# Attention is Kernel Trick Reloaded

**Gokhan Egri**
Harvard University
gegri@g.harvard.edu

**Xinran (Nicole) Han**
Harvard University
xinranhan@g.harvard.edu

## Abstract

The transformer architecture achieves superior performance on many learning tasks, such as machine translation and visual question answering. It can also be applied to a numerical sequential data such as time series data for multi-step predictions. At the core of transformer is its attention mechanism, which computes the similarity between instances and encourages the model to focus on important selective features. Inspired by recent RBF kernel formulation of the attention mechanism and the classical kernel trick literature in machine learning, we aim to understand how various kernels such as the locally periodic (LP) kernel, linear kernel etc. affect the performance of the transformer. We will first derive the mathematical formulation for those kernels, and then test their empirical performance on both the machine translation task and the time-series prediction task. Experiment results show that our proposed attention modules not only achieve competitive performance on the complex translation task but improve model performance on long- and short-range time-series prediction, suggesting that varying the attention kernel can be a promising future research direction.[1]

## 1 Introduction

With the advent of powerful NLP models such as GPT-3 [1] and BERT [3], transformers are quickly becoming the standard architecture for large-scale machine learning, not only for NLP tasks but also for similar problems in images and other media [4, 2, 12]. This exponential growth in popularity is largely attributable to the transformers' ability to benefit from and scale with the extensive amounts of data available on the internet. Different from conventional models in NLP such as RNN and LSTM that relies on sequential computation, transformer paralleled text processing by using self-attention and feed-forward networks as its building blocks.

Broadly, attention is used as a means to gauge the inter-relation between different parts of the input such that at each iteration, the network learns to "attend" to only the relevant sections of the input at that specific time step. The implementation of attention usually consists of three steps:

1. From the embedding of each word, we create a *Query* vector, a *Key* vector and a *Value* vector. These vectors are computed by multiplying the embedding with matrices with trainable parameters.
2. For each word in the sequence, we compute a self-attention score with respect to another target word by taking the dot product of the query vector and the key vector of the word we are inspecting. This score indicates how much focus we want to put on the target word.
3. Then we divide the score by some normalization factor for more stable gradients. The original paper from Vaswani *et al.* divides the score by the square root of the dimension of key vectors. They also apply a softmax layer to obtain a probability distribution over each token that sums up to 1.

[1] We release our implementation of kernel attention publicly as a PyPi package `pykernsformer` (`https://pypi.org/project/pykernsformer/`) for future work.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Recent papers from [10][9] presented a kernel formulation of the attention mechanism in transformer. Specifically, the attention in transformer is viewed as a product of the Radial Basis Function (RBF) kernel between instances and the exponential $\ell^2$ norm for each instance. Inspired by this formulation, we are interested in how different attention kernels may impact the performance of transformer on different data types. We will first derive the mathematical expressions of various kernels, and then evaluate them on two tasks: (1) English to Chinese sentence translation and (2) Time-series prediction.

## 2 Related Work

The kernel trick [8][5] has been a crucial component of many machine learning models, such as Support Vector Machines (SVM) and Multi-layer Kernel Machines (MKM). The usage of positive definite kernel functions generalizes distance and similarity measures to the high-dimensional embedding space and provides elegant solutions for many learning problems.

While attention in transformers has been identified as an RBF kernel, prior studies on the attention component of transformer [13][7] have mostly focused on improving the speed of kernel computation but not experimenting with other types of kernels. The work that is most similar to ours is [9], where the authors generalized the attention module with an implicit kernel function and an $\ell^p$ norm and demonstrated improved performance on language tasks such as sentence classification and translation. However, their derivations and experiments revolve around variations of the RBF kernel and does not consider the more classical kernels in previous literature, such as the periodic kernel and the rational quadratic kernel.

Besides applications in natural language tasks, recent works [12] have attempted to apply transformer to the task of time series forecasting. As classical kernel methods and signal processing literature show that certain kernels work well with particular signal patterns or data types, our work also explores whether different kernels integrated with the transformer architecture will yield similar results for different signal types.

## 3 Methodology

### 3.1 Attention kernel decomposition

Vaswani *et al.* [11] attention is defined as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

for query $Q_{[n_Q, d_k]}$, key $K_{[n_K, d_k]}$ and value $V_{[n_K, d_v]}$ matrices.

Our work builds on the observation that this attention formulation can be decomposed into a weighted kernel calculation between individual query and key vectors $q^i_{[1,d_k]}$ and $k^i_{[1,d_k]}$ shown below [9].

**Proposition 1** *Let $a_{ij}$ be $\frac{1}{Z_1(a)} exp(\frac{q_i k_j^T}{\sqrt{d_k}})$ where $Z_1(a)$ is a normalizing constant. Then $a_{ij}$ has the form:*

$$a_{ij} = \frac{1}{Z_1(a)} \times exp(\frac{-||q_i - k_j||_2^2}{2\sqrt{d_k}}) \times exp(\frac{||q_i||_2^2 + ||k_j||_2^2}{2\sqrt{d_k}}) \tag{2}$$

*where we used the fact that*

$$||q_i - k_j||_2^2 = ||q_i||_2^2 + ||k_j||_2^2 - 2q_i k_j^T \tag{3}$$

Notice that in Equation 2, the first exponential term corresponds to an RBF-kernel distance between the vectors $q_i$ and $k_j$ with $l = \sqrt[4]{d_k}$. The second exponential term acts as a magnitude term which weighs each query-key pair based on $\ell_2$-norm.

### 3.2 Kernel construction

Following this decomposition, we construct attention equations for the following well-known kernels:

1. Linear kernel ($k_L$)
2. Periodic kernel ($k_P$)
3. Locally periodic kernel ($k_{LP}$)
4. Rational quadratic kernel ($k_{RQ}$)

**Linear kernel ($k_L$).** The linear kernel has the form

$$k_L(x, x') = (x - c)(x' - c) \tag{4}$$

where $c$ is a parameter specifying the origin of the non-stationary kernel.

**Proposition 2** *Let $a_{ij}$ be $\frac{1}{Z_2(a)} k_L(q_i, k_j)$ where $Z_2(a)$ is a normalizing constant. Then $a_{ij}$ has the form:*

$$\begin{aligned} a_{ij} &= \frac{1}{Z_2(a)} \times (q_i - \mu)(k_j - \mu)^T && (5)\\ &= \frac{1}{Z_2(a)} \times (q_i k_j^T - q_i \mu^T - \mu k_j^T + \mu\mu^T) && (6)\\ &= \frac{1}{Z_2(a)} \times (q_i k_j^T) + \frac{1}{Z_2(a)} \times (\mu\mu^T - q_i \mu^T - \mu k_j^T) && (7) \end{aligned}$$

In implementation, this corresponds to

$$Attention_{Linear}(Q, K, V) = \frac{QK^T}{\sum_k QK^T} V \tag{8}$$

for $\mu = [0]_{[1,d_k]}$.

**Periodic kernel ($k_P$).** The periodic kernel has the form

$$k_P(x, x') = \exp\left( -\frac{2\sin^2(\pi \frac{||x - x'||_2}{p})}{\ell^2} \right) \tag{9}$$

where $\{\ell, p\}$ are parameters specifying the length-scale and periodicity of the kernel.

**Proposition 3** *Let $a_{ij}$ be $\frac{1}{Z_3(a)} k_P(q_i, k_j)$ where $Z_3(a)$ is a normalizing constant. Then $a_{ij}$ has the form:*

$$\begin{aligned} a_{ij} &= \frac{1}{Z_3(a)} \times \exp\left( -\frac{2\sin^2(\pi \frac{||x - x'||_2}{p})}{\ell^2} \right) && (10)\\ &= \frac{1}{Z_3(a)} \times \exp\left( -\frac{2\sin^2(\pi \frac{\sqrt{||q_i||_2^2 + ||k_j||_2^2 - 2q_i k_j^T}}{p})}{\ell^2} \right) && (11) \end{aligned}$$

Notice that here, we are not able to decompose the expression into kernel and magnitude terms as in Equation 2 due to the square root. Neglecting the magnitude terms expectedly hurts convergence.

We circumvent this problem by normalizing our query and key vectors such that $||q_i||_2 = ||k_j||_2 = 1$ which yields

$$a_{ij} = \frac{1}{Z_3(a)} \times \exp\left(-\frac{2\sin^2(\pi\frac{\sqrt{2-2q_ik_j^T}}{p})}{\ell^2}\right) \tag{12}$$

In implementation, this corresponds to

$$Attention_{Periodic(p)}(Q,K,V) = softmax(-\frac{2\sin^2(\pi\frac{\sqrt{2-2QK^T}}{p})}{\sqrt{d_k}})V \tag{13}$$

**Locally periodic kernel ($k_{LP}$).** The locally periodic kernel has the form

$$k_{LP}(x,x') = \exp\left(-\frac{2\sin^2(\pi\frac{||x-x'||_2}{p})}{\ell^2}\right)\exp\left(-\frac{||x-x'||_2^2}{2\ell^2}\right) \tag{14}$$

where $\{\ell, p\}$ are parameters specifying the length-scale and periodicity of the kernel.

**Proposition 4** *Let $a_{ij}$ be $\frac{1}{Z_4(a)}k_{LP}(q_i,k_j)$ where $Z_4(a)$ is a normalizing constant. Then $a_{ij}$ has the form:*

$$a_{ij} = \frac{1}{Z_4(a)} \times \exp\left(-\frac{2\sin^2(\pi\frac{\sqrt{||\hat{q}_i||_2^2+||\hat{k}_j||_2^2-2\hat{q}_i\hat{k}_j^T}}{p})}{\ell^2}\right)\exp\left(\frac{q_ik_j^T}{\ell^2}\right) \tag{15}$$

*where $\hat{q}_i = \frac{q_i}{||q_i||_2}$ and $\hat{k}_j = \frac{k_j}{||k_j||_2}$.*

In implementation, this corresponds to

$$Attention_{LocallyPeriodic(p)}(Q,K,V) = softmax(-\frac{2\sin^2(\pi\frac{\sqrt{2-2\hat{Q}\hat{K}^T}}{p})}{\sqrt{d_k}} + \frac{QK^T}{\sqrt{d_k}})V \tag{16}$$

**Rational quadratic kernel ($k_{RQ}$).** The rational quadratic kernel has the form

$$k_{RQ}(x,x') = \left(1 + \frac{||x-x'||_2^2}{2\alpha\ell^2}\right)^{-\alpha} \tag{17}$$

where $\{\ell, \alpha\}$ are parameters specifying the length-scale and the trade-off between smale- and large-scale features of the kernel respectively.

**Proposition 5** *Let $a_{ij}$ be $\frac{1}{Z_5(a)}k_{RQ}(q_i,k_j)$ where $Z_5(a)$ is a normalizing constant. Then $a_{ij}$ has the form:*

$$a_{ij} = \frac{1}{Z_5(a)} \times \left(1 + \frac{1}{\alpha\ell^2} - \frac{2\hat{q}_i\hat{k}_j^T}{2\alpha\ell^2}\right)^{-\alpha} \tag{18}$$

*where $\hat{q}_i = \frac{q_i}{||q_i||_2}$ and $\hat{k}_j = \frac{k_j}{||k_j||_2}$.*

In implementation, this corresponds to

$$Attention_{RationalQuadratic(\alpha)}(Q,K,V) = \frac{\left(1+\frac{1}{\alpha\sqrt{d_k}}-\frac{2QK^T}{2\alpha\sqrt{d_k}}\right)^{-\alpha}}{\sum_k\left(1+\frac{1}{\alpha\sqrt{d_k}}-\frac{2QK^T}{2\alpha\sqrt{d_k}}\right)^{-\alpha}}V \tag{19}$$

# 4 Experiments

## 4.1 English to Chinese Translation

We first test the transformer with different attention kernels on the common application of transformer - the machine translation task. In particular, our translation dataset consists of around 10,000 English-Chinese sentence pairs with varying length and difficulty, from basic greetings to more descriptive and complex sentences. The transformer model we use follows the same setup as in the original paper. We train the model with *label smoothing* ($\epsilon_{1s=0.1}$) and KL divergence loss. We update the model parameters using *warmup learning rate with Adam optimizer* with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$.

In addition to the validation loss, we also evaluate the model based on BLEU score [6], which is commonly used in machine translation task to measure the similarity between model translation output and reference translation.

Table 1: Transformer Performance on English to Chinese Translation Task

| Attention Kernel Type | Validation Loss (↓) | BLEU Score (↑) |
|---|---|---|
| Original RBF kernel | 0.83 | 56.78 |
| **Linear kernel** | **0.68** | **62.45** |
| Periodic Kernel | 1.73 | 50.92 |
| Locally Periodic Kernel | 2.40 | 48.46 |
| Rational Quadratic Kernel | 1.83 | 51.71 |

From the performance in Table 1 we see that the linear kernel achieves the best performance, with the highest BLEU score and the lowest validation loss. However, previous machine translation literature has suggested that higher BLEU score may not indicate better translation quality. After a manual inspection of the translation (see Appendix), we found that while both the original and the linear kernel transformer have comparable translation qualities on longer sentences, linear kernel often outputs correct translation for shorter sentences where the original RBF kernel fails. Note that the BLEU metric often favors short translations, which might explain the superior performance of linear kernel. Overall, we think this is an interesting observation that could benefit from further investigation and a more thorough evaluation on a larger scale translation dataset.

## 4.2 Time series data

While we have shown that our proposed kernels are valid and provide competitive performance on the complex task of text translation, the main contribution of our method lies in the ability to use different kernel attentions for different data types.

To this end, we experiment with time-series prediction using different signal models and observe improvements in transformer performance.

To accommodate time-series prediction, we modify our transformer architecture as follows: We remove the Transformer decoder and replace it with a fully-connected layer that outputs the predicted value. As the input to the encoder is now an array of values rather than an array of words, we also remove the embedding layer and operate on the input values directly.

### 4.2.1 Data types

We run our experiments on the data shown in Figure 1. These signals are intended to demonstrate different time series patterns that can occur and have the forms shown in Table 2.

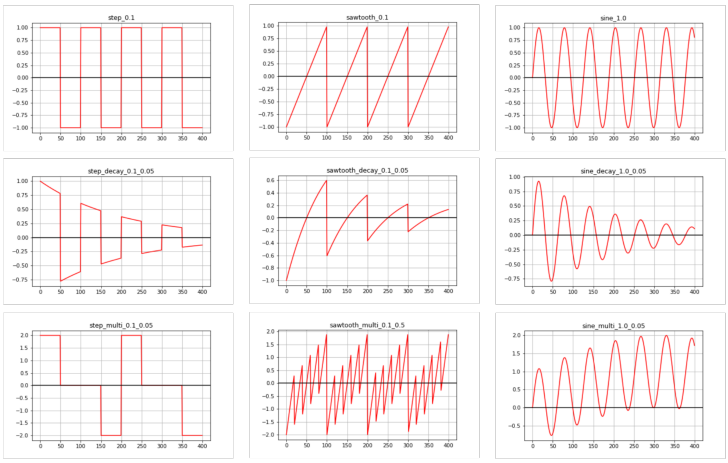


Figure 1: Synthetic data for experimentation.

Table 2: Signal generating equations

| Signal | Equation |
|---|---|
| Step | $\text{square}(\alpha t)$ |
| Step (decaying) | $\text{square}(\alpha_1 t) \times \exp(-\alpha_2 t)$ |
| Step (multiple) | $\text{square}(\alpha_1 t) + \text{square}(\alpha_2 t)$ |
| Sine | $\sin(\alpha t)$ |
| Sine (decaying) | $\sin(\alpha_1 t) \times \exp(-\alpha_2 t)$ |
| Sine (multiple) | $\sin(\alpha_1 t) + \sin(\alpha_2 t)$ |
| Sawtooth | $\text{sawtooth}(\alpha t)$ |
| Sawtooth (decaying) | $\text{sawtooth}(\alpha_1 t) \times \exp(-\alpha_2 t)$ |
| Sawtooth (multiple) | $\text{sawtooth}(\alpha_1 t) + \text{sawtooth}(\alpha_2 t)$ |

#### 4.2.2 Experiment details

For each data type, we run long- and short-term predictions using the five kernel types from before and investigate the final prediction accuracies as well as convergence behaviour for each attention module.

#### 4.2.3 Results

**Long-term prediction results.** We show results for long-term predictions in Figure 2.

We observe that all attention variants perform similarly and are able to recover long-term trends from the data, which further confirms our conclusion that the kernel attention variants show competitive performance.

Some of our further observations are as follows:

- **For the Sine (decaying) signal, $Attention_{Linear}$ prediction is center-biased:** Looking at the $Attention_{Linear}$ prediction for the Sine (decaying) signal, we observe that the predicted signal is centered around $y = -0.025$ and has an amplitude smaller than that of the true prediction. This is likely caused by the non-stationary nature of the linear distance kernel which has pre-specified center and spread parameters.
- **All methods perform relatively poorly on the Sawtooth (multiple) signal:** Due to the high-frequency nature of the Sawtooth (multiple) signal (Figure 1), we observe that while all kernel variants are able to recover the primary trend of the signal (repeating spikes), they are unable to generate a perfect reconstruction and fall short in terms of the signal amplitude. We can note that $Attention$ and $Attention_{Localperiodic}$ perform best.

- $Attention_{Localperiodic}$ **is able to accurately model the Step signal:** While all kernel variants are able to recover the long term trend of the Step signal, they fail to reconstruct the square nature of individual signal blocks (see $Attention_{Linear}$ and $Attention_{Rationalquadratic}$ results). We observe that $Attention_{LocalPeriodic}$ performs best as it is able to preserve the perpendicular angles of the form.

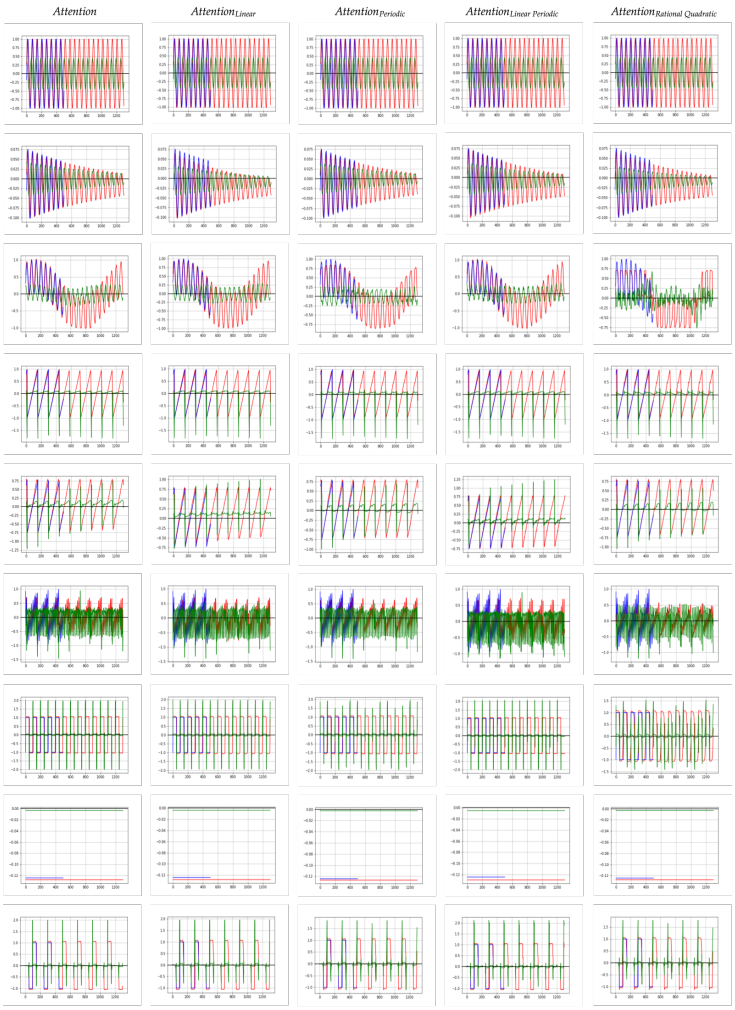


Figure 2: Long-term prediction results. Input signal shown in blue, prediction shown in red, difference shown in green.

**Short-term prediction results.** We show results for short-term predictions in Figure 4.

We observe that the short-term predictions are more indicative of model performance, as extrapolating trends from only a short window of data is harder since the system is required to not only extrapolate the long-term trend of the signal but also the form of the repeating patterns.

Some of our further observations are as follows:

- $Attention$ **performs poorly on the Sawtooth family of signals whereas** $Attention_{Periodic}$ **and** $Attention_{RationalQuadratic}$ **perform well:** Looking at the $Attention$ predictions for the Sawtooth signal family, we observe that the module not only fails to identify the long terms trends (decay, multiple, etc.), but is also unsuccessful in recovering the pattern form, instead outputting noise. On the other hand, we observe that the $Attention_{Periodic}$ and $Attention_{RationalQuadratic}$ kernels are able to repeat the

generating pattern successfully, and even reconstruct reasonable outputs for the more difficult Sawtooth (multiple) signal. Of the two, we can state that $Attention_{Periodic}$ ultimately performs the best, which can be attributed to the periodicity-bias of the periodic kernel.

- $Attention_{Locallyperiodic}$ **is able to preserve the box form of the Step signal more successfully than** $Attention$**:** As in the long-term predictions, we observe that while both of them can generate reasonable continuations, $Attention_{Locallyperiodic}$ outperforms $Attention$ in terms of repeating the original pattern form more closely.
- $Attention_{Periodic}$ **and** $Attention_{Localperiodic}$ **converge faster than** $Attention$ **for Sine (decaying):** While both $Attention$ and $Attention(Localperiodic)$ converge to good approximations to the signal, we observe that $Attention(Periodic)$ converges early on while $Attention$ initially misidentifies the long-term trend.

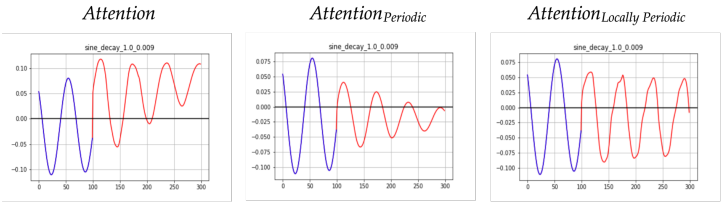


Figure 3: Short-term prediction results (Epoch 5) for Sine (decaying).

## 5 Conclusion

We propose a method which benefits from the observation that Vaswani *et al.*'s attention formulation comprises an implicit kernel distance calculation and report the performance of our model by varying the kernel calculation in the attention module. We first show that our proposed kernel methods perform competitively on the complex task of English-to-Chinese translation task, with the linear kernel surpassing the regular attention baseline. We then show that our method is not only able to match but outperform Vaswani *et al.* in time-series prediction by exploiting well-known kernel/signal compatibilities commonly used in classical kernel literature. While our results are significant on their own, their significance is compounded by their role as an instance of the fingerpost towards making the extensive kernel literature available for experimentation in transformer-based models.

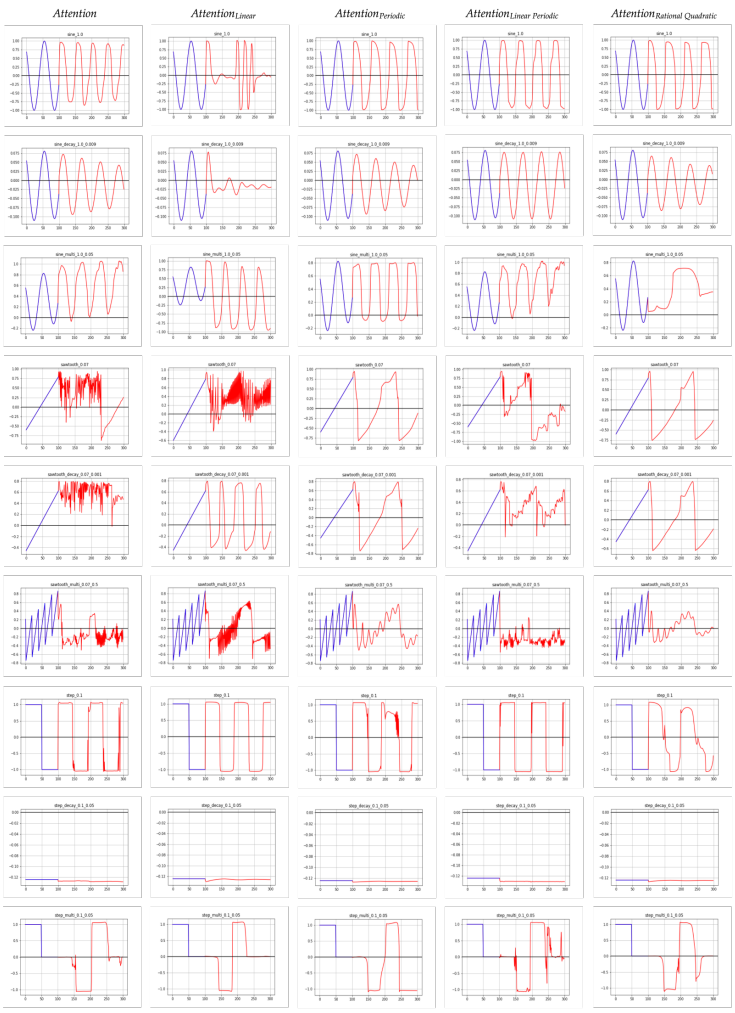


Figure 4: Short-term prediction results. Input signal shown in blue, prediction shown in red.

## References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] 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. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[5] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[7] Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021.

[8] Bernhard Scholkopf. The kernel trick for distances. *Advances in neural information processing systems*, pages 301–307, 2001.

[9] Kyungwoo Song, Yohan Jung, Dongjun Kim, and Il-Chul Moon. Implicit kernel attention, 2021.

[10] Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: A unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc., 2017.

[12] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case, 2020.

[13] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nystr\" omformer: A nystr\" om-based algorithm for approximating self-attention. *arXiv preprint arXiv:2102.03902*, 2021.

## A Qualitative translation outputs

From a qualitative comparison of the translation output from the original model and the transformer with linear kernel (Figure 5), we see that the latter is able to give correct outputs for many sentences where the original model (RBF kernel) fails.

The first line is the English input; the second line is the Chinese reference translation and the third line is the output from transformers.

## B Kernel hyper-parameters

For all five kernels, we use $l = \sqrt[4]{d_k}$ as in the original Vaswani *et al.* paper.

For the periodic and locally periodic kernels, we set $p = 0.01$. For the rational quadratic kernel, we set $\alpha = 99$.

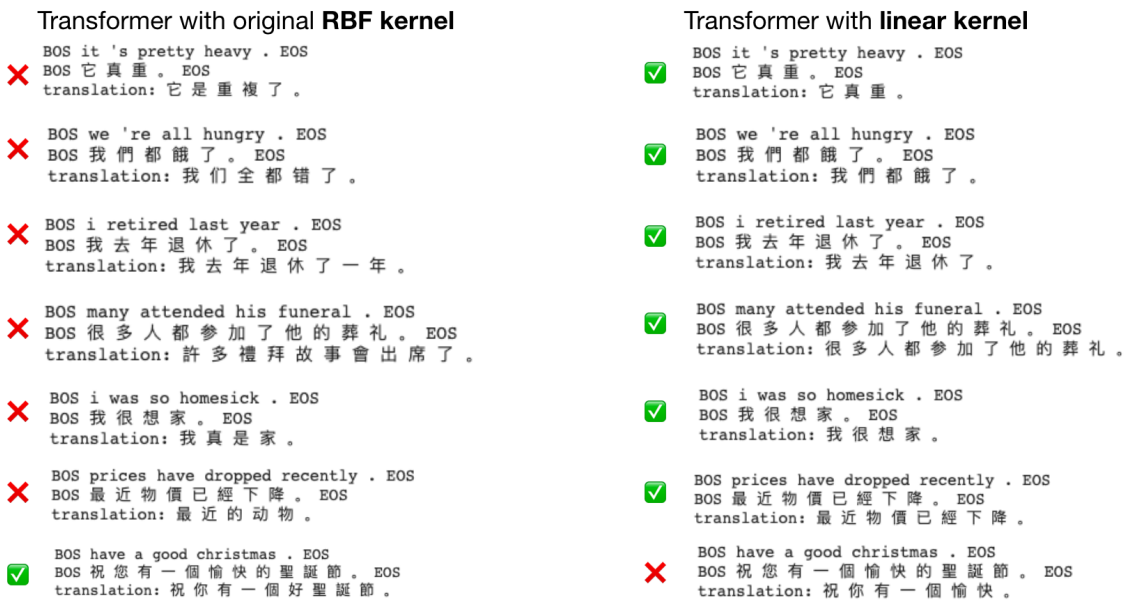


Figure 5: Translation output examples