In this appendix, A contains additional results for DropKey. A.1 shows further ablations and A.2 shows the visualization on HUMBI. B contains additional implementation details for: CIFAR10/CIFAR100 B.1, ImageNet B.2, COCO B.3 and HUMBI B.4.

A Additional Results

A.1 Is Dropkey help model robust to occlusions?

To further verify that DropKey can alleviate local-bias problem, we study whether vision transformer with DropKey perform robustly in occluded scenarios, where some content of the image is missing. Specifically, a subset of patches is randomly selected and dropped before input to self-attention layer. In this experiment, with the drop ratio found in the manuscript, we randomly drop 10%, 30%, 50%, 70% and 90% of the patches to test the performance of trained model. For convenience, we define information loss as the ratio of dropped and total patches. To eliminate the influence of random occlusions, we report the mean of accuracy across 5 runs. The results reported in Table 1 show significantly robust performance of the model with DropKey against it with Dropout trick. For example, T2T24+Dropout achieves 75.7% accuracy in comparison to T2T24+DropKey which obtains 78.0% accuracy when 30% of the patches is removed. An surprising phenomenon can be observed that when 90% of the image information is randomly dropped, T2T19+DropKey and T2T24+DropKey still exhibits 32.7% and 29.7% accuracy, respectively. Consequently, compared with Dropout, models with DropKey show significant robustness to the content removal.

Table 1: Robustness against occlusion in images is studied under T2T19 and T2T24 on CIFAR100. The drop ratio is set as 0.3 for all models.

Information Loss	0.0	0.1	0.3	0.5	0.7	0.9
T2T19 [4]	78.4	77.6$^{-0.8}$	75.6$^{-2.8}$	73.1$^{-5.3}$	65.1$^{-13.3}$	28.2$^{-50.2}$
T2T19 + Dropout	78.5	77.4$^{-1.1}$	76.5$^{-2.0}$	73.9$^{-4.6}$	66.0$^{-12.5}$	30.2$^{-48.3}$
T2T19 + DropKey	80.1	79.4$^{0.7}$	78.2$^{-1.9}$	75.8$^{-4.3}$	68.7$^{-11.4}$	32.7$^{-47.4}$
T2T24 [4]	77.8	77.1$^{-0.7}$	75.2$^{-2.6}$	72.7$^{-5.1}$	63.5$^{-14.3}$	23.7$^{-54.1}$
T2T24 + Dropout	78.2	77.3$^{-0.9}$	75.7$^{-2.5}$	72.8$^{-5.4}$	65.3$^{-12.9}$	27.9$^{-50.3}$
T2T24 + DropKey	79.3	78.6$^{-0.7}$	78.0$^{-1.3}$	74.3$^{-5.0}$	66.9$^{-12.4}$	29.7$^{-49.6}$

A.2 Visualization on HUMBI

We present visualization on HUMBI in Figure 1 and results show that DropKey can encourage model to capture dense interactions with highly correlated vertices to learn the robust representation. METRO w/ Dropout only focus on vertices around elbow joint which ignores global context. For METRO w/ DropKey, it considers the interactions with vertices (eg. wrist joint and arm) which are helpful to predict precise location of target joint. This further demonstrates that the proposed DropKey can stimulate model to capture vital information in a global manner.

B Implementation Details

B.1 Details: CIFAR10 and CIFAR100

For CIFAR10 and CIFAR100, all models are trained with random initialization on all datasets and trained with 2 NVIDIA V100 GPUs. As shown in Table 2 and Table 3, we conduct grid search on learning rate and epoch number for T2T [4] and VOLO [5] to explore the best performance of vanilla backbones. The other hyper-parameters used in our experiments such as weight decay, Mixup and Cutmix follow [4, 5] on ImageNet.

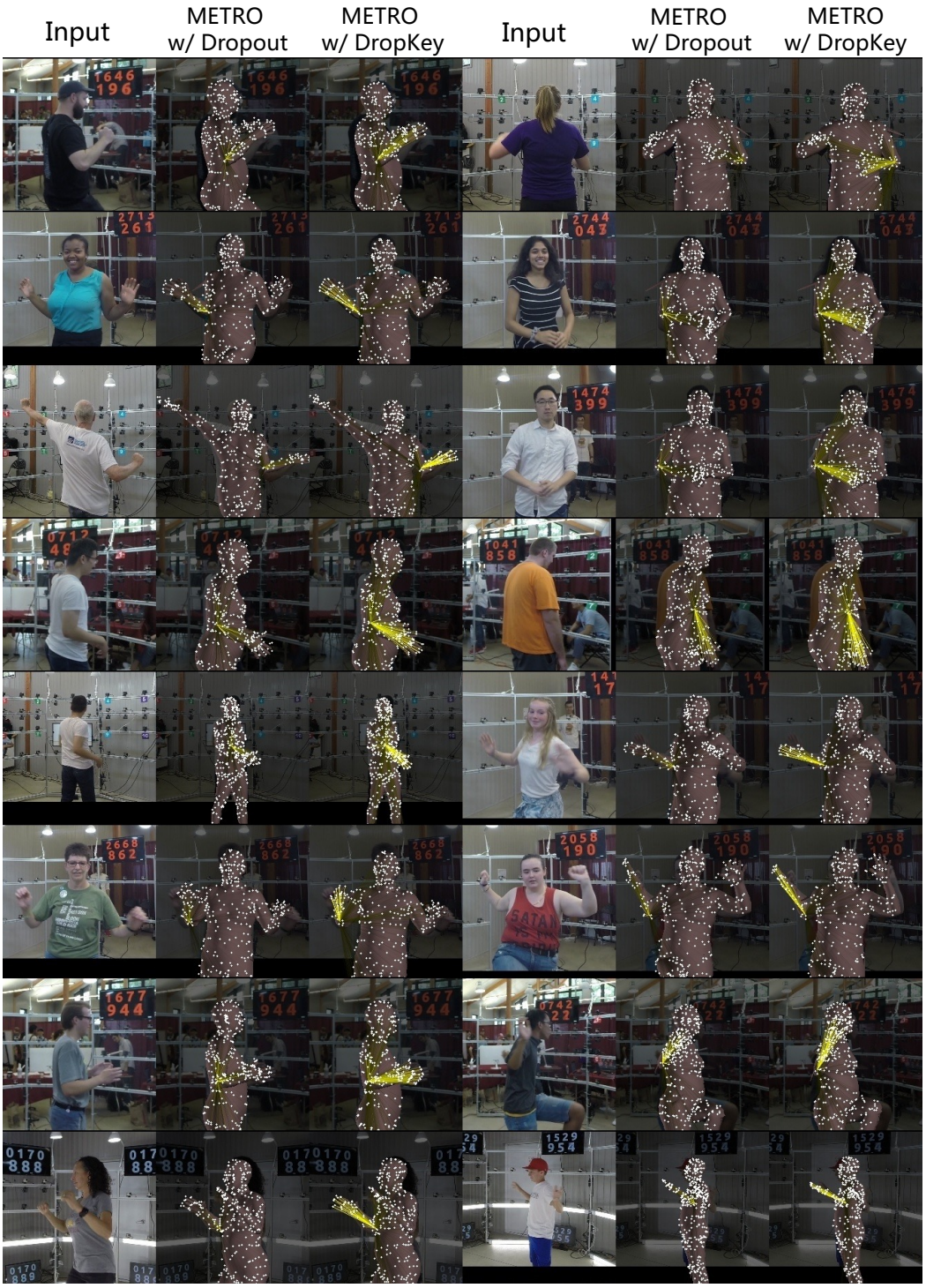

Figure 1: DropKey encourages model to learn robust interactions among body joints and mesh vertices for human mesh reconstruction. Given an input image, METRO w/ Dropout predicts elbow joint only by taking sparse interactions with mesh vertices which is related to elbow into consideration. DropKey encourages model to capture dense interactions with highly correlated vertices to learn the robust representation.

Table 2: The hyper-parameters for all T2T/VOLO models on CIFAR10.

Model	T2T14	T2T19	T2T24	VOLOd1	VOLOd2	VOLOd3
Epochs	400	500	600	600	600	600
Warmup Epochs	5	5	5	5	5	5
Batch Size	250	200	120	200	200	140
Learning Rate	5e-4	2e-4	1e-4	3e-4	3e-4	1e-4

Table 3: The hyper-parameters for all T2T/VOLO models on CIFAR100.

Model	T2T14	T2T19	T2T24	VOLOd1	VOLOd2	VOLOd3
Epochs	600	600	600	600	600	600
Warmup Epochs	5	5	5	5	5	5
Batch Size	200	160	120	200	200	140
Learning Rate	3e-4	3e-4	1e-4	3e-4	3e-4	1e-4

B.2 Details: ImageNet

For ImageNet, we strictly follow the training hyper-parameters in T2T [4], VOLO [5], CeiT [3] and DeiT [3] for reproduction. Due to the limited GPUs, 4 NVIDIA V100 GPUs are used for all models.

B.3 Details: COCO

For COCO, we strictly follow the training hyper-parameters in [1] for reproduction. Due to the limited GPUs, 8 NVIDIA V100 GPUs are used for all models.

B.4 Details: HUMBI

The METRO model design is the same as the official publication [2]. Firstly, we use Adam as the optimizer to train the model for 200 epochs and set the initial learning rate as 1e-4, which is decreased by a factor of 0.1 every 100 epochs. The batch size is set as 16. Then the model is fine-tuned with the learning rate of 1e-6 for 50 epochs. We implement the model with PyTorch and deploy it with 4 NVIDIA GeForce GTX 1080Ti.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[2] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021.

[3] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *ICCV*, pages 579–588, 2021.

[4] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021.

[5] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*, 2021.