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) to be the model solution to learning sequential data.

Recently, there have been plenty of works contending that gating mechanisms could play a vital role or even entirely substitute RNNs or Transformers to model language sequences. Dauphin et al. (2017) firstly claimed that non-recurrent networks are also highly competitive with conventional RNN-dominated models in LM. They proposed the hierarchical gated temporal convolution neural networks (CNNs) with Gated Linear Units (GLU) to replace the recurrent connections in RNNs and achieved strong performance with faster training speed. Gehring et al. (2017) integrated absolute positional embedding, multi-step attention, GLU, and residual connections into entirely convolutional models to outperform strong LSTM models in NMT and abstractive summarization tasks. Wu et al. (2019) applied dynamic convolutions using shared softmax-normalized filters of depth-wise on GLU-regulated inputs within a fixed reception field rather than global contexts, challenging the common self-attention-dominated intuition.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6887–6900
July 5 - 10, 2020.

However, all of the models, as mentioned earlier, adopt stacked CNNs rather than self-attention networks (SAN) to attend to the global contexts. It is well-known that CNNs are good at learning local-region features rather than long-term dependency, while SANs are adept in attending global dependencies. Context-based self-attention can capture the importance of relative relations under a valid context and is thus location-unaware. It focuses on the object-wise attention distributions between any two words but ignores the fundamental importance of feature-wise information.

Intuitionally, people need to consider not only the global contextual dependency but the meaning of individual words to comprehend the reading materials better. Grounding on this, we apply self-gating approaches on Transformer blocks for seq2seq modeling that combines gating units with skip-connections and Transformers to jointly take into account both the inner feature-wise importance and the relation-aware content-based attention distribution.

We adopt the self-dependency gating approach to intrinsically draw a binary importance ratio of itself and decide how much information of each feature to retain or remove. Our key contributions are:

- to illustrate that our self-dependency units on shallow Transformer layers could expedite the convergence speed during both the training and validation process without hyperparameter tuning.
- to support the claim that Transformer layers in different depth attend to information of different aspects, wherein bottom layers focus on local-range encodings. It substantiates the argument that the bottom layers of SAN can learn more in local contexts (Yang et al., 2018).
- to empirically prove that self-gating mechanisms are complementary to recurrence mechanisms in R-Transformer and Transformer-XL components.

2 Preliminaries

This section briefly introduces the related background of Transformer and Highway Networks.

SAN has been dominant in most SOTA sequence learning models, whose basic components consist of stacked Transformers modules. We conduct comparison experiments on the Transformer and its two variants, Transformer-XL (Dai et al., 2019) and R-Transformer (Wang et al., 2019).

2.1 Multi-head Dot Product Attention

Scaled dot product attention (DPA) (Vaswani et al., 2017) computes global attention weights between pairs within the context across an arbitrarily long distance, which could allow the simultaneous training and space-saving, impeding the drawbacks of sequential dependency of RNNs.

Given the input word representation $\mathbf{X} \in \mathbb{R}^{L\times dh}$, where L is the sequence length, d is the input dimension of each head and h is the number of attention heads, DPA uses the linear projection to acquire the query $\mathbf{Q}$, key $\mathbf{K}$ and value $\mathbf{V}$. Denoting splitted inputs for i-th head as $\mathbf{X}_i \in \mathbb{R}^{L\times d}$, where $i \in \{1, \cdots, h\}$, single-head self-attention can be calculated as:

$$\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i = \mathbf{X}_i\mathbf{W}_q, \mathbf{X}_i\mathbf{W}_k, \mathbf{X}_i\mathbf{W}_v \quad (1)$$

$$\text{head}_i = \text{softmax}\left(d^{-1/2}\mathbf{Q}_i\mathbf{K}_i^\top\right)\mathbf{V}_i \quad (2)$$

where learnable weights $\{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\} \in \mathbb{R}^{d\times d}$, $d^{-1/2}$ is a scaling factor to prevent the effect of large values. In LM tasks, attention weights before softmax function are masked to only attend to history sequences.

MHDPA (Fig 1a) linearly projects the single DPA into h heads and performs attention operation in parallel, to jointly learn different semantic meanings of different subspaces (Vaswani et al., 2017). MHDPA can be calculated as:

$$\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{head}_1 \circ \cdots \circ \text{head}_h]\,\mathbf{W_o} \quad (3)$$

where $\circ$ denotes the concatenation of h different heads, $\mathbf{W_o} \in \mathbb{R}^{dh\times dh}$ is the trainable weight.

2.2 Transformer

Absolute Positional Encoding Transformer applies sinusoidal timing signal as the absolute positional encoding (PE) and directly element-wise add the dense word embeddings $\mathbf{E} \in \mathbb{R}^{L\times dh}$ on the PE before feeding into Transformer modules:

$$PE_{(\text{pos},2i)} = \sin(\frac{\text{pos}}{10000^{2i/d}}) \quad (4)$$

$$PE_{(\text{pos},2i+1)} = \cos(\frac{\text{pos}}{10000^{2i/d}}) \quad (5)$$

$$\mathbf{X} = \mathbf{E} + PE(\mathbf{E}) \quad (6)$$

where 'pos' indicates the position of sequences, i denotes the order along the embedding dimension.

Given input representations $\mathbf{X}$, Transformer components with a sternward Layer Normalization (LN) is:

$$\mathbf{U} = LN(\mathbf{X} + \text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \quad (7)$$

$$\text{FFN}(\mathbf{U}) = FF\big(\text{ReLU}(FF(\mathbf{U}))\big) \quad (8)$$

$$\mathbf{O} = LN(\mathbf{U} + \text{FFN}(\mathbf{U})) \quad (9)$$

where Eq. 8 indicates the position-wise feed-forward networks (FFN), $\mathbf{O} \in \mathbb{R}^{L \times dh}$ represents the output of transformer layer. FF denotes the feed-forward fully-connected layer, ReLU is used as the non-linear activate function.

2.3 Transformer-XL

Transformer-XL (Dai et al., 2019) injected relative PE and segment-level recurrence to provide historical information for LM tasks.

Relative Positional Encoding Transformer-XL decomposed the dot product calculation of MHDPA, merged terms with similar meanings of positional bias, and reduced trainable weights with global positional semantics. It incorporated partial trainable parameters of relative sinusoidal PE in the MHDPA operation.

The Relative PE A^{rel} of Transformer-XL is:

$$a = \mathbf{Q}^\top \mathbf{K} \quad (10)$$

$$b = \mathbf{Q}^\top \mathbf{W}_{\mathbf{k},\mathbf{R}}\, \mathbf{R} \quad (11)$$

$$c = \mathbf{u}^\top \mathbf{K} \quad (12)$$

$$d = \mathbf{v}^\top \mathbf{W}_{\mathbf{k},\mathbf{R}}\, \mathbf{R} \quad (13)$$

$$A^{\text{rel}}(\mathbf{Q}, \mathbf{K}) = a + b + c + d \quad (14)$$

where $\mathbf{W}_{\mathbf{k},\mathbf{R}} \in \mathbb{R}^{d \times d}$, $\{\mathbf{u}, \mathbf{v}\} \in \mathbb{R}^d$ are trainable parameters. For each two positions i, j in the segment, $\mathbf{R}$ is sinusoidal encoding matrices between relative position $i - j$. The terms a, b, c, d in the Eq. 10, 11, 12, 13 represent the content-based addressing, content-dependent positional biases, global biases between different positions and the global positional biases, respectively.

Segment-level Recurrence In Transformer-XL, the previous hidden states are cached and reused to inject the history information and attend to contexts beyond a fixed length through multi-layer stacks. The MHDPA is computed as:

$$\mathbf{M}_\tau^{n-1} = \overbrace{SG(\mathbf{X}_{\tau-1}^{\mathbf{n-1}})}^{\text{stop gradient}} \circ \mathbf{X}_\tau^{\mathbf{n-1}} \quad (15)$$

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{X}_\tau^{\mathbf{n-1}}\mathbf{W}_{\mathbf{q}},\ \mathbf{M}_\tau^{n-1}\mathbf{W}_{\mathbf{k}},\ \mathbf{M}_\tau^{n-1}\mathbf{W}_{\mathbf{v}} \quad (16)$$

$$DPA(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = A^{\text{rel}}(\mathbf{Q}, \mathbf{K})\mathbf{V} \quad (17)$$

wherein the key and value $\mathbf{M}_\tau^{n-1}$ concatenate the previous memory $\mathbf{X}_{\tau-1}^{\mathbf{n-1}}$ with the current segment inputs $\mathbf{X}_\tau^{\mathbf{n-1}}$ for the τ-th segment in the n-th layer, SG means no backpropagation through the tensor.

2.4 R-Transformer

R-Transformer (Wang et al., 2019) employed short-range RNNs, termed *localRNNs*, to capture the positional information without explicit PEs. *localRNNs* take the recurrent connections within a local context, and shift right with one position at each time step. It can be seen as applying the RNN cells, such as LSTM, on the same receptive fields as the convolutional filters along the sequence direction.

$$\mathbf{X} = \text{localRNN}(\mathbf{E}) \quad (18)$$

$$\mathbf{O} = \text{Transformer-layer}(\mathbf{X}) \quad (19)$$

None of the above Transformer models explicitly consider the essential feature-wise information. We augment several gated units on the Transformer block of the models above and empirically illustrate the effectiveness of gating units on convergence acceleration.

2.5 Highway Networks

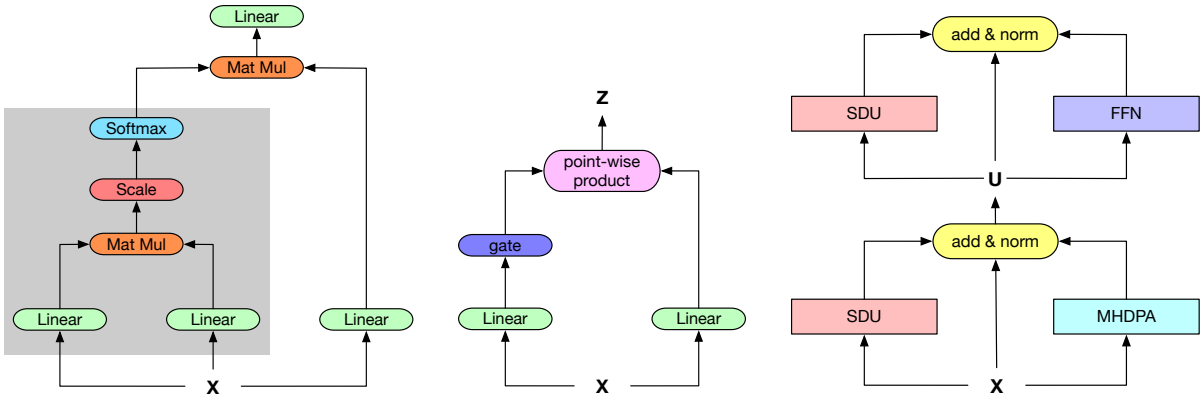
Let we define the non-linear transforms as H, T and C, Highway Network (Srivastava et al., 2015) is defined as:

$$\mathbf{O} = H(\mathbf{X}) \odot T(\mathbf{X}) + \mathbf{X} \odot C(\mathbf{X}) \quad (20)$$

where $T(\cdot)$ and $C(\cdot)$ denote transform and carry gates to control the input transformation, $\odot$ denotes the Hadamard product.

3 Gating Architecture

LSTM-styled gate units have been proven to be effective on sequence learning tasks (Dauphin et al., 2017; Gehring et al., 2017; Wu et al., 2019). We spontaneously wonder whether such gating mechanisms could help when augmenting the Transformer components.



(a) Multi-head dot product self-attention (b) Self-dependency units (c) SDU-augmented Transformer block

Figure 1: Illustration of MHDPA, SDU and SDU-enhanced Transformer block.

3.1 Self-Dependency Units

Similar to GLU (Dauphin et al., 2017) that adopts the inputs as sigmoidal gates, we apply the Self-Dependency Units (SDU) by taking full inputs as their respective self gates and computing the element-wise product upon themselves (Fig 1b).

$$T(\mathbf{X}) = \mathbf{\Psi}(\mathbf{X}\mathbf{W_1} + \mathbf{b_1}) \tag{21}$$

$$\text{SDU}(\mathbf{X}) = T(\mathbf{X}) \odot (\mathbf{X}\mathbf{W_2} + \mathbf{b_2}) \tag{22}$$

where $T(\mathbf{X})$ indicates the *transform* gate, $\mathbf{\Psi}$ is the gate function that confine the linear projection into a fixed range, $\{\mathbf{W_1}, \mathbf{W_2}\} \in \mathbb{R}^{d\times d}$ and $\{\mathbf{b_1}, \mathbf{b_2}\} \in \mathbb{R}^{d}$ are trainable parameters.

The element-wise gating function $\mathbf{\Psi}$ takes sigmoidal-curve functions to regulate the point-wise weights within a fixed region, which have a side effect of relative normalization. Specifically, the sigmoid function $\sigma(x) = 1/(1+\exp(-x))$ and its rescaled version $\tanh(x) = 2\sigma(2x) - 1$, where $x \in \mathbb{R}$.

We interpret the *tanh* function as an update gate, which can restrict the importance range into between -1 and 1, while the σ function bears a resemblance to the input gate in LSTMs to modulate how much information to retain at the feature-wise level.

3.2 Pseudo-highway Connection

MHDPA computes the multi-headed pairwise attention along the sequence dimension by measuring the distance between each word. It might overlook the fundamental importance of individual features. Rather than replacing MHDPA as gating and convolution operations in dynamic convolutions (Wu et al., 2019), we simply add a new branch of inputs to enrich the representations of residual connected MHDPA with augmented gating-modified encodings. The gated units are also supplemented on FFN modules to provide additional self-adaptive information flow (Fig 1c).

From other perspectives, SDU can be considered as a self-dependency non-linear activation function with dynamic adaptation. The self-gating augmented Transformer module is calculated as:

$$\begin{aligned}\mathbf{U} = LN\big(&\mathbf{X} + \text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \\ &+ \text{SDU}(\mathbf{X})\big)\end{aligned} \tag{23}$$

$$\mathbf{O} = LN\big(\mathbf{U} + \text{FFN}(\mathbf{U}) + \text{SDU}(\mathbf{U})\big) \tag{24}$$

where $\mathbf{U}$ and $\mathbf{O}$ represent the intermediate representation and outputs.

Pseudo-highway Transformer When we take σ gate as Ψ, we can have the similar format as highway networks:

$$\begin{aligned}\nabla[\mathbf{f}(\mathbf{X}) \odot \sigma(\mathbf{g}(\mathbf{X}))] = &\overbrace{\sigma(\mathbf{g}(\mathbf{X}))}^{\text{transform gate}} \odot \nabla\mathbf{f}(\mathbf{X}) \\ &+ \overbrace{\big(1 - \sigma(\mathbf{g}(\mathbf{X}))\big)}^{\text{carry gate}} \big(\sigma(\mathbf{g}(\mathbf{X})) \odot \mathbf{f}(\mathbf{X})\big)\end{aligned} \tag{25}$$

where the $\sigma(.)$ can be seen as the transform gate, while $(1-\sigma(.))$ can be seen as the carry gate. This could be regarded as a form of highway networks.

3.3 Variant Gated Connections

Highway Gate Similar to the highway networks (Srivastava et al., 2015), let $T(\mathbf{X})$ signal the *transform* gate and $(1 - T(\mathbf{X}))$ be the *carry* gate, we have the highway-network-like structures by regulating the encoding $f(\mathbf{X})$ with *transform* gate and controling $\mathbf{X}$ with *carry* gate. This is quite

similar to highway networks:

$$T(\mathbf{X}) = \sigma(\mathbf{X}\mathbf{W_1} + \mathbf{b_1}) \quad (26)$$

$$f(\mathbf{X}) = \mathbf{X}\mathbf{W_2} + \mathbf{b_2} \quad (27)$$

$$o(\mathbf{X}) = (1 - T(\mathbf{X})) \odot \mathbf{X} + T(\mathbf{X}) \odot f(\mathbf{X}) \quad (28)$$

$$\mathbf{U} = LN\big(o(\mathbf{X}) + \text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V})\big) \quad (29)$$

where Eq. 28 is the element-wise summation of highway networks, $o(\cdot)$ represents the intermediate output.

Gated MHDPA Similar to previous highway gates, we can apply the carry gate and transform gate on the attention and FFN units respectively. Thus we have:

$$o(\mathbf{X}) = (1 - T(\mathbf{X})) \odot \text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + T(\mathbf{X}) \odot f(\mathbf{X}) \quad (30)$$

$$\mathbf{U} = LN\big(o(\mathbf{X}) + \mathbf{X}\big) \quad (31)$$

Such gates can be regarded as dynamically adjusting the information flow between the feature-wise representations and SANs (Eq. 30).

4 Experiments and Results

We apply the gating mentioned above on Transformer variants described in section 2 on LM tasks and respectively make comparisons in terms of both the convergence process and the final performance. For fairness, we apply SDU components based on the same hyperparameters as the original paper[1]. Our code is available[2].

4.1 vs. Transformer / R-Transformer

We first evaluate the gating units on the Penn TreeBank (PTB) LM task. The SDU gates are added on Eq. 7, 9 for each Transformer block. All models in this section are trained on single NVIDIA Titan Xp GPU.

4.1.1 Char-level PTB

Hyperparameter and Training The gated components are evaluated on character-level PTB LM tasks (see Appendix A.1 for hyperparameter settings). The loss and bit per character (bpc) provide the metrics to evaluate the trained models. All models are trained with 100 epochs.

[1]Some results of baselines are slightly lower than those reported in original papers using the code obtained from authors but are within the limits of experimental error and variance.

[2]https://github.com/cyk1337/Highway-Transformer

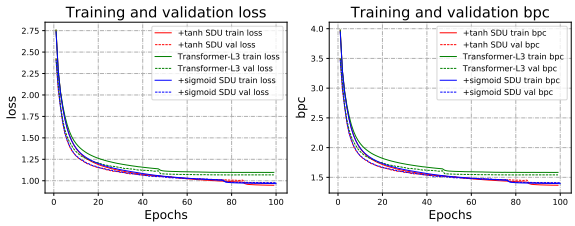


Figure 2: The 3-layer **Transformer**'s curve of training and evaluation performance on character-level PTB LM.

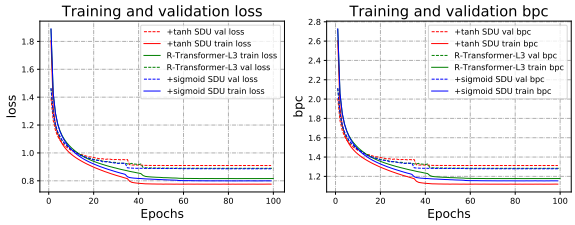


Figure 3: The 3-layer **RT**'s curve of training and evaluation performance on character-leval PTB LM task.

Results of Transformer As shown in Table 1, all the gating-enhanced models conspicuously surpass the performance of the loss and perplexity over the baseline on both training and validating set, revealing the positive influence of self-gating units in supporting Transformer blocks. Furthermore, Fig. 2 presents the beneficial effect of gating units in accelerating the convergence process in both training and evaluation set by a clear margin, validating the accumulative effect that our gating units bring out. In which SDUs with $\tanh$ gates (8.76% improvement) outperform the counterpart with sigmoid gates (8.2% improvement) in terms of the final perplexity on the test set.

model	eval loss	eval ppl	test loss	test ppl
T-L3	1.068	1.541	1.036	1.495
+σ SDU	0.9776	1.410⇓	0.950	1.371⇓
+tanh SDU	**0.9714**	**1.401**⇓	**0.945**	**1.364**⇓

Table 1: Performance of 3 Layer **Transformers** and SDU components on char-level PTB LM task. The best performance is marked bold.

Results of RT It can be seen in Fig. 3 that supplementing SDUs can increase the speed of the convergence process of training and evaluation, strengthening our previous claim. As for the final perplexity on the test set, σ-gate SDUs could achieve better than baselines while $\tanh$-gate SDUs perform a bit worse, as shown in Table 2. The influence of σ-gate SDUs might be owing to that σ function compresses the input into the dense non-zero ratios within $(0, 1)$ and results in stable variation range. In contrast, the zero-centered property and possibly

zeroed values of tanh may cause the corresponding units easier to be trapped into the premature convergence during the training process. Besides, σ gates have been empirically proved to be more stable than tanh gates in the follow-up experiments.

model	eval loss	eval ppl	test loss	test ppl
RT-L3	0.8896	1.283	0.867	1.250
+tanh SDU	0.9096	1.312	0.883	1.274
+σ SDU	**0.8863**	**1.279**⇓	**0.863**	**1.245**⇓

Table 2: Performance of 3 Layer **R-Transformers** and SDU components on char-level PTB LM task.

4.1.2 Word-level PTB

Hyperparameter and Training We compare the performance between 3-layer Transformer and R-Transformer (RT) with and without SDU gating units. Appendix A.1 illustrates the hyperparameter setup. All experiments are conducted with 100 epochs, and the loss and perplexity (ppl) values on the development set serve as evaluation metrics.

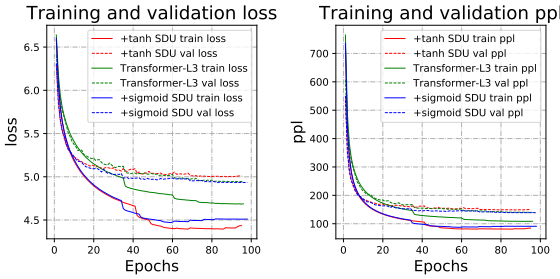


Figure 4: Loss and perplexity of 3-layer **Transformers** on the word-level PTB training and validation set.

model	eval loss	eval ppl	test loss	test ppl
T-L3	4.937	139.4	**4.87**	130.43
+σ SDU	**4.934**	**138.9**⇓	**4.87**	**130.30**⇓
+tanh SDU	5.001	148.5	4.94	139.53

Table 3: Performance of 3-layer basic **Transformer** (T-L3) and SDU components on word-level PTB LM.

Results of Transformer Figure 4 shows a noticeable downward trend on the evaluation performance (i.e., the validation loss and perplexity) of the attention model with tanh and sigmoid functions over the beginning 30 epochs, again indicating the convergence acceleration effect of our gated units. Also, σ-gate enhanced models outmatches the baseline on the test perplexity, but models with tanh gates reach into a plateau untimely. As for the training curves, Transformers with SDUs have seen a remarkably sharper fall in comparison with the baseline model over all the training period.

Results of RT As in Fig. 5 and Table 4, models with SDUs entirely surpass the performance of the baseline involving both the convergence speed and perplexity on the test set. Similar to the word-level R-Transformer, tanh-gate SDUs behave a bit better than the counterpart with sigmoid gates, both showing stable curvatures of convergence.

model	eval loss	eval ppl	test loss	test ppl
RT-L3	4.58	97.63	4.53	92.31
+σ SDU	4.53	92.91⇓	4.48	87.88⇓
+tanh SDU	**4.50**	**89.97**⇓	**4.44**	**84.92**⇓

Table 4: The performance of 3-layer R-Transformers (RT-L3) and SDU components on word-level PTB LM.

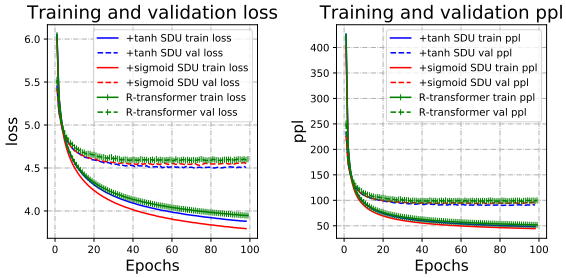


Figure 5: Loss and perplexity of 3-layer **RT** on the word-level PTB training and validation sets.

4.2 Sub-total

To sum up, gating units have empirically expedited the convergence of Transformer blocks due to the enrichment of self-regulated features with skip-connections. It can be seen that σ-gate presents the stability to bear a hand to reach the plateau without hurting the test performance, but tanh-gate seems to be task- and data-dependent and could be better than σ-gate SDUs in some circumstances. We can see that our proposed gated units are complementary to the recurrent connections in RNNs and can boost the performance based on *localRNN*-encoded representations.

In the following experiment, we check whether it is necessary to apply gates on all the layers and probe the effect of SDU variants (i.e., "*highway gate*" and "*gate MHDPA*"). Due to the small size of PTB, we experiment on a larger LM dataset *enwik8* and adopt the impressive Transformer-XL, one of the vital variant structures used in XLNet (Yang et al., 2019).

4.3 vs. Transformer-XL

Hyperparameter See Appendix A.3 for detailed hyperparameter settings.

4.3.1 Results of 6-layer Transformer-XL

It is noticeable that Transformer-XL models with different gating variants all outperform the baseline with different margins in terms of both performance and convergence speed, as shown in Table 5. Fig. 6 shows that SDUs benefit the convergence and validation performance compared with baselines. Among which σ-gate SDUs ranked top by achieving 3.1% improvement of bpc on the dev set, followed by *gates with tanh*, *gated MHDPA*, *highway gate* with 2.7%, 1.8%, 1.7% advance respectively. We attribute such improvements to the augmented refined representations learned by our gated units, preventing the basic self-attention blocks from purely considering the contextual dependency. It is also illustrated that SDUs do not conflict with recurrence mechanisms in Transformer-XL.

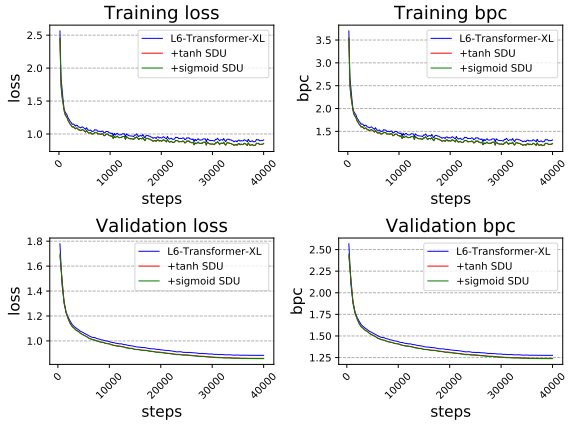


Figure 6: The comparison between 6-layer Transformer-XL with adding different SDUs.

4.3.2 Ablation Study

6-layer Transformer-XL To probe whether it is required to augment SDUs on each Transformer layer, we supplement gates on layer 1-3, layer 3-6, and layer 1-6 but removing gates on FFN components (denoted "\FFN") as in Table 5 (see Fig. 8 in Appendix B for detailed convergence curvatures). We find that supplementing tanh-gates on the bottom three layers contribute most to the overall performance while tanh-gates on the top three layers could hinder the test set performance. Low-level Transformer blocks can capture the information from localness while top layers usually focus on the global long-range dependency (Yang et al., 2018).

model	eval loss	eval bpc	test loss	test bpc
L6-XL	0.8843	1.276	0.86	1.24339
+tanh SDU	0.8602	1.241⇓	**0.84**	1.21424⇓
+σ SDU	**0.8577**	**1.237**⇓	**0.84**	**1.21123**⇓
+highway gate	0.8692	1.254⇓	0.85	1.22177⇓
+gated MHDPA	0.8682	1.253⇓	0.85	1.22398⇓
Ablation study				
+tanh L1-6\FFN	0.8720	1.258⇓	0.85	1.22866⇓
+tanh L1-3	**0.8660**	**1.249**⇓	**0.85**	**1.22039**⇓
+tanh L3-6	0.8852	1.277⇓	0.86	1.24420⇓
+σ L1-6\FFN	0.8752	1.263⇓	0.85	1.23332⇓
+σ L1-3	0.8792	1.268⇓	0.86	1.23589⇓
+σ L3-6	0.8843	1.276⇓	0.86	1.24261⇓

Table 5: Results of **6-layer Transformer-XL** (L6-XL) and augmented SDUs with different settings. "+σ" *L1-6\FFN* represents adding σ-SDUs on MHDPAs of 1-st to 6-th layers but not on FFN sublayers.

Thus gates on bottom layers could aid in learning syntax and superficial representations to some extent. It also indicates that our gates may be beneficial for the encoding of low-level fine-granularity representations rather than semantic meaning regulation on high-level layers.

12-layer Transformer-XL Previous experiments are all conducted on shallow models and illustrate the positive effects. To investigate the performance on deep stacked models, we further extend our trials to 12-layer Transformer-XL. All hyperparameters are the same as 6-layer Transformer-XL, as shown in Appendix A.3. Each Model is trained 400k steps for more than 100 hours on 4 x GeForce 2080Ti GPUs in parallel.

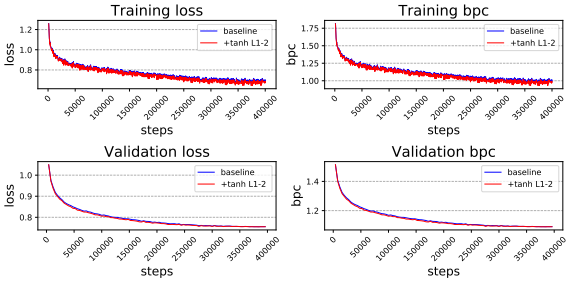


Figure 7: The comparison between **12-layer Transformer-XL** with and without tanh gated units on bottom two layers.

The experimental results illustrate that SDU components have contributed to expediting the convergence during training (see Fig. 9 and 10 in Appendix C for details). But supplementing gated units on each Transformer block could encounter the premature convergence phenomenon. It is also observed that adding the bottom few layers with gated units could strengthen the convergence pro-

model	eval loss	eval bpc	test loss	test bpc
L12-XL	0.7554	1.090	0.74	1.07160
Ablation study				
+tanh L1-12	0.7919	1.143	0.78	1.12797
+tanh L1-6	0.7623	1.100	0.75	1.08234
+tanh L1-3	0.7558	1.090	0.74	1.07140⇓
+tanh L1-2	0.7548	**1.089**⇓	0.74	**1.06904**⇓
+tanh L1	0.7549	**1.089**⇓	0.74	1.06960⇓
+tanh L6-12	0.7572	1.092	0.74	1.07313
+tanh $\backslash FFN$	0.7734	1.116	0.76	1.09920
+σ L1-12	0.7752	1.118	0.77	1.10462
+σ L1-6	0.7635	1.101	0.75	1.08283
+σ L1-3	0.7580	1.094	0.74	1.07383
+σ L1-2	0.7552	1.090	0.74	1.07148⇓
+σ L1	0.7557	1.090	0.74	1.07157⇓
+σ L6-12	0.7585	1.094	0.75	1.07607
+σ $\backslash FFN$	0.7647	1.103	0.75	1.08652
+highway gate	0.7784	1.120	0.77	1.10922
+gated MHDPA	0.7741	1.117	0.76	1.10292

Table 6: Final results of **12-layer Transformer-XL** (XL-L12) and augmented SDUs with different settings.

cess without impeding the final performance, as shown in Table 6. It is observed from Fig. 7 that tanh-gates on the bottom two layers promote the convergence process and further improve the bpc performance on the dev and test set.

Interestingly, the performance does not follow a positive correlation with the increase of gated layer numbers. We can see that enriching the bottom 2 layers with tanh and σ gated functions (denoted "+tanh L1-2" and "+σ L1-2" in Table 6) could impressively benefit for the convergence on both training and evaluation process and even marginally increase the final test bpc (see Fig. 9 and Fig. 10 in Appendix C for details). Therefore, the lower layers benefit more from our proposed gated units than higher layers, again illustrating that SDUs could enhance feature-wise information on shallow layers of deep-stacked Transformer components.

4.4 Gating Mechanism Analysis

It can be concluded that gating units could boost the convergence, especially on low-level layers. Enhancing the bottom layers of deep-stacked models may result in faster convergence of optimization. This may be owing to that SDU gates can enrich the original representations with adaptive self-dependency encodings. The final hidden state can be regarded as a revised representation that incorporating additional self-attentive features.

Meanwhile, we find that supplementing SDU gates does not increase much of the time cost in comparison with baselines. Instead, the total running time of each experimental setting is quite similar. We summarize the training time costs of 6-layer Transformer-XL as table .7.

model	time cost (hour)
xl-L6	21.16
+tanh SDU	21.45
+σ SDU	21.87
+ highway gate	21.93
+gated MHDPA	21.10

Table 7: Summary of training time costs of 6-layer Transformer-XL.

It is argued that low-level transformers learn the local-region information while high-level layers pay more attention to global dependencies (Yang et al., 2018). Our experimental results could verify that gated representation on bottom layers can strengthen the performance by introducing additional gated encodings on localness.

Further, the visualization of learned gate bias parameters of 6-layer and 12-layer models, as shown in Fig. 11 in Appendix D.1, presenting the layer separation with the increase of layer depth. It has seamlessly verified our previous hypothesis that SDU on shallow layers could promote the learning process and attend to different information with top layers. The scatter plot of Fig. 12 in Appendix D.2 indicates that gates on different sublayers learn from different aspects in the identical representation space.

SDUs calculate the output by regulating the information flow of inputs conditioned on themselves. Given the hidden dimension of d, the additional cost of trainable parameters on each SDU unit in our experiments is $O(2d(d+1))$. Meanwhile, convolutions along the sequence direction can substitute fully-connected feedforward SDU to curtail the extra parameter cost. Such gating units equip good scalability to attach to different Transformer structures with only minor modification of implementation.

The gradient of our SDU components is:

$$\nabla[\mathbf{f}(\mathbf{x}) \odot \Psi(\mathbf{g}(\mathbf{x}))] = \nabla\mathbf{f}(\mathbf{x}) \odot \Psi(\mathbf{g}(\mathbf{x})) \quad (32)$$
$$+ \mathbf{f}(\mathbf{x}) \odot \nabla\Psi(\mathbf{g}(\mathbf{x})) \quad (33)$$

where $\mathbf{f}, \mathbf{g}$ are linear projections and Ψ takes tanh or σ function. The addition operation of two terms provides an unimpeded information flow, which can be regarded as a multiplicative skip connection (Dauphin et al., 2017) while the second term is usually vanishing due to the derivative of the gating

function Ψ. Based on the experimental results, we hypothesize that it could accelerate the optimization process to move towards a local minimum.

5 Related Work

In recent years, there have been plenty of works adopting gating units into CNNs to help learn sequential information. Dauphin et al. (2017) proposed stacked gated CNNs by incorporating GLUs into the 1-dimensional convolution operation, achieving the competitive results in comparison to recurrent models on LM tasks. Based on this, Gehring et al. (2017) augmented the attention mechanism together with GLUs on the convolutional structures, also surpassing the deep LSTMs on NMT tasks. Recently, dynamic convolutions were used to replace MHDPA components in Transformers entirely and also get the impressive results on the WMT-14 dataset (Wu et al., 2019).

Amounts of works employed gating mechanisms to modulate self-attention sublayers. Gated-Attention Reader (Dhingra et al., 2016) introduced gated attention by computing gates on the query encoding to interact with document representations for reading comprehension. Zhang et al. (2018) replaced the first layer of Transformer decoding stacks with an average attention layer by computing forget gates using averaged preceding contextual encodings to regulate the current state information. Distance-based SAN (Im and Cho, 2017) and DiSAN (Shen et al., 2018) put a fusion gate to aggregate the representations after the multi-dimensional self-attention block for natural language inference. Lai et al. (2019) proposed a gated self-attention memory network with aggregated interactions between input sequences and context vectors for answer selection of question answering.

Notably, our SDU bears a resemblance to the activation Swish (Ramachandran et al., 2017) in terms of the equation format. Both of them use the sigmoidal function and self-gating mechanism. However, Swish controls the input gated on itself in a tandem way while the proposed SDU applies the gate after a linear projection and performs using a shunt connection in Transformer stacks.

6 Conclusion and Future Work

Gating-enhanced architecture enjoys both the advantage of MHDPA and self-regulated gating mechanism, allowing for the *pseudo*-highway information flow for better convergence by elastically introducing a few trainable parameters. It outperforms or matches the performance of common Transformer variants without hyperparameter tuning. It is empirically proved that self-gating units on shallow layers could provide more internal representations of importance and significantly benefit for convergence. This also supports the argument that different levels of Transformer components attend to different semantic aspects while lower levels pay more attention to local regions. In the future, it is necessary to interpret the semantics that Transformer layers in different depths can convey, which is beneficial for the computing-efficiency.

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3159–3166.

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv: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*, pages 933–941. JMLR. org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*.

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*, pages 1243–1252. JMLR. org.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jinbae Im and Sungzoon Cho. 2017. Distance-based self-attention network for natural language inference. *arXiv preprint arXiv:1712.02047*.

Tuan Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2019. A gated self-attention memory network for answer selection. *arXiv preprint arXiv:1909.09696*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. *arXiv preprint arXiv:1802.05751*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5.

Zhiwei Wang, Yao Ma, Zitao Liu, and Jiliang Tang. 2019. R-transformer: Recurrent neural network enhanced transformer. *arXiv preprint arXiv:1907.05572*.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*.

Baosong Yang, Zhaopeng Tu, Derek F Wong, Fandong Meng, Lidia S Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. *arXiv preprint arXiv:1810.10182*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. 2018. Deep reinforcement learning with relational inductive biases.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention network. *arXiv preprint arXiv:1805.00631*.

A Experimental Setup Details

A.1 Hyperparameter Settings for RT on Char-level PTB

For RT on char-level PTB, we adopt the batch size of 16, gradient clipping with maximum L2 norm of 0.15, layer number of 3, hidden dimension of 512, the sequence length of 400 in char-level, the dropout rate for sublayer connection of 0.15, 8 heads for MHDPA, the initial learning rate of 2, SGD optimizer with linear decay, layer number of 3 in both Transformer and RT models. Weights are initialized with uniform distribution $\mathbf{w} \sim U(-0.1, 0.1)$ and biases are all initialized as 0s. The size of GRU cells in *localRNNs* is set to 7 in RT.

A.2 Hyperparameter Settings for RT on Word-level PTB

We use the dropout rates of 0.35 and 0.15 for sublayer connections and word embeddings, the initial learning rate of 2, gradient clipping with the maximum L2 norm of 0.35, the hidden dimension of 128, 8-head attention, sequence length of 80 in both Transformer and RT. The weights are initialized with uniform distribution $U(-0.01, 0.01)$, and the biases are constant 0s. The optimizer is stochastic gradient descent (SGD) with annealed decay. The *localRNN* context size for LSTM cells is set to 9 in RT.

A.3 Hyperparameter Settings for Transformer-XL on enwik8

We use layer number of 6, 8 heads for MHDPA with hidden size 64 for each head, hidden size of 2,048 in FFN components, the dropout rate of 0.1 in FFN, embedding size of 512, learning rate 0.00025, memory length of 512, batch size of 22, Adam optimizer without the warm-up strategy. We initialize weights under the Gaussian $\mathcal{N}(0, 1)$ and biases as 0s.

B Experimental Results of 6-layer Transformer-XL

Fig 8 displays all the experimental curvatures with different SDU settings on 6-layer Transformer-XL.

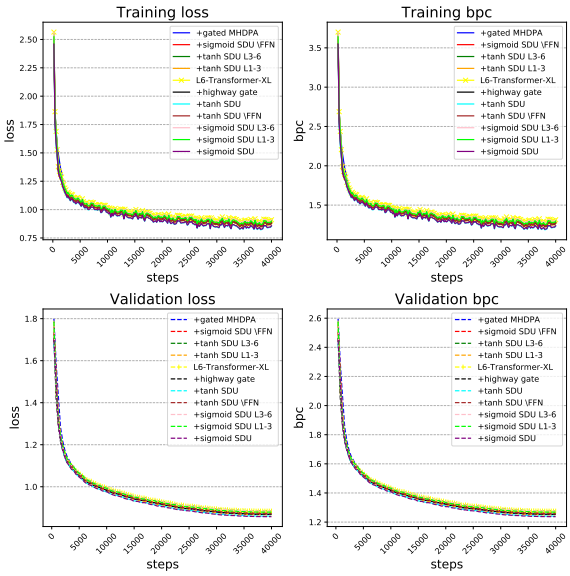


Figure 8: The performance of **6-layer Transformer-XL** experiments with various settings of gated units.

C Experimental Results of 12-layer Transformer-XL

C.1 Transformer-XL v.s. +tanh Gates

Fig. 9 shows the curve of tanh-gate enhanced Transformer-XL during the training and evaluation process. Adding tanh-gates on the first few layers greatly boost the convergence performance in both the training and evaluation process. Among which "+tanh L1-2" presents a rapid convergence trend and marginally outperforms the baseline performance.

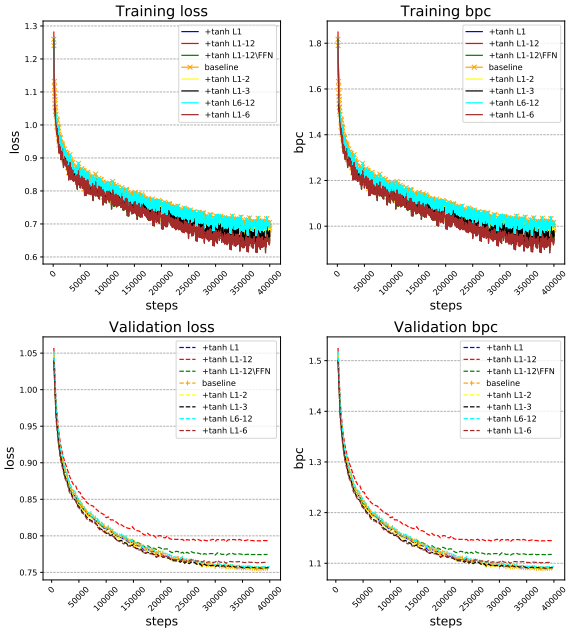


Figure 9: The performance of **12-layer Transformer-XL** experiments augmenting tanh gated units.

C.2 Transformer-XL v.s. +sigmoid Gates

Fig. 10 illustrates the performance of Transformer-XL augmented with σ gates. The sigmoid-gated Transformer-XL has showed a similar trend as $\tanh$ gates in Fig. 9.

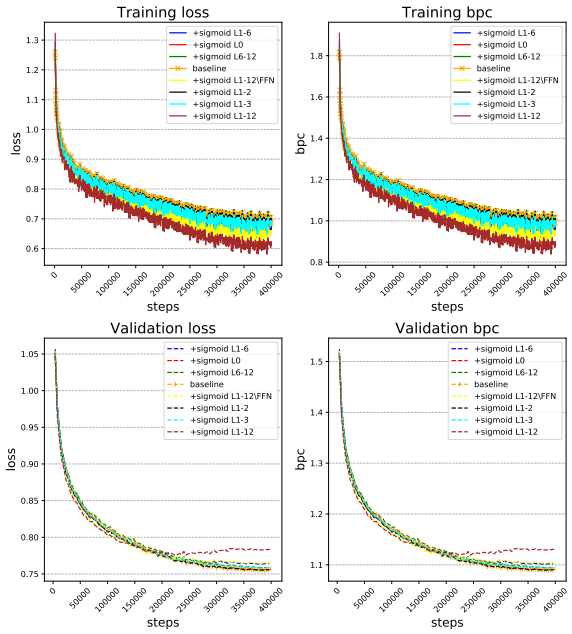


Figure 10: The performance of **12-layer Transformer-XL** experiments augmenting σ gated units.

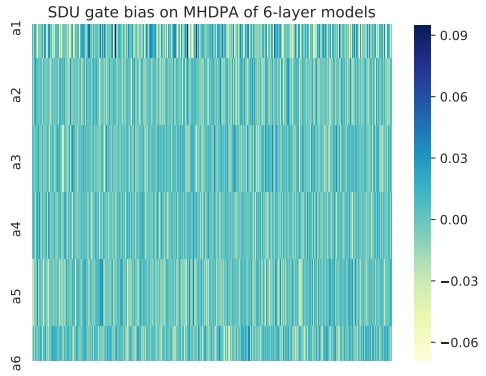
D Plot of Gate Biases of Transformer-XL

D.1 Heatmap Visualization

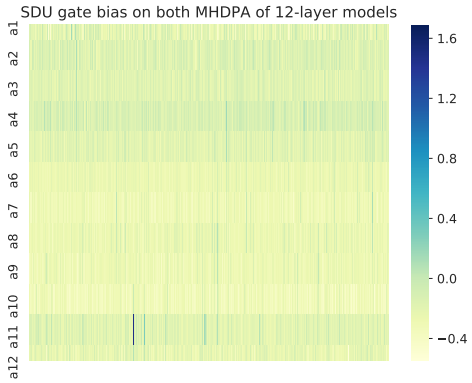
Fig 11 witnesses the visualization of learned biases, which are all initialized as zeros at the beginning. Obviously, the trainable biases of SDU gates perform quite different between on MHDPA and FFN sublayers as in Fig. 11a, 11c for 6-layer models and Fig. 11b, 11d for 12-layer models. Also, the gate biases are similarly distributed on all of the 6 layers, as in Fig. 11e, while showing the layer separation on the bottom few transformer layers as shown in Fig. 11f. This also verifies the experimental evidence that SDU gates on 6-layer models all positively influence the final test performance, but those only on the previous few layers of 12-layer transformers could have better results on both convergence speed and the final test bpc.

D.2 Scatter Visualization

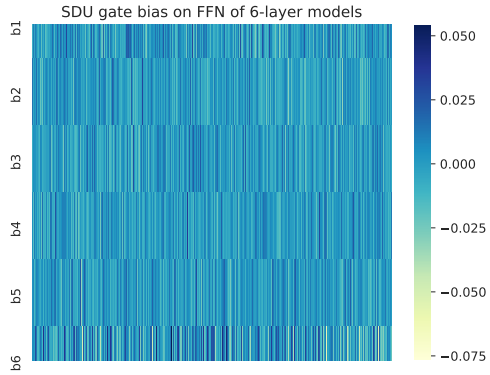
Fig. 12 illustrates the uniform distribution on both 6-layer and 12-layer Transformer-XL models. Due to the existence of residual connections, the representation space can be seen as the same. Hence the evenly distributed gate biases may learn from different aspects accordingly, which also matches our common intuition.



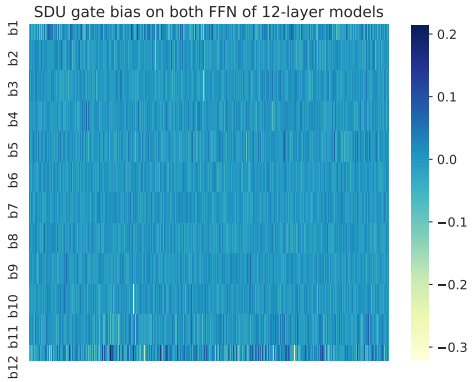
(a) Plot of gate biases on MHDPA of 6-layer models.



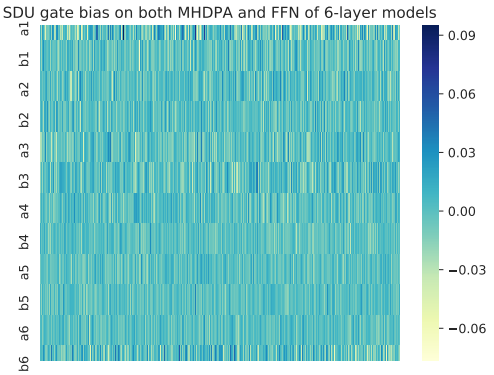
(b) Plot of gate biases on MHDPA of 12-layer models.



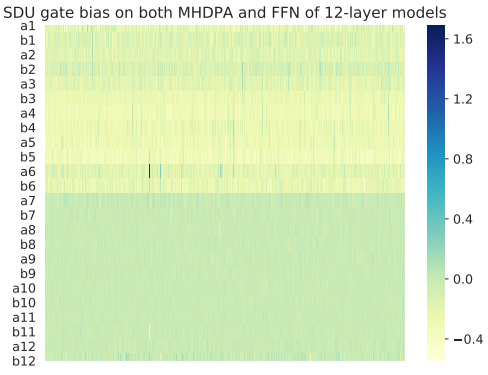
(c) Plot of gate biases on FFN of 6-layer models.



(d) Plot of gate biases on FFN of 12-layer models.

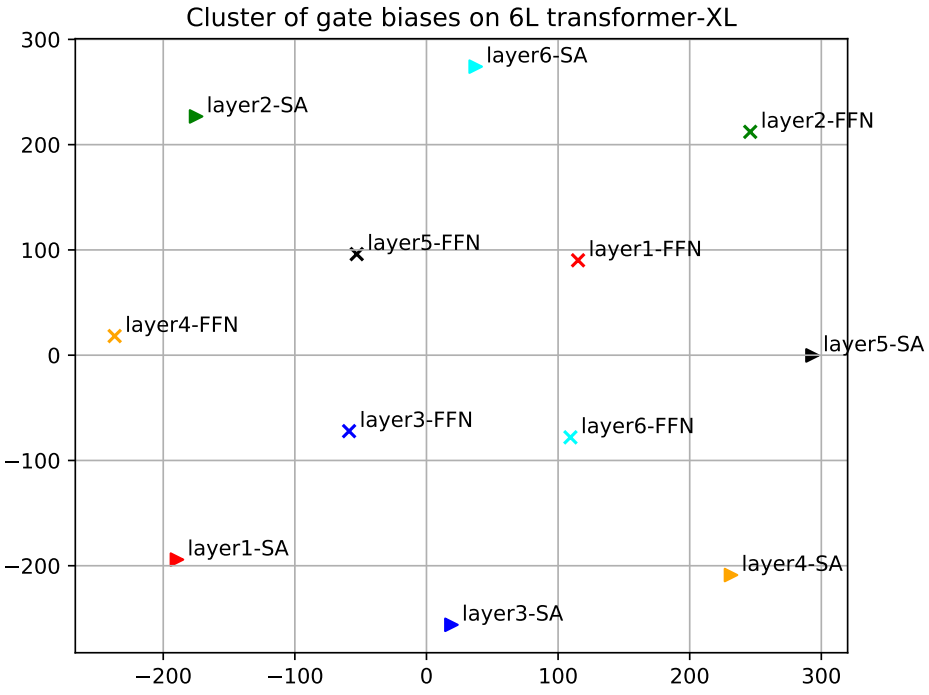


(e) Plot of gate biases on all sublayers of 6-layer models.

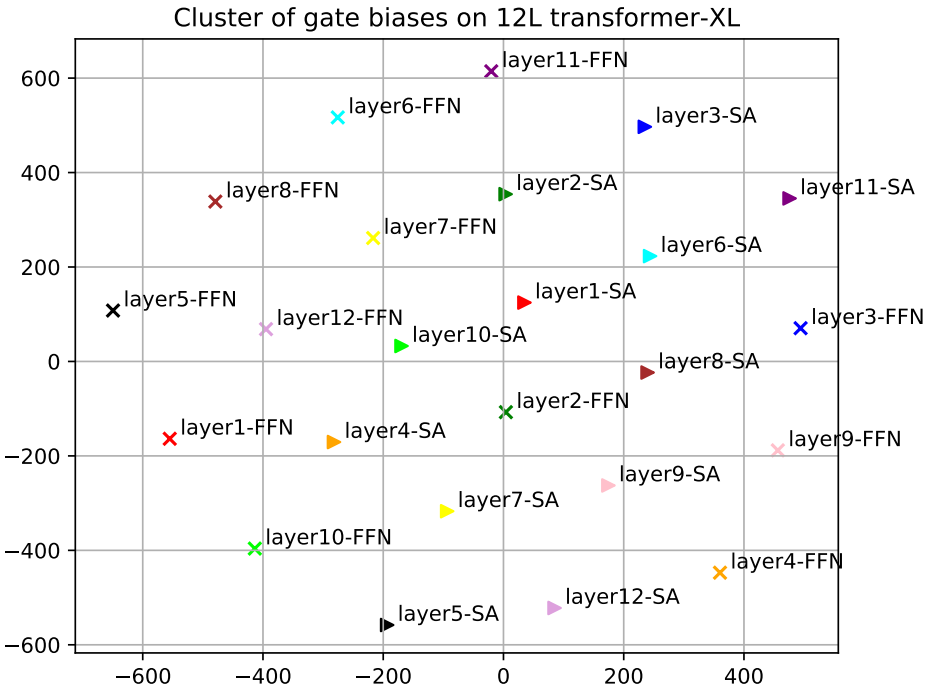


(f) Plot of gate biases on all sublayers of 12-layer models.

Figure 11: The heatmap visualization of **learnable biases** (i.e., $\mathbf{b_1}$ in Eq. 21) on σ gate units of 6-layer (left column) and 12-layer (right column) **Transformer-XL** models, where vertical axises represent the layer number of our models, and "a1" and "b3" denote the 1-st MHDPA sublayer and 3-rd FFN sublayer, respectively. All gate biases are initialized as 0s with 512 dimension of each.



(a) Plot of bias distributions on 6-layer models.



(b) Plot of bias distributions on 12-layer models.

Figure 12: Scatter visualization of SDU gate biases on 6-layer and 12-layer Transformer-XL, where "layer2-SA" denotes the gate bias on 2-nd self-attention sublayer. We employ t-Distributed Stochastic Neighbor Embedding (t-SNE) to reduce the dimension from 512 to 2.