GLID Supplementary Material

A. Additional Results

A.1. ImageNet Classification

We perform ImageNet-1K [7] classification with two settings (1) only using the backbone and (2) using the [CLS] token. We show the results in Tab. 1.

Method	MAE	iBOT	EsViT	SimMIM	GLID (1)	GLID (2)
LIN	67.8	79.5	81.3	56.7	75.9	76.2
FT	83.6	84.0	83.9	83.8	85.4	85.3

Table 1. Linear probing (LIN) and fine-tuning (FT) performance on ImageNet-1K.

A.2. Feature Pyramid Networks (FPN)

By default, GLID uses BiFPN [15] for interactions of the multi-scale feature maps. We also use popular MSDeformAttn following Deformable DETR [16]. The results are in Tab. 2.

FPN type	FLOPs	Params	ADE20K (mIoU)
BiFPN	33.8G	5.0M	52.7
MSDeformAttn	55.3G	5.5M	53.1

Table 2. Ablation of the FPN architectures.

A.3. Head Parameter Size

In Tab. 3, we show the numbers of parameters in different linear heads.

Keypoint	Det	Seg^{sem}	Seg^{ins}	Seg^{pan}	Depth
0.6M	0.5M	0.9M	0.6M	0.9M	1.3M

Table 3. Numbers of parameters of task heads.

A.4. Ablation of Fine-tuning Data

We conduct additional experiments using MAE and SimMIM pre-trainings to further ablate the impact of fine-tuning data, with results shown in Tab. 4. We observe that our encoder-decoder pre-training consistently outperforms other encoder-only pre-training methods.

% Data	**10**			**20**			**50**			**100**		
Method	SimMIM	MAE	GLID	SimMIM	MAE	GLID	SimMIM	MAE	GLID	SimMIM	MAE	GLID
mIoU↑	27.1	27.5	**31.2**	30.9	33.0	**35.0**	39.8	42.5	**46.3**	50.6	51.5	**52.7**
RMSE↓	0.471	0.403	**0.317**	0.401	0.363	**0.303**	0.384	0.341	**0.295**	0.343	0.340	**0.293**

Table 4. Fine-tuning with limited data.

B. Training Details

B.1. Pre-training

Hyper-parameters. The default setting is in Tab. 5. We use xavier_uniform [9] to initialize all Transformer blocks following original ViT [8]. By default, we use batch size of 1024 and scale the learning rate with linear rule, lr = base_lr × batch_size / 256 [10].

config	value
optimizer	AdamW [14]
base learning rate	1.5×10^{-4}
weight decay	0.05
optimizer momentum	β_1, β_2=0.9,0.95 [3]
learning rate schedule	cosine decay [13]
warmup epochs	40
augmentation	RandomResizedCrop

Table 5. Pre-training on ImageNet-1K [7].

B.2. Fine-tuning

Object detection. The default setting is in Tab. 6. We use the multi-scale augmentation strategy introduced in DETR [1] for data augmentation. We use a step-wise learning rate decay schedule and decay the learning by 10× at epoch of 40.

Image segmentation. The default setting is in Tab. 7. Following Mask2Former [4], we use random scale jittering between 0.5 and 2.0, random horizontal flipping, random cropping, and random color jittering for data augmentation. We use the crop size of 640×640. We apply the poly [2] learning rate schedule to decay the learning rate.

Pose estimation. The default setting is in Tab. 8. The default training setting in mmpose [6] is utilized for fine-tuning. The data augmentations include random flipping, half-body transformation, random scale, random rotation,

config	value
optimizer	AdamW
learning rate	1×10^{-4}
backbone learning rate	1×10^{-5}
batch size	16
weight decay	0.05
optimizer momentum	β_1, β_2=0.9,0.999
training epochs	50
drop path [11]	0.1

Table 6. Fine-tuning on COCO object detection.

config	value
optimizer	AdamW
learning rate	1×10^{-4}
backbone learning rate	1×10^{-5}
batch size	16
weight decay	0.05
optimizer momentum	β_1, β_2=0.9,0.999
training iterations	160K
drop path	0.1
decoder drop path	0.2

Table 7. Fine-tuning on ADE20K segmentation tasks.

and color jittering. The models are trained for 210 epochs, and we decay the learning by $10\times$ at the 170th and 200th epochs. We use layer-wise learning rate decay following [5].

config	value
optimizer	AdamW
learning rate	5×10^{-4}
batch size	512
weight decay	0.1
layer-wise decay[5]	0.8
optimizer momentum	β_1, β_2=0.9,0.999
training epochs	210
drop path	0.3

Table 8. Fine-tuning on COCO pose estimation.

Depth estimation. The default setting is in Tab. 9. The linear learning rate warm-up strategy is applied for the first 30% iterations and the cosine annealing learning rate strategy is adopted for the learning rate decay. Following BinsFormer [12], we utilize random flipping, random crop, random rotation, and color jittering for data augmentation.

config	value
optimizer	AdamW
learning rate	1×10^{-4}
batch size	16
weight decay	0.05
optimizer momentum	β_1, β_2=0.9,0.999
training iterations	38.4K
drop path	0.1

Table 9. Fine-tuning on NYUv2 depth estimation.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 1

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[3] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 1

[4] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1

[5] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. 2

[6] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[8] 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. In *ICLR*, 2021. 1

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 1

[10] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017. 1

[11] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 2

[12] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022. 2

[13] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2016. 1

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017. 1

[15] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 1

[16] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1