A APPENDIX

B GC VIT MODEL CONFIGURATIONS

GC ViT model configurations are presented in Table S.1 describing the choice of internal hyper parameters to obtain models with various compute load and parameter number.

	Output Size (Downs. Rate)	GC ViT-XT	GC ViT-T	GC ViT-S	GC ViT-B
Stem	128×128 (2×)	Conv, C:64, S:2, LN	Conv, C:64, S:2, LN	Conv, C:96, S:2, LN	Conv, C:128, S:2, LN
		[F-MBConv C:64] × 1	[F-MBConv C:64] × 1	[F-MBConv C:96] × 1	[F-MBConv C:128] × 1
Stage 1	56×56 (4×)	Conv, C:128, S:2, LN	Conv, C:128, S:2, LN	Conv, C:192, S:2, LN	Conv, C:256, S:2, LN
		[LG-SA, C:64, head:2] × 3, F-MBConv, C:128	[LG-SA, C:64, head:2] × 3, F-MBConv, C:128	[LG-SA, C:96, head:3] × 3, F-MBConv, C:192	[LG-SA, C:128, head:4] × 3, F-MBConv, C:256
Stage 2	28×28 (8×)	Conv, C:256, S:2, LN	Conv, C:256, S:2, LN	Conv, C:384, S:2, LN	Conv, C:512, S:2, LN
		[LG-SA, C:64, head:4] × 4, F-MBConv, C:256	[LG-SA, C:64, head:4] × 4, F-MBConv, C:256	[LG-SA, C:96, head:6] × 4, F-MBConv, C:384	[LG-SA, C:128, head:8] × 4, F-MBConv, C:512
Stage 3	14×14 (16×)	Conv, C:512, S:2, LN	Conv, C:512, S:2, LN	Conv, C:768, S:2, LN	Conv, C:1024, S:2, LN
		[LG-SA, C:64, head:8] × 6 , F-MBConv, C:512	[LG-SA, C:64, head:8] × 19, F-MBConv, C:512	[LG-SA, C:96, head:12] × 19, F-MBConv, C:768	[LG-SA, C:128, head:16] × 19, F-MBConv, C:1024
Stage 4	7×7 (32×)	Conv, C:1024, S:2, LN	Conv, C:1024, S:2, LN	Conv, C:1536, S:2, LN	Conv, C:2048, S:2, LN
		[LG-SA, C:64, head:16] × 5, F-MBConv, C:1024	[LG-SA, C:64, head:16] × 5, F-MBConv, C:1024	[LG-SA, C:96, head:24] × 5, F-MBConv, C:1536	[LG-SA, C:128, head:32] × 5, F-MBConv, C:2048

Table S.1 – Architecture configurations for GC ViT. LG-SA and Conv denotes local, global self-attention and 3×3 convolutional layer, respectively. GC ViT-XT, GC ViT-T, GC ViT-S and GC ViT-B denote XTiny, Tiny, Small and Base variants, respectively.

C ABLATION

C.1 GLOBAL QUERY

We performed ablation studies to validate the effectiveness of the proposed global query. Using the same architecture, instead of global query, we compute: (1) global key and value features and interact them with local query (2) global value features and interact it with local query and key. As shown in Table S.2, replacing global query may significantly impact the performance for image segmentation and downstream tasks such as object detection, instance segmentation and semantic segmentation.

	ImageNet	COCO		ADE20k
	top-1	AP^{box}	AP^{mask}	mIoU
w. Global KV	82.5	49.9	41.3	44.6
w. Global V	82.7	50.8	42.4	45.1
GC ViT-T	**83.4**	**51.6**	**44.6**	**47.0**

Table S.2 – Ablation study on the effectiveness of the proposed global query for classification, detection and segmentation.

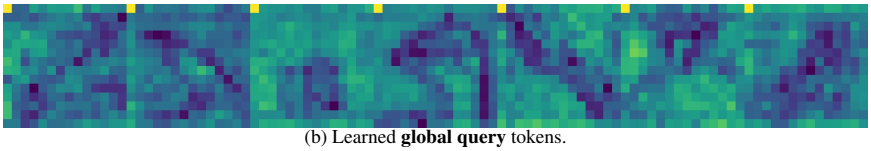
C.2 EMA AND BATCH SIZE

We also used used Exponential Moving Averages (EMA) and observed slight improvement in terms of ImageNet TOp-1 accuracy. Furthermore, the performance of the model across different batch sizes were stable as we did not observe significant changes. Table S.3 demonstrates the effect of EMA and batch size on the accuracy of a GCViT-T model.

Model	Local Batch Size	Global Batch Size	EMA	Top-1
GC ViT-T	32	1024	No	83.37
GC ViT-T	128	4096	No	83.38
GC ViT-T	32	1024	Yes	83.39
GC ViT-T	128	4096	Yes	83.40

Table S.3 – Ablation study on the effect of EMA and batch size on GC ViT-T ImageNet Top-1 accuracy.

(a) Original images from ImageNet-1K validation set.



(b) Learned **global query** tokens.

Figure S.1 – Visualization of : (a) input images (b) learned global query token feature maps.

D Interpretability

In Fig. S.1, we illustrate the learned global query token maps and demonstrate their effectiveness in capturing long-range contextual representations from different image regions.

E Training Details

For image classification, GC ViT models were trained using four computational nodes with 32 NVIDIA A100 GPUs. The total training batch size is 1024 (32 per GPU) for GC ViT-S, GC ViT-B, GC ViT-L and 4096 (128 per GPU) for GC ViT-XXT, GC ViT-XT and GC ViT-T. On average, each model required 32 hours of training with the specified hyper-parameters as indicated in the paper. All classification models were trained using the `timm` package (Wightman, 2019). Object detection and instance segmentation models as well as semantic segmentation models were trained using one computational node with 8 NVIDIA A40 GPUs using a total batch size of 16, hence a batch size of 2 per GPU. Detection and instance segmentation models were trained using `mmdetection` (Chen et al., 2019) package and on average required 56 hours of training. Semantic segmentation models were trained using `mmsegmentation` (Contributors, 2020) package, and on average required 34 hours of training.

F Complexity Analysis

Given an input feature map of $x \in \mathcal{R}^{H \times W \times C}$ at each stage with a window size of $h \times w$, the computational complexity of GC ViT is as follows

$$\mathcal{O}(\text{GC ViT}) = 2HW(2C^2 + hwC), \tag{7}$$

The efficient design of global query token generator and other components allows to maintain a similar computational complexity in comparison to Swin Transformer Liu et al. (2021) while being able to capture long-range information and achieve better higher accuracy for classification and downstream tasks such as detection and segmentation.

G Comparison to Other Global Self-attention Modules

Other efforts such as EdgeViT (Pan et al., 2022) in computer vision and BigBird (Zaheer et al., 2020) in NLP have proposed global self-attention in their respective applications. In this section, we discuss the differences between the proposed global self-attention in GC ViT and these efforts.

Model	Accuracy-Matched Frequency	Accuracy-Threshold-0.7
GC ViT-XXT	69.3	77.2
GC ViT-XT	71.3	78.8
GC ViT-T	73.1	80.5
GC ViT-S	73.8	80.7
GC ViT-B	74.4	81.1
GC ViT-L	74.5	81.5

Table S.4 – Classiication benchmarks of GC ViT models on ImageNetV2 dataset.

Model	Added Component	Top-1
Swin-T	None	81.3
Swin-T	GC Module	82.2
Swin-S	None	83.0
Swin-S	GC Module	83.7

Table S.5 – Ablation study on the effectiveness of Global Context (GC) module in Swin Transformers architecture on ImageNet Top-1 accuracy.

EdgeViT : EdgeViT and GC ViT use completely different self-attention blocks. The EdgeViT uses a series of local aggregation (convolution), sparse attention and local propagation(depthwise convolution), whereas GC ViT only uses an interleaved pattern of local and global self-attention layers without convolution in order to compute self-attention. The proposed global sparse attention in EdgeViT and GCViT are competently different. EdgeViT samples representative tokens and only computes sparse self-attention between these representative tokens with reduced feature size. On the contrary, GC ViT computes self-attention between the global queries (not just the token) and local keys and values without any subsampling in their respective local regions. Furthermore, in EdgeViT, only subsampled representative tokens per region interact In the self-attention module; however, in GC ViT, the global queries interact with the entire local regions, instead of interacting with each other, and hence provide an effective mechanism for capturing both short and long-range spatial dependencies.

In addition, GC ViT generates global query tokens by using a series of modified Fused MB-Conv from the entire image and without subsampling. Note that the resolution of global query tokens are the same as local query and values. However, in EdgeViT: (A) the representative tokens are obtained per local window, not the entire image, and by subsampling and reducing the feature resolution. Hence, since generated tokens have a lower resolution compared to their respective local windows, this could result in loss of spatial information and impact the effectiveness of self-attention. Unlike EdgeViT, the downsampler in GCViT also benefits from modified Fused-MBConv blocks which allows for modeling cross channel interactions and impose more locality and convolutional inductive bias.

BigBird : Bigbird, which is primarily introduced for NLP applications with 1D inputs, has significant differences compared to GC ViT, which is proposed for computer vision with mainly 2D inputs. Firstly, BigBird uses a combination of random, window and global attention mechanisms, which is different from the proposed local and global self-attention scheme in GC ViT. In addition, BigBird does not have any specific mechanisms for extracting global tokens as the existing tokens or additional special tokens can be specified as global tokens. On the contrary, the global tokens in GC ViT are extracted by the proposed global query generator module which consists of a series of modified Fused MB-Conv blocks to extract contextual information from the entire input features. Lastly, BigBird employs a set of global tokens which attend to the entire input sequence; in this case, select global query, key and values attend to local query, key and value tensors. However, as opposed to this formulation, in GC ViT, the global query tokens attend to local key and value tokens in partitioned windows. This is due to the fact that attending to the entire input sequence, as done in BigBird, is not feasible considering the larger size of input features in computer vision.

H IMAGENETV2 BENCHMARKS

In Table S.4, we have evaluated the performance of GC ViT on ImageNetV2 dataset (**?**) to further measure its robustness. Specifically, we have used different sampling strategies of Matched Frequency and Threshold-0.7. These benchmarks demonstrate the competetive performance of GC ViT on ImageNetV2 dataset and validates its effectiveness in robustness and generalizability.

I EFFECT OF GLOBAL CONTEXT MODULE

In order to demonstrate the effectiveness of Global Context (GC) module, we use Swin Transformers as the base model and add our propoped GC module. In this analysis, we remove the window shifting operation from Swin Transformers, since GC module is capable of modeling cross-region interactions. As shown in Table S.5, addition of GC module improves the ImageNet Top-1 accuracy by $+0.9\%$ and $+0.7\%$ for Swin Transformers Tiny and Small variants respectively.

J IMAGENET CLASSIFICATION BENCHMARKS

In Table S.6, we provide a comprehensive benchmark in terms of Top-1 accuracy for the models that are only trained on ImageNet-1K (Deng et al., 2009) dataset, and without additional data.

Table S.6 – Image classification benchmarks on **ImageNet-1K** dataset (Deng et al., 2009).

Method	Param (M)	FLOPs (G)	Image Size	Top-1 (%)
ResMLP-S12 (Touvron et al., 2021a)	15	3.0	224^2	76.6
PVT-v2-B1 (Wang et al., 2022)	13	2.1	224^2	78.7
GC ViT-XXT	12	2.1	224^2	79.8
EdgeViT-S (Pan et al., 2022)	11	1.9	224^2	**81.0**
DeiT-Small/16 (Touvron et al., 2021b)	22	4.6	224^2	79.9
T2T-ViT-14 (Yuan et al., 2021)	22	5.2	224^2	81.5
GC ViT-XT	20	2.6	224^2	**82.0**
ResNet50 (He et al., 2016)	25	4.1	224^2	76.1
PVT-Small (Wang et al., 2021)	24	3.8	224^2	79.8
Swin-T (Liu et al., 2021)	29	4.5	224^2	81.3
CoAtNet-0 (Dai et al., 2021)	25	4.2	224^2	81.6
Twins-SVT-S (Chu et al., 2021a)	24	2.9	224^2	81.7
PVT-v2-B2 (Wang et al., 2022)	25	4.0	224^2	82.0
ConvNeXt-T (Liu et al., 2022)	29	4.5	224^2	82.1
Focal-T (Yang et al., 2021b)	29	4.9	224^2	82.2
CSwin-T (Dong et al., 2022)	23	4.3	224^2	82.7
GC ViT-T	28	4.7	224^2	**83.4**
ResNet-101 (He et al., 2016)	44	7.9	224^2	77.4
ResMLP-S24 (Touvron et al., 2021a)	30	6.0	224^2	79.3
PVT-Medium (Wang et al., 2021)	44	6.7	224^2	81.2
T2T-ViT-19 (Yuan et al., 2021)	39	8.9	224^2	81.9
Twins-PCPVT-B (Chu et al., 2021a)	44	6.7	224^2	82.7
Swin-S (Liu et al., 2021)	50	8.7	224^2	83.0
Twins-SVT-B (Chu et al., 2021a)	56	8.6	224^2	83.2
ConvNeXt-S (Liu et al., 2022)	50	8.7	224^2	83.1
PVT-v2-B3 (Wang et al., 2022)	45	6.9	224^2	83.2
CoAtNet-1 (Dai et al., 2021)	42	8.4	224^2	83.3
Focal-S (Yang et al., 2021b)	51	9.1	224^2	83.5
CSwin-S (Dong et al., 2022)	35	6.9	224^2	83.6
GC ViT-S	51	8.5	224^2	**83.9**
ResNet-152 (He et al., 2016)	60	11.6	224^2	78.3
ViT-Base/16 (Dosovitskiy et al., 2020)	86	17.6	224^2	77.9
ResMLP-B24 (Touvron et al., 2021a)	116	23.0	224^2	81.0
PVT-Large (Wang et al., 2021)	61	9.8	224^2	81.7
DeiT-Base/16 (Touvron et al., 2021b)	86	17.6	224^2	81.8
CrossViT-B (Chen et al., 2021)	104	21.2	224^2	82.2
T2T-ViT-24 (Yuan et al., 2021)	64	14.1	224^2	82.3
CPVT-B (Chu et al., 2021b)	88	17.6	224^2	82.3
Twins-PCPVT-L (Chu et al., 2021a)	61	9.8	224^2	83.1
Swin-B (Liu et al., 2021)	88	15.4	224^2	83.3
CoAtNet-2 (Dai et al., 2021)	42	8.4	224^2	83.3
PVT-v2-B4 (Wang et al., 2022)	62	10.1	224^2	83.6
Twins-SVT-L (Chu et al., 2021a)	99	15.1	224^2	83.7
ConvNeXt-B (Liu et al., 2022)	89	15.4	224^2	83.8
Focal-B (Yang et al., 2021b)	90	16.0	224^2	83.8
PVT-v2-B5 (Wang et al., 2022)	82	11.8	224^2	83.8
CSwin-B (Dong et al., 2022)	78	15.0	224^2	84.2
BoTNet (Dong et al., 2022)	79	19.3	256^2	84.2
GC ViT-B	90	14.8	224^2	**84.4**
ConvNeXt-L (Liu et al., 2022)	198	34.4	224^2	84.3
CoAtNet-3 (Dai et al., 2021)	168	34.7	224^2	84.5
GC ViT-L	201	32.6	224^2	**84.6**