Making Vision Transformers Efficient from A Token Sparsification View Supplementary Materials

Shuning Chang[1*] Pichao Wang[2†‡] Ming Lin[2‡] Fan Wang[2] David Junhao Zhang[1]
Rong Jin[2] Mike Zheng Shou[1†]
[1]Show Lab, National University of Singapore [2]Alibaba Group

A.1. Detailed Architectures

The detailed architectures of our STViT-DeiT and STViT-Swin are shown in Table 1 and Table 2, where an input image size 224×224 is assumed for all the networks and the default numbers of semantic tokens are 16 and 36 separately. "Tr(ST)" denotes the transformers processing semantic tokens.

A.2. Computational complexity analysis

Global vision transformer (DeiT). We define that the number of image tokens is N, the number of semantic tokens is M, and their dimension is C. The patch embedding layer is neglected. The computational complexity of a global transformer processing image tokens (IT) is:

$$\begin{aligned} \Omega(MHA(IT)) &= 4NC^2 + 2N^2C, \\ \Omega(FFN(IT)) &= 8NC^2. \end{aligned} \tag{1}$$

The computational complexity of a global transformer processing semantic tokens (ST) is:

$$\begin{aligned} \Omega(MHA(ST)) &= 4MC^2 + 2M^2C, \\ \Omega(FFN(ST)) &= 8MC^2. \end{aligned} \tag{2}$$

The relationships between computational complexity and token number in attention and FFN are quadratic and linear, respectively. Due to the $N \ll M$, our method significantly reduces the cost of transformers, especially the attention. The computational complexity of STGM is:

$$\Omega(STGM) = 2MC^2 + 2NC^2 + 2MNC. \tag{3}$$

In global vision transformers, our STGM is also an efficient module. The computational complexity of whole DeiT and our STViT-DeiT are:

$$\begin{aligned} \Omega(DeiT) &= 144NC^2 + 24N^2C, \\ \Omega(STViT) &= 52NC^2 + 12M^2C + 76MC^2 \\ &\quad +8N^2C + 4MNC. \end{aligned} \tag{4}$$

Local vision transformer (Swin). We define that the number of image tokens (IT) is N, the number of image tokens in each window is W, the number of semantic tokens (ST) in each window is M, and their dimension is C. We only compute the computational complexity in each transformer. The computational complexity of a local transformer processing image tokens (IT) is:

$$\begin{aligned} \Omega(MHA(IT)) &= 4NC^2 + 2W^2NC, \\ \Omega(FFN(IT)) &= 8NC^2. \end{aligned} \tag{5}$$

The computational complexity of a local transformer processing image tokens (ST) is:

$$\begin{aligned} \Omega(MHA(ST)) &= 4(N/W)MC^2 + 2M^2(N/W)C, \\ \Omega(FFN(ST)) &= 8MC^2. \end{aligned} \tag{6}$$

Swin makes the computational complexity linear to token number, while our method further reduces the computational complexity. The computational complexity of STGM is:

$$\Omega(STGM) = 2(N/W)MC^2 + 2NC^2 + 2(N/W)M^2C. \tag{7}$$

A.3. The results on LV-ViT

Setting. In LV-ViT [18], by default, the STGM employs the 6^{th} and 7^{th} transformer layers of LV-ViT-S (with 16 layers in total). We downsample the token labels to match the size of our semantic tokens.

*Work done during an internship at Alibaba Group.
†Equal corresponding authors.
‡Work done at Alibaba Group, and now affiliated with Amazon.

	Output size	STViT-DeiT-T	STViT-DeiT-S	STViT-DeiT-B
Base	14×14	Patch Embedding $\left[dim\ 192, head\ 3\right] \times 4$	Patch Embedding $\left[dim\ 384, head\ 6\right] \times 4$	Patch Embedding $\left[dim\ 768, head\ 12\right] \times 4$
STGM	4×4	$\left[dim\ 192, head\ 3\right] \times 2$	$\left[dim\ 384, head\ 6\right] \times 2$	$\left[dim\ 768, head\ 12\right] \times 2$
Tr(ST)	4×4	$\left[dim\ 192, head\ 3\right] \times 6$	$\left[dim\ 384, head\ 6\right] \times 6$	$\left[dim\ 768, head\ 12\right] \times 6$

Table 1. Detailed architecture of STViT-DeiT.

		Output size	STViT-Swin-T	STViT-Swin-S	STViT-Swin-B
Base	Stage 1	56×56	Patch Embedding $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 96, head\ 3\end{matrix}\right] \times 2$	Patch Embedding $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 96, head\ 3\end{matrix}\right] \times 2$	Patch Embedding $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 192, head\ 6\end{matrix}\right] \times 2$
	Stage 2	28×28	Patch Merging $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 192, head\ 6\end{matrix}\right] \times 2$	Patch Merging $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 192, head\ 6\end{matrix}\right] \times 2$	Patch Merging $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 256, head\ 8\end{matrix}\right] \times 2$
	Stage 3	14×14	Patch Merging $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 384, head\ 12\end{matrix}\right] \times 2$	Patch Merging $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 384, head\ 12\end{matrix}\right] \times 10$	Patch Merging $\left[\begin{matrix}win.\ sz.\ 7\times 7,\\ dim\ 512, head\ 16\end{matrix}\right] \times 10$
STGM	Stage 3	6×6	$\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 384, head\ 12\end{matrix}\right] \times 2$	$\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 384, head\ 12\end{matrix}\right] \times 2$	$\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 512, head\ 16\end{matrix}\right] \times 2$
Tr(ST)	Stage 3	6×6	$\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 384, head\ 12\end{matrix}\right] \times 2$	$\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 384, head\ 12\end{matrix}\right] \times 6$	$\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 512, head\ 16\end{matrix}\right] \times 6$
	Stage 4	6×6	Linear Layer $\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 768, head\ 24\end{matrix}\right] \times 2$	Linear Layer $\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 768, head\ 24\end{matrix}\right] \times 2$	Linear Layer $\left[\begin{matrix}win.\ sz.\ 3\times 3,\\ dim\ 1025, head\ 32\end{matrix}\right] \times 2$

Table 2. Detailed architecture of STViT-Swin.

Model	Metrics	Base	No. of semantic tokens 36	49	100
STViT-LV-ViT-S	Top-1 Acc(%)	83.3	82.7(-0.6)	82.8(-0.5)	83.1(-0.2%)
	FLOPs(G)	6.6	3.69(-44%)	3.91(-41%)	4.62(-30%)
	Throughput(img/s)	1159	2073(+78%)	1933(+72%)	1592(+37%)

Table 3. Results of STViT on LV-ViT-S.

Results. The main results are shown in Table 3. Token labelling in LV-ViT is not friendly for our method. Token labelling emphasizes the importance of all the output tokens and advocates that each output token should be associated with an individual location-specific label [18], while our semantic tokens generated by clustering emphasize high-level semantic information. However, we still achieve good performance. In Table 4, we compare our STViT with the state-of-the-art token sparsification method EViT [22] on LV-ViT-S. Results indicate that our method outperforms it.

A.4. Applications in semantic segmentation

Settings. ADE20K is a widely-used semantic segmentation dataset, including a broad range of 150 semantic classes. It has 25K images in total, with 20K for training, 2K for validation, and 3K for testing. UperNet in mmseg is utilized as our base framework. The w_s is set to 3. Models are trained for 240K iterations. All the other settings follow the Swin Transformer [24].

Comparison to Swin Transformers. Table 5 presents the results of STViT-R-Swin on semantic segmentation. With similar FLOPs reduction, the drop on mIoU is larger com-

pared with those in object detection tasks, which shows that our method still has a gap on dense prediction compared to the full-token network.

We analyze the relatively poor performance from two views. First, the STGM strictly prunes more than 80% tokens by attention, which remains the high-level semantic information but loses nearly all the detailed information. Semantic segmentation is a dense pixel-level classification task, and the semantic tokens are difficult to enhance the pixel-level representation. Second, our spatial pooling layer with large kernel size in STGM and self-attention layers can be regarded as low-frequency filters. STGM filters most high-frequency information, which is necessary for semantic segmentation.

A.5. Additional visualization

We visualize the attention map of the second attention layer in STGM in Figure 1a. The shape of attention map is $N_s \times (N_s + N_i)$, where $N_s = 16$, and $N_i = 196$. The results of the attention computation between semantic tokens S^1 (queries) and semantic tokens S^1 (keys) are shown in the most left 16 columns, and the rest columns show the computation between semantic tokens S^1 and image tokens X. The figure shows that the second semantic token highlights the region of semantic tokens, while other semantic tokens highlight the image tokens. Figure 1c visualizes the attention maps in the self-attention layers after STGM. The second semantic token is incorporated by the majority of semantic tokens. These phenomenons illustrate that the second semantic token focuses on more global semantic information, which further verifies our global cluster center initialization can guide the semantic tokens to extract global semantic information. The phenomenons in Figure 1a and Figure 1c nearly emerge in all the images.

Neglecting the most left 16 columns of Figure 1a, we reshape it into 16 14×14 attention maps like Figure 3 and show them in Figure 1b. Thanks to the clustering of second attention and global initialization G, we can see that the semantic information is more accurate and meaningful.

We visualize the attention maps of semantic tokens with single global initialization in Figure 1d. Without spatial initialization, the response regions are more global and similar. In contrast, our semantics of each semantic token are associated with the specific spatial location, which is the basis to allow our method to be applied in local self-attention and downstream tasks. Additionally, our attention maps contain more recognized and diverse semantic information, reflecting the effectiveness of our spatial initialization.

Method	Top-1 Acc	FLOPs(G)
EViT [22]	82.5(-0.8)	3.9(-41%)
EViT [22]	83.0(-0.3)	4.7(-29%)
STViT(Ours)	82.7(-0.6)	3.7(-44%)
STViT(Ours)	83.1(-0.2)	4.6(-30%)

Table 4. Comparisons with the state-of-the-art token sparsification method EViT on LV-ViT-S.

A.6. Additional ablation study

All the following experiments of STViT and STViT-R are conducted on DeiT-S and Swin-S unless otherwise specified, respectively.

The position of STGM. The effects of different position of STGM are shown in Table 6. Two transformer layers are employed in STGM in all the experiments. We can see that the performance achieves improvement with appropriately moving the STGM towards deep layers due to better features of image tokens.

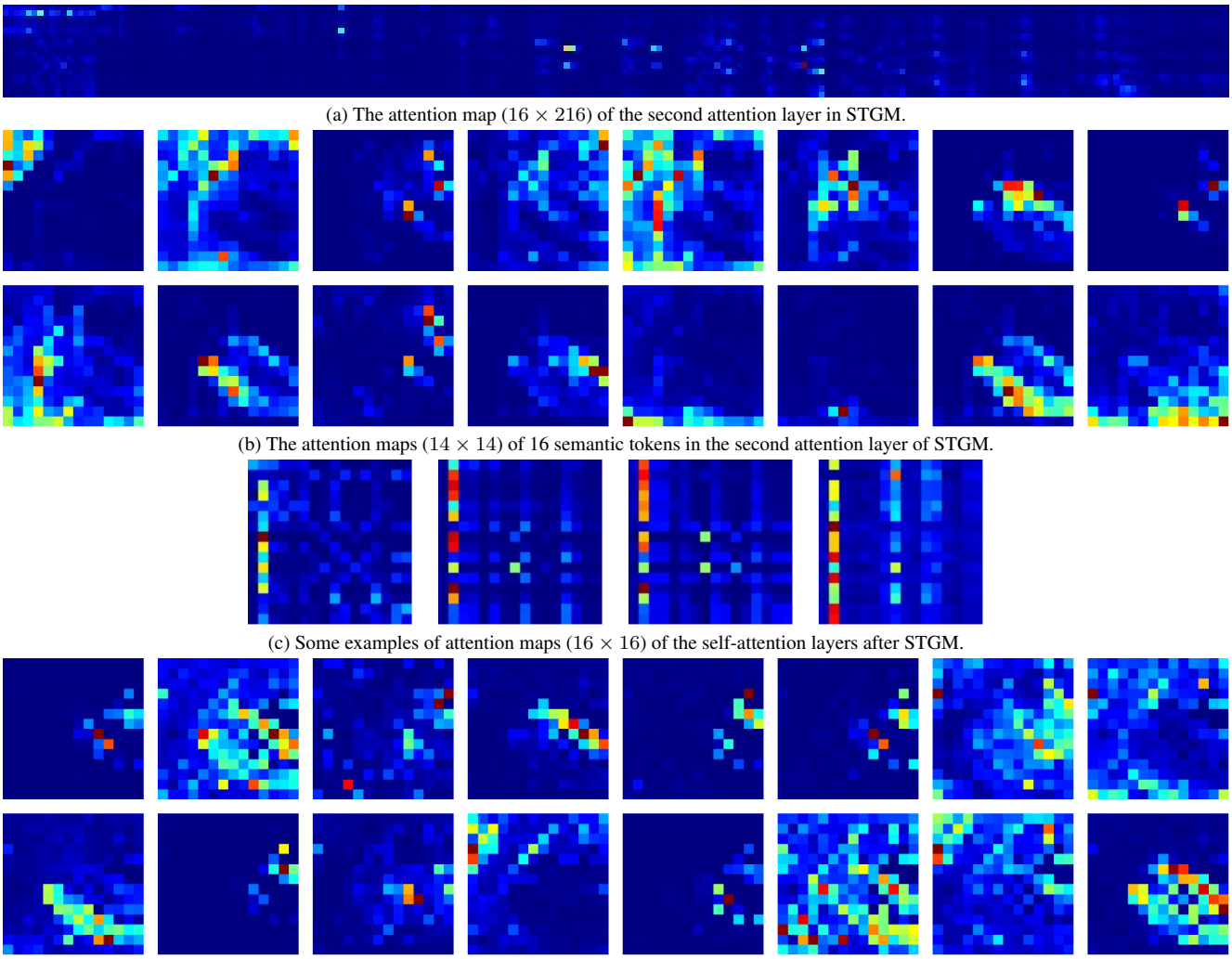
Positional encoding. We try to apply positional encoding to our semantic tokens. Table 7 shows comparisons of different positional encoding methods, including learned positional encoding, conditional positional encoding, and relative positional encoding [24]. All the positional encoding methods do not work on DeiT-S and Swin-T, even though relative positional encoding improves Swin-T by 1.2%. These experiments demonstrate that the interaction between our semantic tokens depends on high-level semantic information and nearly does not use position relationships.

Method	Backbone	mIoU	FLOPs(G)
UperNet	Swin-S	49.3	49
UperNet	STViT-R-Swin-S	48.3	34(-31%)
UperNet	Swin-B	49.7	87
UperNet	STViT-R-Swin-B	48.9	60(-31%)

Table 5. Results of semantic segmentation on the ADE20K val set. A multi-scale inference with resolution $[0.5, 0.75, 1.0, 1.25, 1.5, 1.75]\times$ is applied. FLOPs and latency are measured in backbones with resolution 512×512.

Pos.	3-5	4-6	5-7	7-9	8-10	10-12
Top-1 Acc(%)	79.3	79.8	79.8	80.3	80.3	79.8
FLOPs(G)	1.56	1.91	2.25	2.95	3.30	4.00

Table 6. Performance evaluation on the different positions of our STGM.



(a) The attention map (16×216) of the second attention layer in STGM.

(b) The attention maps (14×14) of 16 semantic tokens in the second attention layer of STGM.

(c) Some examples of attention maps (16×16) of the self-attention layers after STGM.

(d) The attention maps (14×14) of 16 semantic tokens generated by single global initialization.

Figure 1. Additional visualization of attention maps.

	STViT-DeiT-S Acc	STViT-Swin-T Acc
Learned	79.6	81.5
Conditional	79.7	81.4
Relative	79.8	81.3
No pos.	79.8	81.5

Table 7. Performance evaluation on different positional encoding methods. *Learned*, *Conditional*, and *Relative* indicate learned positional encoding, conditional positional encoding, and relative positional encoding, respectively.

Alternative schemes of spatial pooling. We use an intra and inter-window spatial pooling in STGM to generate initial cluster centers, which adaptively save meaningful semantic information and avoids overlap between adjacent windows as much as possible. Furthermore, we explore more spatial pooling schemes, including: (i) spatial pooling with large-size kernel and overlap, (ii) multi-scale spatial pooling, and (iii) adaptive spatial pooling. We adopt 25 semantic tokens in these experiments. In (i), the kernel size and overlap are set to 6 and 4, respectively. In (ii), we use two adaptive pooling layers which produce 9 and 16 tokens separately. The results are presented in Table 8 on DeiT-T. We can see that overlap and multiple scales cannot boost the performance, which also demonstrates that discrete semantic tokens with high-level semantic information benefit our method.

	Scheme i	Scheme ii	Adaptive spatial pooling	Ours
Top-1 Acc(%)	71.6	71.7	71.9	72.2

Table 8. Alternative schemes of spatial pooling.

A.7. Cluster center recovery by self attention

We present an analysis showing how cluster centers are recovered through the attention mechanism. Let K be the number of clusters. Let $\mathcal{N}(\mu_i, \sigma^2 I/d), i = 1, \ldots, K$ be the K Gaussian distributions, with center $\mu_i \in \mathbb{R}^d$ and covariance matrix $\sigma^2 I/d$. Let $x_{i,j} \in \mathbb{R}^d, j = 1, \ldots, n$, be the n data points independently sampled from $\mathcal{N}(\mu_i, \sigma^2 I/d)$. Given data points $\mathcal{D} = \{x_{i,j}, i \in [K], j \in [n]\}$, of course without knowing the association of each data point to its underlying Gaussian distribution, our goal is to recover the underlying cluster centers $\mu_i, i \in [K]$. We assume that all the center vectors of Gaussian distributions are well separated, i.e. $\langle \mu_j, \mu_k \rangle \leq \gamma$ if $j \neq k$. For the convenience of study, we assume $|\mu_i| = 1, i \in [K]$.

Let $\widehat{\mu}_i \in \mathbb{R}^d, i \in [K]$ the initialized cluster centers, with all the cluster centers being well normalized. Define Δ as the gap for any initialized $\widehat{\mu}_i$ to the target cluster centers μ_i than to other clusters μ_j, i.e.

$$\Delta = \min_{i \in [K]} \min_{j \neq i} \langle \widehat{\mu}_i, \mu_i - \mu_j \rangle$$

The new cluster centers are estimated through the self-attention mechanism, i.e.

$$\widehat{\mu}'_k = \frac{1}{Z_k} \sum_{i=1}^{K} \sum_{j=1}^{n} \exp\left(\lambda \langle \widehat{\mu}_k, x_{i,j} \rangle\right) x_{i,j}$$

where $\lambda > 0$ is a scaling factor and Z_i is defined as

$$Z_k = \sum_{i=1}^{K} \sum_{j=1}^{n} \exp\left(\lambda \langle \widehat{\mu}_k, x_{i,j} \rangle\right)$$

Theorem 1. *With sufficiently large d and $n \gg d$, with a probability $1 - O(K/n^2)$, we have*

$$\frac{\langle \mu_k, \widehat{\mu}'_k \rangle}{|\widehat{\mu}'_k|} \geq 1 - O\left(\frac{\log K + \log d}{d\Delta}\right)$$

Proof. Define $u_{i,j} = x_{i,j} - \mu_i$. We have

$$\widehat{\mu}'_k = \frac{1}{Z_k} \sum_{i=1}^{K} \exp\left(\lambda \langle \mu_i, \widehat{\mu}_k \rangle\right) \left\{ \left(\sum_{j=1}^{n} \exp\left(\lambda \langle \widehat{\mu}_k, u_{i,j} \rangle\right) \right) \mu_i + \sum_{j=1}^{n} \exp\left(\lambda \langle \widehat{\mu}_k, u_{i,j} \rangle\right) u_{i,j} \right\}$$

We first bound $\sum_{j=1}^{n} \exp\left(\lambda \langle \widehat{\mu}_k, u_{i,j} \rangle\right)$. Since $u_{i,j} \sim \mathcal{N}(0, \sigma^2 I/d)$ and $|\widehat{\mu}_k| = 1$, we know that $\langle \widehat{\mu}_k, u_{i,j} \rangle \sim \mathcal{N}(0, \sigma^2/d)$. Hence, with a probability $1 - 2\delta$, we have

$$\left| \sum_{j=1}^{n} \exp\left(\lambda \langle \widehat{\mu}_k, u_{i,j} \rangle\right) - n \mathrm{E}_{x \sim \mathcal{N}(0, \sigma^2/d)} \left[\exp(\lambda x)\right] \right| \leq 3 \exp\left(\lambda \sigma \sqrt{\frac{2}{d} \log \frac{n}{\delta}}\right) + 2 \sqrt{n \mathrm{E}_{x \sim \mathcal{N}(0, \sigma^2/d)} \left[\exp(2\lambda x)\right] \log \frac{2}{\delta}}$$

Since

$$\mathrm{E}_{x \sim \mathcal{N}(0, \sigma^2/d)} \left[\exp(\lambda x)\right] = \sqrt{\frac{d}{2\pi\sigma}} \int_{-\infty}^{+\infty} \exp\left(\lambda x - \frac{x^2 d}{2\sigma^2}\right) dx = \exp\left(\frac{\lambda^2 \sigma^2}{2d}\right)$$

we have, with a probability $1 - 2\delta$,

$$\left| \sum_{j=1}^{n} \exp\left(\lambda \langle \widehat{\mu}_k, u_{i,j} \rangle\right) - n \exp\left(\frac{\lambda^2 \sigma^2}{2d}\right) \right| \leq 3 \exp\left(\lambda \sigma \sqrt{\frac{2}{d} \log \frac{n}{\delta}}\right) + 2 \sqrt{n \exp\left(\frac{2\lambda^2 \sigma^2}{d}\right) \log \frac{2}{\delta}}$$

With large enough n, we have

$$3 \exp\left(\lambda \sigma \sqrt{\frac{2}{d} \log \frac{n}{\delta}}\right) + 2 \sqrt{n \exp\left(\frac{2\lambda^2 \sigma^2}{d}\right) \log \frac{2}{\delta}} \leq C \sqrt{n} \exp\left(\frac{\lambda^2 \sigma^2}{2d}\right)$$

and therefore

$$(1-\tau)n\exp\left(\frac{\lambda^2\sigma^2}{2d}\right) \leq \sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k, u_{i,j}\rangle\right) \leq (1+\tau)n\exp\left(\frac{\lambda^2\sigma^2}{2d}\right)$$

where

$$\tau \leq \frac{C}{\sqrt{n}}$$

Here $C>0$ is a universal constant.

We second bound $\sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k, u_{i,j}\rangle\right)u_{i,j}$. We write each $u_{i,j} = u_{i,j}^{\perp} + u_{i,j}^{\parallel}$, where $u_{i,j}^{\parallel} = \langle u_{i,j}, \widehat{\mu}_k\rangle\widehat{\mu}_k$ and $u_{i,j}^{\perp}$ is a $d-1$ dimensional Gaussian vector. We have

$$\sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k, u_{i,j}\rangle\right)u_{i,j} = \left(\sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k, u_{i,j}\rangle\right)\langle\widehat{\mu}_k, u_{i,j}\rangle\right)\widehat{\mu}_k + \sum_{j=1}^{n}u_{i,j}^{\perp}$$

Since $u_{i,j}^{\perp} \sim \mathcal{N}\left(0, \sigma^2 I_{d-1}/d\right)$, we have $\sum_{j=1}^{n}u_{i,j}^{\perp} \sim \mathcal{N}\left(0, n\sigma^2 I_{d-1}/d\right)$. Using the concentration of χ_{d-1}^2 distribution, we have, with a probability $1-\delta$

$$\left|\sum_{j=1}^{n}u_{i,j}^{\perp}\right|^2 \leq \frac{n\sigma^2}{d}\left(d-1+2\sqrt{(d-1)\log\frac{1}{\delta}}+2\log\frac{1}{\delta}\right) \leq n\sigma^2\left(1+3\sqrt{\frac{\log(1/\delta)}{d}}\right)$$

To bound $\sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k, u_{i,j}\rangle\right)\langle\widehat{\mu}_k, u_{i,j}\rangle$, following the same procedure, we have, with a probability $1-2\delta$

$$\begin{aligned}&\left|\sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k, u_{i,j}\rangle\right)\langle\widehat{\mu}_k, u_{i,j}\rangle - n\mathrm{E}_{x\sim\mathcal{N}(0,\sigma^2/d)}\left[\exp(\lambda x)x\right]\right| \\ \leq\ & 3\exp\left(\lambda\sigma\sqrt{\frac{2}{d}\log\frac{n}{\delta}}\right)\sigma\sqrt{\frac{2}{d}\log\frac{n}{\delta}} + \sqrt{n\mathrm{E}_{x\sim\mathcal{N}(0,\sigma^2/d)}\left[\exp(2\lambda x)x^2\right]\frac{2}{\delta}}\end{aligned}$$

Since

$$\mathrm{E}_{x\sim\mathcal{N}(0,\sigma^2/d)}\left[\exp(\lambda x)x\right] = \sqrt{\frac{d}{2\pi\sigma^2}}\int\exp\left(\lambda x - \frac{x^2 d}{2\sigma^2}\right)xdx = \frac{\lambda\sigma^2}{d}\exp\left(\frac{\lambda^2\sigma^2}{2d}\right)$$

and

$$\begin{aligned}&\mathrm{E}_{x\sim\mathcal{N}(0,\sigma^2/d)}\left[\exp(2\lambda x)x^2\right] \\ =\ & \sqrt{\frac{d}{2\pi\sigma^2}}\int\exp\left(2\lambda x - \frac{x^2 d}{2\sigma^2}\right)x^2dx \\ =\ & \sqrt{\frac{d}{2\pi\sigma^2}}\exp\left(\frac{2\lambda^2\sigma^2}{d}\right)\int\exp\left(\frac{d}{2\sigma^2}\left[x-\frac{\lambda\sigma^2}{d}\right]^2\right)\left(\left[x-\frac{\lambda\sigma^2}{d}\right]^2 + 2\frac{\lambda\sigma^2}{d}\left[x-\frac{\lambda\sigma^2}{d}\right] + \frac{\lambda^2\sigma^4}{d^2}\right)dx \\ =\ & \exp\left(\frac{2\lambda^2\sigma^2}{d}\right)\left(\frac{\lambda^2\sigma^4}{d^2} + \left(\frac{2\sigma^2}{d}\right)^{3/2}\mathrm{E}_{x\sim\mathcal{N}(0,1)}\left[x^2\right]\right) \\ =\ & \exp\left(\frac{2\lambda^2\sigma^2}{d}\right)\left(\frac{\lambda^2\sigma^4}{d^2} + 2\left(\frac{2\sigma^2}{d}\right)^{3/2}\right) \leq \frac{2\lambda^2\sigma^4}{d^2}\exp\left(\frac{2\lambda^2\sigma^2}{d}\right)\end{aligned}$$

We thus have, with a probability $1-2\delta$,

$$\begin{aligned}&\left|\sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k, u_{i,j}\rangle\right)\langle\widehat{\mu}_k, u_{i,j}\rangle - \frac{n\lambda\sigma^2}{d}\exp\left(\frac{\lambda^2\sigma^2}{2d}\right)\right| \\ \leq\ & 3\exp\left(\lambda\sigma\sqrt{\frac{2}{d}\log\frac{n}{\delta}}\right)\sigma\sqrt{\frac{2}{d}\log\frac{n}{\delta}} + \sqrt{\frac{4n\lambda^2\sigma^4}{d^2}\exp\left(\frac{2\lambda^2\sigma^2}{d}\right)\log\frac{2}{\delta}}\end{aligned}$$

When n is sufficiently large, we have, with a probability $1-2\delta$

$$(1-\tau)\,\frac{n\lambda\sigma^2}{d}\exp\left(\frac{\lambda^2\sigma^2}{2d}\right) \leq \sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k,u_{i,j}\rangle\right)\langle\widehat{\mu}_k,u_{i,j}\rangle \leq (1+\tau)\,\frac{n\lambda\sigma^2}{d}\exp\left(\frac{\lambda^2\sigma^2}{2d}\right)$$

where $\tau \leq C/\sqrt{n}$. By putting them together, with a probability $1-4\delta$, we have

$$\sum_{j=1}^{n}\exp\left(\lambda\langle\widehat{\mu}_k,u_{i,j}\rangle\right)u_{i,j} = n(1+\beta)\frac{n\lambda\sigma^2}{d}\exp\left(\frac{\lambda^2\sigma^2}{2d}\right)\widehat{\mu}_k + n\nu_i$$

with $\beta \in [1-\tau, 1+\tau]$ and

$$|\nu_i| \leq \frac{2\sigma}{\sqrt{n}}$$

Finally, we have, with a probability $1-4K\delta$

$$\mu_k' = \frac{n}{Z_k}\sum_{i=1}^{K}\exp\left(\lambda\langle\mu_i,\widehat{\mu}_k\rangle\right)\left((1+\alpha_i)\exp\left(\frac{\lambda^2\sigma^2}{2d^2}\right)\mu_i + (1+\beta_i)\frac{\lambda\sigma^2}{d}\exp\left(\frac{\lambda^2\sigma^2}{2d}\right)\widehat{\mu}_k + \nu\right)$$

Using the same analysis, we have, with a probability $1-4K\delta$,

$$\frac{Z_k}{n} = \exp\left(\frac{\lambda^2\sigma^2}{2d^2}\right)\sum_{i=1}^{K}\exp\left(\lambda\langle\mu_i,\widehat{\mu}_k\rangle\right)(1+\alpha_i)$$

Now, we can bound $|\widehat{\mu}_k' - \mu_k|$. We have

$$\begin{aligned}
&|\mu_k - \widehat{\mu}_k'| \\
&\leq \left|\frac{n}{Z_k}\exp\left(\lambda\langle\mu_k,\widehat{\mu}_k\rangle + \frac{\lambda^2\sigma^2}{2d^2}\right)(1+\alpha_k) - 1\right| + \frac{n}{Z_k}\sum_{i\neq k}\exp\left(\lambda\langle\mu_i,\widehat{\mu}_k\rangle + \frac{\lambda^2\sigma^2}{2d^2}\right)(1+\alpha_i) \\
&\quad + \frac{n\lambda\sigma^2|\mu_k-\widehat{\mu}_k|}{Z_k d}\sum_{i=1}^{K}\exp\left(\lambda\langle\mu_i,\widehat{\mu}_k\rangle + \frac{\lambda^2\sigma^2}{2d^2}\right)(1+\beta_i) + \frac{n}{Z_k}\sum_{i=1}^{K}\exp\left(\lambda\langle\mu_i,\widehat{\mu}_k\rangle\right)\nu_i
\end{aligned}$$

To further develop the bound for $|\mu_k - \widehat{\mu}_k'|$, we have

$$Z_k \geq n\exp\left(\frac{\lambda^2\sigma^2}{2d^2} + \lambda\langle\mu_k,\widehat{\mu}_k\rangle\right)\left(1-\frac{C}{\sqrt{n}}\right)$$

and

$$Z_k \leq n\exp\left(\frac{\lambda^2\sigma^2}{2d^2} + \lambda\langle\mu_k,\widehat{\mu}_k\rangle\right)\left(1+\frac{C}{\sqrt{n}}\right)\left(1+(K-1)\exp\left(-\lambda\Delta\right)\right)$$

We thus have

$$\begin{aligned}
&|\mu_k - \widehat{\mu}_k'| \\
&\leq \left|\frac{\exp\left(\lambda\Delta\right)}{\exp\left(\lambda\Delta\right)+K-1}\left(1-\frac{2C}{\sqrt{n}}\right)^2 - 1\right| + \left(1+\frac{2C}{\sqrt{n}}\right)^2\frac{K-1}{\exp(\lambda\Delta)}\left(1+\frac{\lambda\sigma^2}{d}|\mu_k-\widehat{\mu}_k|\right) \\
&\quad + \frac{\lambda\sigma^2}{d}|\mu_k-\widehat{\mu}_k| + \left(1+\frac{2C}{\sqrt{n}}\right)\frac{\sigma}{\sqrt{n}}
\end{aligned}$$

By choosing $\lambda = (\log d + \log K)/\Delta$, and by assuming n is significantly larger than d, we have

$$|\mu_k - \widehat{\mu}_k'| \leq O\left(\frac{\log d + \log K}{d\Delta}\right)$$

implying that

$$\frac{\langle\mu_k,\widehat{\mu}_k'\rangle}{|\widehat{\mu}_k'|} \geq 1 - O\left(\frac{\log d + \log K}{d\Delta}\right)$$

□