Supplement to "Improving Transformer with an Admixture of Attention Heads"

A Experiment Details

In this section, we provide model and training details for experiments in Section 3. In our experiments, we consider the number of global attention matrices as a hyper parameter to finetune. We observe that in all of our experiments, choosing the global attention matrices in FiSH to be 1/4 and 1/2 of the number attention heads in the original MHA results in models with good accuracy and efficiency. All of our experiments are conducted on a server with 4 NVIDIA A100 GPUs.

A.1 Language Modeling

Datasets and metrics WikiText-103 consists of articles from Wikipedia and is a dataset with long contextual dependencies. The training set is made up of about $28K$ articles containing $103M$ running words; this corresponds to text blocks of about 3600 words. The validation and test sets are composed of $218K$ and $246K$ running words, respectively. Each of them contains 60 articles and about $268K$ words. Our experiment follows the standard setting [46, 66] and splits the training data into L-word independent long segments. For evaluation, we use a batch size of 1 and go through the text sequence with a sliding window of size L. We consider only the last position for computing perplexity (PPL) except in the first segment, where all positions are evaluated as in [2, 66].

A.2 Machine Translation

Datasets and metrics The dataset IWSLT'14 De-En contains about $170K$ training sentence pairs, $7K$ validation pairs, and $7K$ test pairs. In this task, the model does the translation from German to English. The WMT dataset is a rich-resource English-German machine translation dataset, containing about $4.5M$ training sentence pairs. Validation and test data are from the corresponding newest data. The BLEU score [54] is used to measure the performance of the trained model.

A.3 Image Classification

Datasets and metrics The ImageNet dataset [63] contains about $1.281M$ training images and $50K$ validation images, the model learns to predict which one of 1000 classes an image belongs to. Our Swin Transformer [44] experiments are based on the public code https://github.com/microsoft/Swin-Transformer, we implemented our Hard GFiSH models with the Swin-T version. We add our global heads to the last 8 of the total 12 layers of the model, on each layer we set the number of global heads to half the number of heads, which are 6 and 12 global heads for layers with 12 and 24 heads, respectively. Our experiments were conducted on a server with 1 NVIDIA RTX 3090. We set the batch size to 128 and the learning rate to 1.25e-4, all models are trained with single precision.

A.4 UEA Time Series Classification

Datasets and metrics There are 30 datasets in the benchmark [5]. Following [81], to evaluate our models on temporal sequences, we choose 10 datasets, which vary in input sequence lengths, the number of classes, and dimensionality. We report the test accuracy as an evaluation for the benchmark.
Models and baseline The experiment setups and configurations for the softmax and our models are the same as in [81] [1] (for the PEMS-SF, SelfRegulationSCP2, UWaveGestureLibrary datasets) and [83] [2] (for the other tasks). In all models, the number of heads is 8, whereas the model dimension and number of transformer layers are varied. Our GFiSHformer uses 4 global heads to produce 8 local heads.

A.5 Reinforcement learning on the D4RL benchmark

Datasets and metrics The D4RL benchmark [29] contains the continuous control tasks for offline reinforcement learning. We adapt the selection from [81], including HalfCheetah, Hopper, and Walker as experiment environments and Medium-Expert, Medium, and Medium-Replay as behavior policies.
Models and baseline The softmax baseline trained on this benchmark adopts the configuration from [81], with 3 transformer layers and 4 heads per layer. The 2-global-head GFiSHformer also shares the same configuration and training set-ups.

[1] Implementation available at https://github.com/thuml/Flowformer.
[2] Implementation available at https://github.com/gzerveas/mvts_transformer.

A.6 Applying (G)FiSH on linear transformers

Here we provide the detailed implementation of (G)LFiSHformer, i.e. (G)FiSH + linear transformer, discussed in Section 3.7 and Table 3 in the main text. The linear transformer reduces the quadratic computational cost of self-attention to linear complexity, in terms of the sequence length, by linearizing the softmax kernel [37]. We combine (G)FiSHformer with linear transformer by generating the global K^TV and then sampling the local K^TV from the global ones, resulting in the Transformer with a Linear Finite Admixture of Shared Heads (LFiSHformer). Similar to (G)FiSHformer, we derive four different LFiSH-based transformers: LFiSHformer, Hard LFiSHformer, Generalized LFiSHformer (GLFiSHformer), and Hard Generalized LFiSHformer (Hard GLFiSHformer). Our LFiSH-variants improve the performance of the linear baseline, as demonstrated in Table 3

B Additional Experimental Results

B.1 A Comparison on the Model Efficiency for the IWSLT14 De-En Machine Translation Task

Fig. 4 summarizes the advantage in efficiency of 2-global-head GFiSHformer over the 4-head baseline on the IWSLT' 14 De-En task.

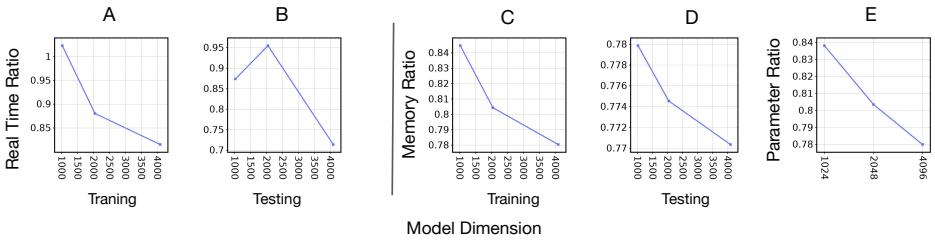

Figure 4: (Left) Training (A) and Inference (B) real time ratios between a 2-global-head GFiSHformers with 4-head MHA baselines across different model dimensions D trained on the IWSLT14 De-En machine translation task. (Right) GPU memory usage at train time (C) and test time (D) and number of parameters (E) ratios between 2-global-head GFiSHformers with 4-head MHA baselines across different model dimensions D. 2-global-head GFiSHformers are significantly more efficient than the baseline as D increase, indicating the benefits of our method for long-range and large-scale tasks. Note that the ratios do not change much when N increase for this task.

B.2 Train and validation PPL of models trained for the WikiText-103 language modeling task

Figure 5 shows the train and valid PPL of 4-global-head FiSH-based models vs . 8-head MHA Transformer trained for the WikiText-103 language modeling task.

B.3 More Results to Show that FiSHformer Helps Reducing Head Redundancy

Table 12 presents the layer-average mean and variance of distances between heads in Hard FiSHformers and Hard GFiSHformers compared with those in the MHA softmax baselines. Models are trained for the WikiText-103 language modeling task.

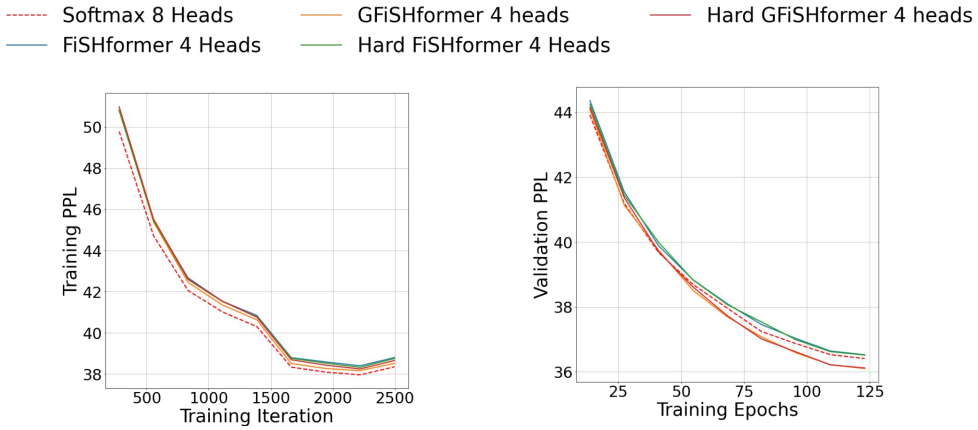

Figure 5: Train and validation PPL of 4-global-head FiSH-based models vs . 8-head MHA Transformer trained for the WikiText-103 language modeling task.

Table 12: Laver-Average mean and variance of $\mathcal{L}_2$ distances between heads of models trained for the WikiText-103 language modeling task.

Method	Mean	Variance
Softmax 8 heads	1.62	0.66
Hard FiSHformer 4 global heads	1.75	1.38
Hard GFiSHformer 2 global heads	2.99	2.79
Hard GFiSHformer 4 global heads	3.58	3.01
Hard GFiSHformer 6 global heads	2.90	1.71

Table 13: Layer-average number of principle components for 95% variance explained of models trained for the WikiText-103 language modeling task.

Method	Mean
Softmax 8 heads	296
Hard FiSHformer 4 global heads	302
Hard GFiSHformer 2 global heads	1161
Hard GFiSHformer 4 global heads	1317
Hard GFiSHformer 6 global heads	1102

B.4 More Results on Eigen Analysis

Table 13 presents the layer-average number of principle components for 95% variance explained of the covariance of attention matrices in Hard FiSHformers and Hard GFiSHformers trained for the WikiText-103 language modeling task compared with those in the MHA softmax baselines. Models are trained for the WikiText-103 language modeling task.

C Efficiency Analysis

In this section, we present the efficiency improvement of (G)FiSHformers over the MHA baseline on the WikiText-103 language modeling task. We show that the advantage in the efficiency of (G)FiSHformers increases significantly as the model dimension D and sequence length N grows, making (G)FiSHformers more suitable and superior for large-scale applications. In our analysis, we report the number of FLOPs, the number of model parameters, and memory usage (bytes) as the measures of model efficiency.

Analysis setting We investigate the benefits of our model computation and memory reduction through different $D \in \{256, 512, 1024, 2048, 4096\}$ and $N \in \{128, 256, 512, 1024, 2048, 4096, 8192\}$. For the FLOP calculation, we use fvcore. Another notice is that, for model complexity, we distinguish between input-embedding parameters and non-embedding parameters. Input-embedding parameters are used to represent the inputs before sending them to the model. Non-embedding parameters are the parameters of the main model. Since our method aims at reducing the size of the transformer model, it is important to compare the reduction in non-embedding parameters. Hence, we report both model's total parameters and non-embedding parameters in this analysis. All measurements are calculated when running the model through data of batch size 1.

FiSH-based models significantly improves computational cost

GFiSHformer benefits computation in large-scale tasks. By showing the FLOP ratios between 2-global-head GFiSHformers with the 8-head MHA baseline across different model dimensions and sequence lengths, Figure 3 indicates that our model requires significantly less computation than the baseline. Especially for both training and inference, this substantially grows with the increases of D and N (up to more than 30 % reduction), which makes our method more preferable for long-range and large-scale tasks.

FiSH-based variants share comparable advantages in computation saving. For further analysis, we compare the FLOP of FiSHformers, Hard FiSHformers, GFiSHformers, and Hard GFiSHformer. Figure 6 shows that given a D, when N is increased, FiSH-based models have comparable training and inference computation (FLOP). Since we have shown that GFiSHformer benefits computation, the FLOP comparison among FiSH-based models further confirms that our methods have a substantial advantage in computational saving.

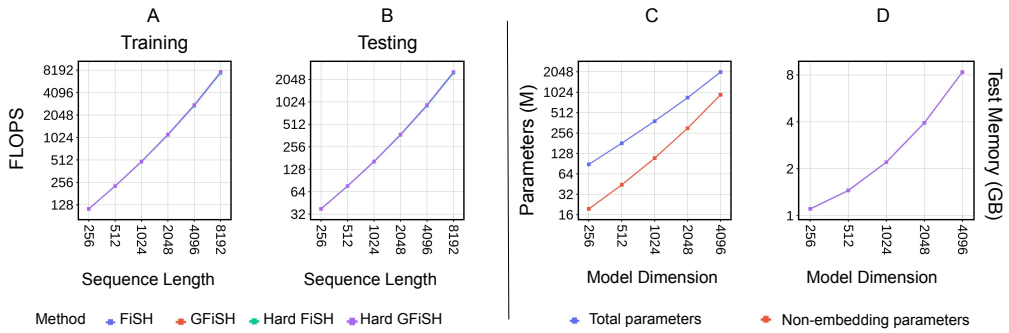

Figure 6: Training (Left) and Inference (Right) FLOPS (B) of 2-global-head FiSHformers, Hard FiSHformers, GFiSHformers, and Hard GFiSHformer. FiSH-based models have comparable computational costs trained on the WikiText-103 language modeling task. Here the model size is 1024.

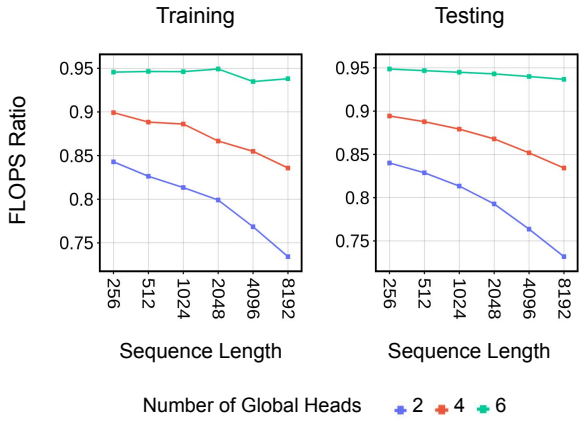

Figure 7: Training (Left) and Inference (Right) FLOP ratio between the GFiSHformers and the 8-head MHA baseline trained on the WikiText-103 language modeling task, as the number of heads vary in {2, 4, 6}. A decrease in the number of heads leads to a significant reduction in computational cost at each sequence length. Here, the model dimension is 1024.

GFiSHformer computation efficiency rapidly increases as the number of global heads decreases We compare the computational-cost reduction of GFiSHformer as the global heads vary. Figure 7 shows the FLOP ratio of GFiSHformer with 2/4/6-global-head versus the 8-head baseline for a given D and various sequence lengths. As the number of heads decreases, GFiSHformers achieve significant computation reduction (in both training and inference) (Figure 7).

(G)FiSHformers improves memory usage and model complexity In addition to the computational saving, our method achieves significant benefits in memory cost (at test time) and model complexity (total/non-embedding parameters) over the baseline. Following the computation analysis, we first present the advantage of the GFISHformers compared to the 8-head MHA Transformer. Figure 8 shows the number-of-parameter-ratio (Left) and the memory-ratio (Right) of the GFiSHformers with 2/4/6 global heads and the 8-head baseline. At a fixed sequence length $N = 1024$, as we scale up the model dimensions, our method becomes significantly more beneficial than MHA Transformer, indicating the advantage of GFiSHformer in large-scale applications.

Secondly, we show that FiSH-based models are comparable in model size and memory saving, which further indicates that all variants of FiSHformers benefit space complexity. Figure 6 shows that 2-head FiSHformers, Hard FiSHformers, GFiSHformers, and Hard GFiSHformer have a comparative number of total/non-embedding parameters and memory usage at different model dimensions, for a given sequence length.

Finally, we examine the model space complexity of a Fishformer variant as the number of global heads vary. Figure 8 shows a significant increase in total (Left) /non-embedding parameters (Middle) and memory usage (Right) reduction of 2/6/8-global heads GFiSHformers, as we increase D. While only having the fewest heads, 2-head GFiSHformer is the most efficient model (lowest metric ratio

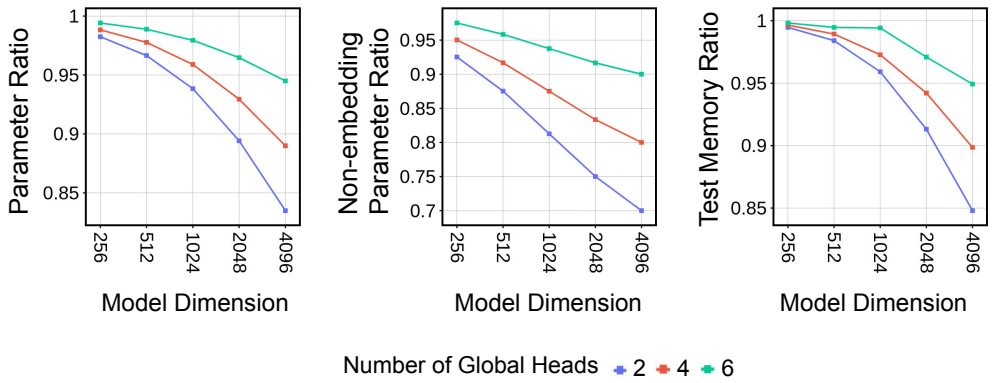

Figure 8: Number-of-parameter ratio (Left) and the memory ratio (at test time) (Right) between the GFiSH-formers and the 8-head MHA Transformer trained on the WikiText-103 language modeling task. Our method achieve significant reduction in memory cost and model complexity over the baseline as we scale up the model dimension, $D \in \{256, 512, 1024, 2048 \text{ to } 4096\}$. Here, the sequence length is 1024, and the number of global heads is chosen to be 2, 4 and 6.

with the 8-head MHA baseline) and achieves comparable performance with the baseline, for model dimension, and sequence length are 128 and 256 respectively, as indicated in Table 3.1

D An Analysis on the Computational Complexity and the Number of Parameters in FiSH and the Softmax Attention

In this section, we compare the computational complexity and the number of parameters in the FiSH with M global attention heads and H local attention heads to the H-head baseline MHA softmax transformer. Following the same notation in Section 1.1 in the main text, we let D_x, N, and D be the input dimension, the input length, and the model/feature dimension, respectively. To simplify the computation, we also do not take the softmax operator into account.

D.1 Computational Complexity

(i) Softmax H-head attention: The number of computations needed to compute attention matrices in a softmax H-head attention is $N^2H(2D-1) + 2NHD(2D_x - 1)$.

Explanation: To calculate the query matrix $\mathbf{Q}$ and the key matrix $\mathbf{K}$ in Step 1 in Section 1.1 at each head, we need $2NDD_x$ multiplications and $2ND(D_x - 1)$ additions. In total, these need $2ND(2D_x - 1)$ computations. Next, to compute the product $\mathbf{QK}^\top$ in Eqn. (1), we need N^2D multiplications and $N^2(D-1)$ additions. In total, computing an attention matrix in Eqn. (1) at each head requires $2ND(2D_x - 1) + N^2D + N^2(D-1) = N^2(2D-1) + 2ND(2D_x - 1)$ computations. The total computation needed to compute attention matrices at H heads is then $N^2H(2D-1) + 2NHD(2D_x - 1)$.

(ii) FiSH with M global attention heads and H local attention heads: The number of computations needed to compute attention matrices in a FiSH with M global attention heads and H local attention heads is $[2(D+H)M - H]N^2 + 2NMD(2D_x - 1)$.

Explanation: Similar to the above derivation, M global attention matrices need $N^2M(2D-1) + 2NMD(2D_x - 1)$ computations. The H local attention matrices need $H(MN^2 + (M-1)N^2) = H(2M-1)N^2$. There are also MN^2 computations needed to add noise to M global attention matrices. Thus, the number of computations needed to compute attention matrices in a FiSH with M global attention heads and H local attention heads is $N^2M(2D-1) + 2NMD(2D_x - 1) + H(2M-1)N^2 + MN^2 = [2(D+H)M - H]N^2 + 2NMD(2D_x - 1)$.

Soft-max H-head attention versus FiSH with M global attention heads and H local attention heads: Given the results in (i) and (ii), when compared to the baseline softmax H-head attention, our FiSH with M global attention heads and H local attention heads saves

$$[2(H-M)D - 2MH)]N^2 + 2(H-M)D(2D_x - 1)N$$

computations in a forward pass. When N is large, this difference is significant.

D.2 The Number of Parameters

(iii) Softmax H-head attention: The number of parameters needed to compute the attention matrices in a softmax H-head attention is $2HDD_x$.

Explanation: $2HDD_x$ parameters is from the linear projects to calculate the query matrix $\mathbf{Q}$ and the key matrix $\mathbf{K}$ in Step 1 in Section 1.1.

(iv) FiSH with M global attention heads and H local attention heads: The number of parameters in a FiSH with M global attention heads and H local attention heads is $2MDD_x + HM + M$.

Explanation: $2MDD_x$ parameters is from the linear projects to calculate M query matrices $\mathbf{Q}$ and M key matrices $\mathbf{K}$, which are used to compute M global attention matrices. The extra HM parameters is from the linear mapping for computing H local attention matrices from M global attention matrices, and the extra M parameters are the M $\{\sigma_k\}_{k=1}^M$ for M global heads.

Softmax H-head attention versus FiSH with M global attention heads and H local attention heads: Given the results in (iii) and (iv), when compared to the baseline softmax H-head attention, our FiSH with M global attention heads and H local attention heads saves $2(H-M)DD_x - HM - M$ parameters. When D is large, this saving is significant.

E Proofs

In this appendix, we provide proof for Lemma 1.

E.1 Proof of Lemma 1

We denote

$$p_G(x) := \int f(x-\theta)dG(\theta) = \int \phi(x|\theta, \sigma^2\mathbf{I})dG(\theta),$$

for all $x \in \mathbb{R}^d$ where $f(x) = \frac{1}{(\sqrt{2\pi}\sigma)^d}\exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)$ for given $\sigma > 0$. From the work of [4], the space of Gaussian mixtures is dense in the space of continuous probability measures. Therefore, we can find probability distribution G_1 such that

$$\sup_{x\in\mathbb{R}^d} |p(x) - p_{G_1}(x)| \leq \frac{\epsilon}{2}. \tag{11}$$

To obtain the conclusion of the lemma, it is sufficient to prove that we can find a probability measure G_2 with at most K supports where $K \leq (C\log(1/\epsilon))^d$ for some universal constant C such that

$$\sup_{x\in\mathbb{R}^d} |p_{G_1}(x) - p_{G_2}(x)| \leq \frac{\epsilon}{2}. \tag{12}$$

Our technique for proving the above approximation bound relies on Lemma A.1 in [30]. In particular, that lemma entails that for any $k \geq 1$ there exists a probability distribution G_2 with at most $(2k-2)^d$ supports such that

$$\int \theta^\alpha d(G_1 - G_2)(\theta) = 0, \tag{13}$$

for any $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d) \in \mathbb{N}^d$ such that $0 \leq |\alpha| = \sum_{j=1}^d \alpha_j \leq 2k-2$, Here, $\theta^\alpha = \prod_{j=1}^d \theta_j^{\alpha_j}$.

Now, for any $M \geq 2a\sqrt{d}$, we have $\|x-\theta\| \geq \|x\| - \|\theta\| > M - a\sqrt{d} > M/2$ as long as $\|x\| > M$ and $\theta \in [-a, a]^d$. It indicates that

$$\begin{aligned}
\sup_{\|x\|>M} |p_{G_1}(x) - p_{G_2}(x)| &= \sup_{\|x\|>M} \left|\int f(x-\theta)d(G_1 - G_2)(\theta)\right| \\
&\leq \sup_{\|x\|>M} \int \frac{1}{(\sqrt{2\pi}\sigma)^d}\exp\left(-\frac{\|x-\theta\|^2}{2\sigma^2}\right) d(G_1 + G_2)(\theta) \\
&\leq \frac{2}{(\sqrt{2\pi}\sigma)^d}\exp\left(-\frac{M^2}{8\sigma^2}\right).
\end{aligned} \tag{14}$$

On the other hand, for any $k \geq 1$ we also have that

$$\begin{aligned}
\sup_{\|x\|\leq M} |p_{G_1}(x) - p_{G_2}(x)| &= \sup_{\|x\|\leq M} \left|\int f(x-\theta)d(G_1 - G_2)(\theta)\right| \\
&\leq \sup_{\|x\|\leq M} \left|\int \left(f(x-\theta) - \sum_{j=0}^{k-1} \frac{(-1)^j\|x-\theta\|^{2j}}{(\sqrt{2\pi})^d\sigma^{d+2j}j!}\right) d(G_1 - G_2)(\theta)\right|,
\end{aligned} \tag{15}$$

where the final inequality stems from

$$\int \sum_{j=0}^{k-1} \frac{(-1)^j \|x-\theta\|^{2j}}{(\sqrt{2\pi})^d \sigma^{d+2j} j!} d(G_1 - G_2)(\theta) = 0,$$

which is due to Eqn. (13).

To further bound the right-hand-side (RHS) of Eqn. (15), we use the following inequality:

$$\left| \exp(y) - \sum_{j=0}^{k-1} (y)^j / j! \right| \leq |y|^k / k!$$

for any $y \in \mathbb{R}$. Since $k! \geq (k/e)^k$ for any $k \geq 1$, the above bound can be rewritten as

$$\left| \exp(y) - \sum_{j=0}^{k-1} (y)^j / j! \right| \leq \frac{|ye|^k}{k^k}. \tag{16}$$

Further simplification of Eqn. (15) leads to

$$\begin{aligned}
\sup_{\|x\| \leq M} |p_{G_1}(x) - p_{G_2}(x)| &\leq \sup_{\|x\| \leq M} \int \left| f(x-\theta) - \sum_{j=0}^{k-1} \frac{(-1)^j \|x-\theta\|^{2j}}{(\sqrt{2\pi})^d \sigma^{d+2j} j!} \right| d(G_1 + G_2)(\theta) \\
&\leq 2 \sup_{\|x\| \leq M, \theta \in [-a,a]^d} \left| f(x-\theta) - \sum_{j=0}^{k-1} \frac{(-1)^j \|x-\theta\|^{2j}}{(\sqrt{2\pi})^d \sigma^{d+2j} j!} \right| \\
&\leq \sup_{\|x\| \leq M, \theta \in [-a,a]^d} \frac{e^k \|x-\theta\|^{2k}}{\sigma^{2k} (2k)^k},
\end{aligned}$$

where the final inequality is based on an application of inequality (16) with $y = -\|x-\theta\|^2/(2\sigma^2)$. For $\|x\| \leq M$ and $\theta \in [-a,a]^d$, we have $\|x-\theta\| \leq \|x\| + \|\theta\| \leq M + a\sqrt{d}$. Therefore, we further have

$$\sup_{\|x\| \leq M} |p_{G_1}(x) - p_{G_2}(x)| \leq \sup_{\|x\| \leq M, \theta \in [-a,a]^d} \frac{e^k \|x-\theta\|^{2k}}{\sigma^{2k}(2k)^k} \leq \frac{e^k (M + a\sqrt{d})^{2k}}{\sigma^{2k}(2k)^k}.$$

When $M \geq 2a\sqrt{d}$, we have $M + a\sqrt{d} \leq \frac{3M}{2}$ and the above bound leads to

$$\sup_{\|x\| \leq M} |p_{G_1}(x) - p_{G_2}(x)| \leq \frac{(9e)^k M^{2k}}{(8\sigma^2 k)^k}. \tag{17}$$

By choosing $M^2 = 8\sigma^2 \log(1/\epsilon')$ for some $\epsilon' > 0$, the bounds in Eqns. (14) and (17) become

$$\begin{aligned}
\sup_{\|x\| \leq M} |p_{G_1}(x) - p_{G_2}(x)| &\leq \frac{2}{(\sqrt{2\pi}\sigma)^d} \epsilon', \\
\sup_{\|x\| > M} |p_{G_1}(x) - p_{G_2}(x)| &\leq \frac{(9e)^k (\log(1/\epsilon'))^k}{k^k}.
\end{aligned} \tag{18}$$

As long as we choose $k = 9e^2 \log(1/\epsilon')$ and $\epsilon' \leq 1$, we have

$$\sup_{\|x\| > M} |p_{G_1}(x) - p_{G_2}(x)| \leq e^{-k} = e^{-9e^2 \log(1/\epsilon')} = (\epsilon')^{9e^2} \leq \epsilon'. \tag{19}$$

By choosing $\epsilon' = \frac{\epsilon}{2 \max\{\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1\}}$, the results from Eqns. (18) and (19) indicate that

$$\sup_{\|x\| \leq M} |p_{G_1}(x) - p_{G_2}(x)| \leq \frac{\epsilon}{2}, \quad \text{and} \quad \sup_{\|x\| > M} |p_{G_1}(x) - p_{G_2}(x)| \leq \frac{\epsilon}{2}.$$

Therefore, if we choose $M = 8\sigma^2 \log\left(\frac{2 \max\{\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1\}}{\epsilon} \right)$ and $k = 9e^2 \log\left(\frac{2 \max\{\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1\}}{\epsilon} \right)$, we have

$$\sup_{x \in \mathbb{R}^d} |p_{G_1}(x) - p_{G_2}(x)| \leq \frac{\epsilon}{2}.$$

It indicates that we obtain the conclusion of claim (12) by choosing $K = (2k-2)^d \leq \left(18e^2 \log\left(\frac{2 \max\{\frac{2}{(\sqrt{2\pi}\sigma)^d}, 1\}}{\epsilon} \right) \right)^d$. As a consequence, we obtain the conclusion of the lemma.