ViBid: Linear Vision Transformer with Bidirectional Normalization (Supplementary Material)

Jeonggeun Song[1,*] **Heung-Chang Lee**[1,*]

[1]AI Lab & Service, Kakao Enterprise, Seongnam-si, South Korea
[*]Equal Contributions

Table 1: **Hyperparameter settings for our various models on ImageNet1k dataset.** Values in parentheses "( )" mean values used in fine-tuning.

Hyperparameter	ViBid-U	ViBid-T	ViBid-S	ViBid-M	ViBid-B
Learning rate	5e-5			4e-5 (0.01)	
Warm-up LR	1e-6 (None)				
Batch size	4096 (4096)				
Optimizer	AdamW (SGD)				
LR scheduler	Cosine (Cosine)				
Gradient clip	0.5 (0.5)				
Stochastic depth	0.0	0.05	0.1	0.15 (0.15)	0.25 (0.25)
Warm-up epochs	5 (0)				
RandAugment	2, 7			2, 9 (2, 9)	2, 12 (2, 12)
Label smoothing	0.1 (0.1)				
Train epochs	400 (10)				
Weight decay	0.05 (0.0)				

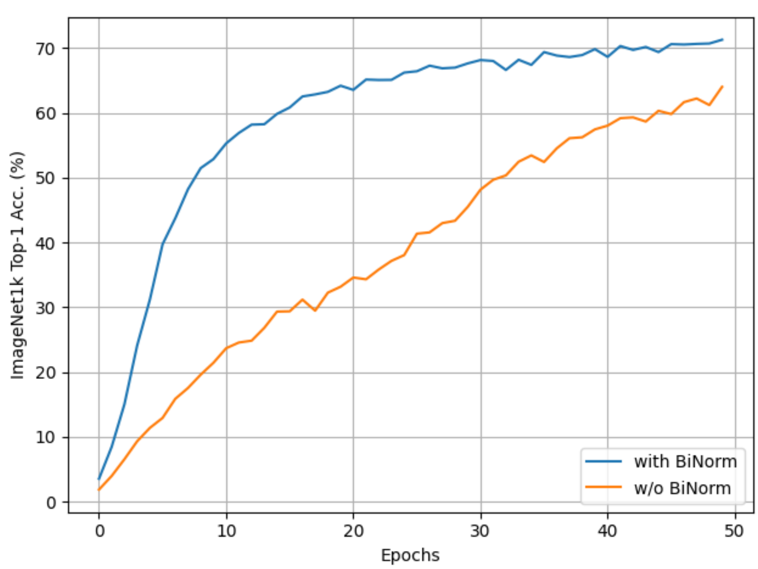

Figure 1: **Comparison of the effects of BiNorm presence or absence on early epoch training by using ViBid-M.** See Section 3 for details.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Algorithm 1: Python style pseudo-code of BiNorm-based attention.

```
def attend(self, x):
    b, n, d = x.shape
    qkv = self.qkv_proj(x)
    h = qkv.shape[-1] // 3
    qkv = qkv.reshape(b, n, 3, self.num_heads, h // self.num_heads)
    qkv = qkv.permute(2, 0, 3, 1, 4)

    q = output[0]
    k = output[1]
    v = output[2]

    # we commented the lines of the original SA
    # output = (q @ k.transpose(-2, -1)) * self.scale
    # output = output.softmax(dim=-1)
    output = k.transpose(-2, -1) @ v
    output = normalize(output, dim=-2)
    q = normalize(q, dim=-1)  # BiNorm

    # output = (output @ v).transpose(1, 2).reshape(b, n, h)
    output = (q @ output).reshape(b, n, h)
    output = self.proj(output)
    return output
```

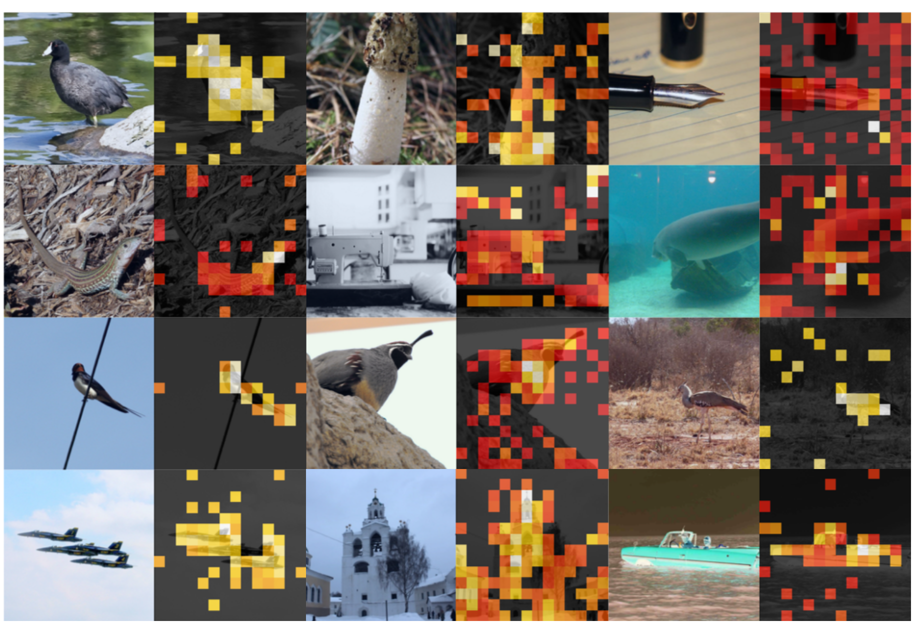

Figure 2: **Visualized attentions.** Visualization of attention matrices using pseudo-inverse scheme. These matrices are extracted from class attention module of pretrained ViBid-S.