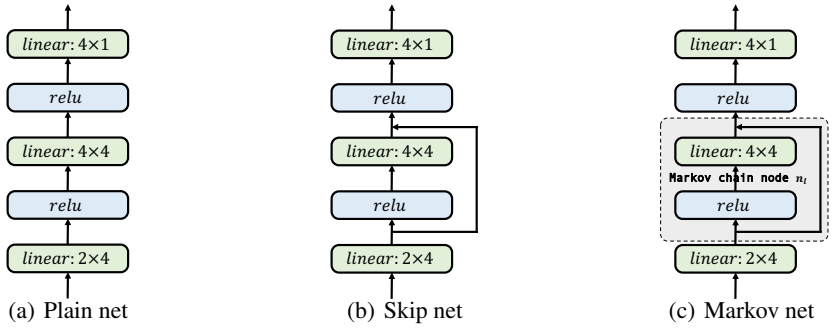
SUPPLEMENTARY MATERIALS

1 EXPERIMENTS DETAILS



(a) Plain net (b) Skip net (c) Markov net

Figure 1: Structure of plain net, skip net, and Markov net.

1.1 TOY MODEL

In order to better demonstrate the difference between plain net, skip net, and Markov net, we conduct a simple task in which maps (x, y) coordinate lies in the range of $[-5, 5]$ to a target distribution $x^2 - y^2$. As shown in Fig. 1, plain net, skip net, and Markov net have the same structure except that a skip connection is added to the skip net, and a penal connection is added to the Markov net. The number of total learnable parameters is 28 and all networks are initialized with the same value. We adopt an SGD Cherry *et al.* (1998) optimizer with a constant learning rate of 0.001, and momentum is 0.98, and train the model with a batch size of 32 for 10000 steps in total. As for the Markov net, we set τ to 0.0001.

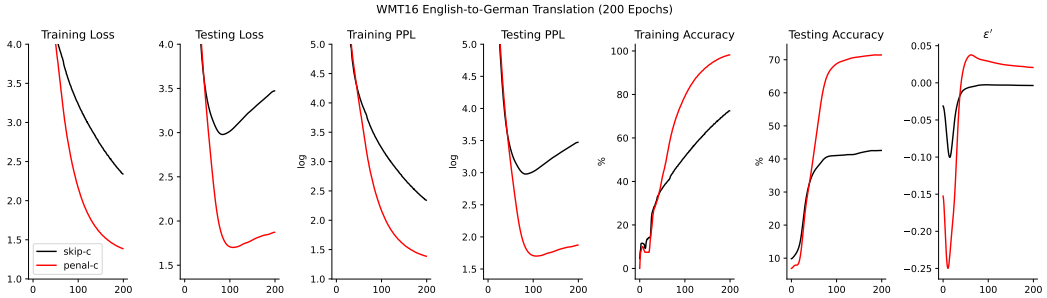
1.2 MULTI-MODAL TRANSLATION

We adopt the most widely used Transformer Vaswani *et al.* (2017) as our baseline. The model consists of 6 transformer blocks for the encoder and also 6 transformer blocks for the decoder. As mentioned above, each transformer block will be converted to two Markov chain nodes. Hence, the total length of the converted Markov chain is 12 for the encoder and 12 for the decoder, respectively. We follow the implementation in public project [1]. The embedding size d_{emb} is 512, and the source and target embedding layers are shared. In the meantime, we weight for target embedding layers is also shared with the last dense layer. Empirically, we set τ to 3×10^{-4}, which generalizes well to all translation tasks. Here, we opt for the mutual translation tasks between English and Germany. The full curve during the training process has been plotted in Fig. 2.

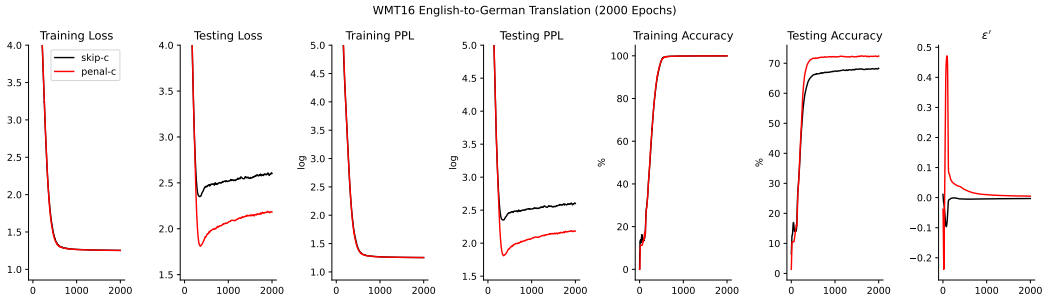
1.3 IMAGE CLASSIFICATION

We flow the implementation in public project[2] to conduct all experiments on ImageNet1k. In order to speed up training and also save memory, the automatically mixed precision (AMP) computing mode is enabled. No pre-trained models are used. More configurations have been listed in Tab. 1.

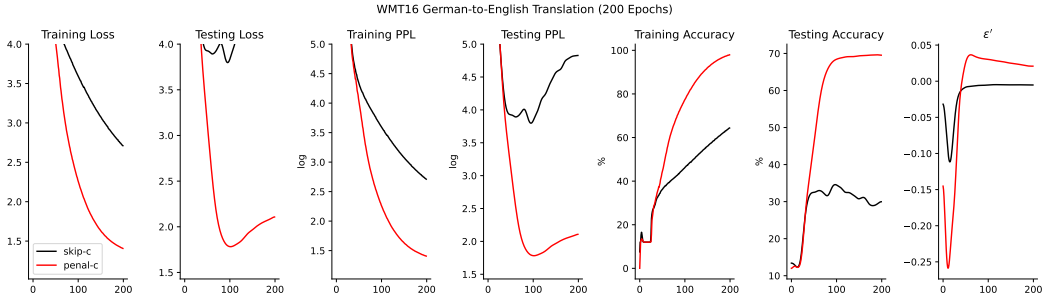
[1] https://github.com/jadore801120/attention-is-all-you-need-pytorch
[2] https://github.com/rwightman/pytorch-image-models



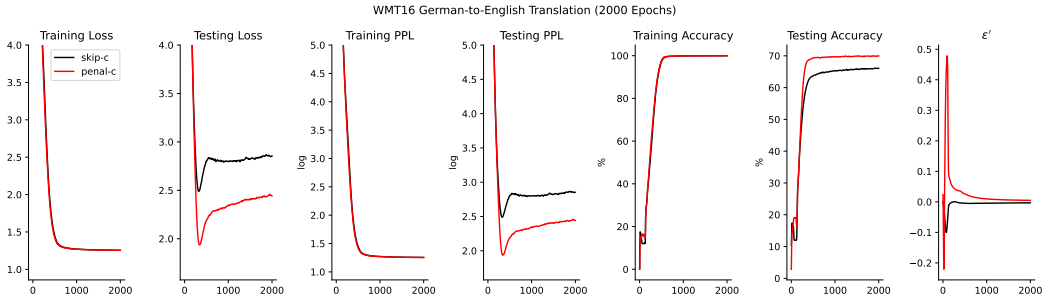
(a) WMT16 English to Germany Translation (Setting Q)



(b) WMT16 English to Germany Translation (Setting S)



(c) WMT16 Germany to English Translation (Setting Q)



(d) WMT16 Germany to English Translation (Setting S)

Figure 2: The training and testing curve on WMT16.

training config	ResNet50	ViT-S/16-224	DeIT-S/16-224	Swin-S/4-7-224
optimizer	sgd	adamw	adamw	adamw
base learning rate	0.6	0.001	0.001	0.001
weight decay	5e-5	0.05	0.05	0.05
optimizer momentum	0.9	(0.9,0.999)	(0.9,0.999)	(0.9,0.999)
batch size	1024	1024	1024	1024
training epoch	300	300	300	300
learning rate schedule	cosine	cosine	cosine	cosine
warmup epochs	10	10	5	20
warmup schedule	linear	linear	linear	linear
auto augment	true	false	false	false
rand augment	false	true	true	true
mixup prob.	0.2	0.2	0.2	0.2
cutmix prob.	0.0	0.0	1.0	1.0
random erasing prob.	0.0	0.25	0.25	0.25
label smoothing	0.1	0.1	0.1	0.1
EMA	disabled	0.999	disabled	0.999

Table 1: ImageNet1k training settings.

1.4 MODEL DEGRADATION

We adapt the basic skip connection model, i.e., ResNet He *et al.* (2016), as a baseline. In order to better invest the performance of different depth models, we build seven different models with different depths, i.e., $L \in \{18, 20, 32, 34, 44, 50, 56\}$. We follow the implementation for CIFAR10 in public project[3] and CIFAR100 in public project[4]. The SGD optimizer with a momentum of 0.9 is used for both datasets, and the weight decay is set at 0.0001 for CIFAR10 and 0.0005 for CIFAR100, respectively. The initial learning rate is 0.1, the batch size is 128 and τ is 3×10^{-9} for all experiments. We train the model for 200 epochs, and the learning rate will be multiplied by 0.1 at epochs 100, and 150 in the experiments on CIFAR10, and will be multiplied by 0.2 at epochs 60, 120, and 160 in the experiments on CIFAR100.

REFERENCES

J Michael Cherry, Caroline Adler, Catherine Ball, Stephen A Chervitz, Selina S Dwight, Erich T Hester, Yankai Jia, Gail Juvik, TaiYun Roe, Mark Schroeder, et al. Sgd: Saccharomyces genome database. *Nucleic acids research*, 26(1):73–79, 1998.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[3] https://github.com/akamaster/pytorch_resnet_cifar10

[4] https://github.com/weiaicunzai/pytorch-cifar100