A Appendix

A.1 Implementation details

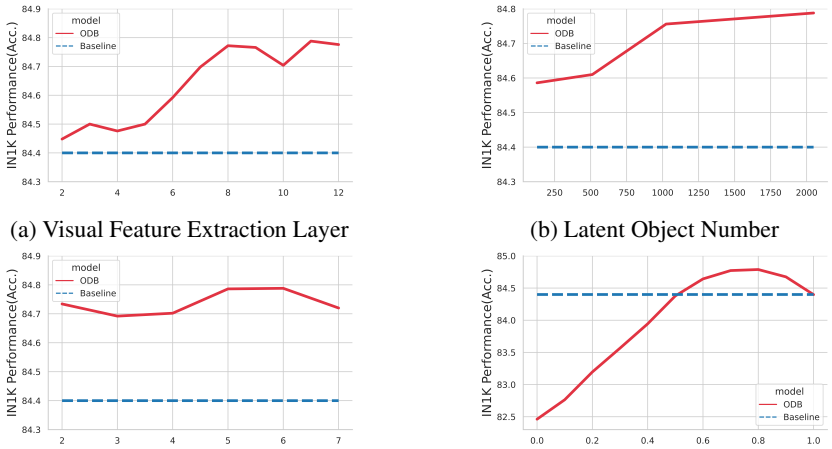
ViT architecture Our LLB is built upon pre-trained ViT backbones. We use ViT-Base and it's scaled version ViT-Large for LLB. Table 1 demonstrates detailed information about model variants. We follow the settings from (Dosovitskiy et al., 2020) for ViT parameters.

LLB stacks L_N layers of transformer layers to structure non-visual features. We report the impact of the number of layers on the LLB in Figure 1c, and selected values for L_N based on the results. Our LLB adds additional `MLP` layers for latent feature extraction and stacks transformer layers for non-visual feature structuring.For latent feature extraction, we use 2 layers of `MLP` with `ReLU` activation function.

	ViT				LLB					
Size	L_V	D	FF	H	L_N	D	FF	H	l_V	O
ViT-Base	12	768	3072	12	6	768	3072	12	11	2048
ViT-Large	24	1024	4096	16	6	1024	4096	16	23	2048

Table 1: Details of model variants

Hyper-parameter selection Depending on the input-domain and UWK in a task, the conflict may be cause by different numbers of objects in different layers. So we set the layer to extract the objects and their number as hyper-parameters for tuning by tasks. We also set the number of layers to structure non-visual and the value of α for integration as a hyper-parameter, and measured their influence on the IN1K classification task. The effective range of the hyper-parameters are shown in Figure 1a.



(a) Visual Feature Extraction Layer

(b) Latent Object Number

(c) Number of Structure Layer

(d) Integration Alpha

Figure 1: Impact of each hyper-parameter on IN1K image classification.

Training details We report our default training settings for IN1K image classification task in Table 2. For other evaluation benchmarks, only normalization values are changed. Table 3 reports the image classification performance on IN1K.

B Additional Qualitative Analysis Results

Object Clusters Figure 2 show successful examples of our latent object extraction. Each grid represents individual object cluster. We randomly sample clusters and clustered patches, and map them to the original image. For example, the second image in the first row has patterns like animal prints, and the second image in the second row has parts of fruit.

Setting	Value
Epochs	70
Batch size	1024
Optimizer	Adam (Kingma & Ba, 2014)
Optimizer Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
Learning rate:	
Schedule	Cosine
Peak	1e-4
Weight decay	5e-4
Loss	CrossEntropy
Augmentations:	
Size	224px or 384px
RandAugment (Cubuk et al., 2020)	
Magnitude	9
Normalize	
mean	[0.485, 0.456, 0.406]
std	[0.229, 0.224, 0.225]

Table 2: LLB training setting

Model	Pre.	Params (M)	Resolution Pre.	Resolution Fine.	Top1 (acc.) IN1K
ViT B/16	IN1K	86.57	224	224	79.00 (77.91)
+ LLB (Ours)	-	+46.45	224	-	79.43±.03
ViT B/16	IN21K	86.57	224	224	84.40 (83.97)
+ LLB (Ours)	-	+46.45	224	-	84.78±.01
ViT L/16	IN21K	304.33	224	224	85.68 (85.15)
+ LLB (Ours)	-	+80.80	224	-	85.92±.02
MAE B/16	IN1K	86.37	224	224	83.63 (83.60)
+ LLB (Ours)	-	+45.92	224	-	83.78±.02
MAE L/16	IN1K	304.33	224	224	86.08 (85.90)
+ LLB (Ours)	-	+80.80	224	-	86.12±.01
SWAG B/16	IB3.6B	86.37	224	384	85.28 (85.30)
+ LLB (Ours)	-	+45.92	224	-	85.35±.04

Table 3: Detailed top-1 accuracy on IN1K (accuracy in parenthesis: reference performance, red: positive, blue: negative).

Object Map on All Patches with Other Images Figure 3 show additional examples of object indices mapped to each patch of an image. In the mapped image in the *top* row, we found that patches of screwdriver are mapped to object 391 and the metal body patches are mapped to object 1736. From the frequency results in the right side, We can see that both feature are distinctive features for each class.

B.1 Empirical Analysis Results

We provide larger version of the visualization in Section 3.3.

C Additional Problem Confirmation and Comparison with LLB

Figure 5a from clearly shows the problem of the dominance of the visual-domain focused bias over the undescribed world knowledge over latent object in human labeling. The dots in the leftside figure represent the centroids of all features in each class of ImageNet, extracted from the ViT network trained on the data. When we zoomed in an area with closed centroids, we found five adjacent but semantically unrelated class labels, shown in the rightside.

We also confirm this problem with the Convolutional Neural Network (CNN). We follow the same procedure that is described in Section 3.1, but replace ViT with the well-known CNN network ResNet50 (Krizhevsky et al., 2017). We used two versions of ResNet50 pre-trained with ImageNet training data. First, we used the pre-trained ResNet50 (Krizhevsky et al., 2017) in a supervised manner. For supervised pre-trained ResNet50, we followed the details of (Krizhevsky et al., 2017)

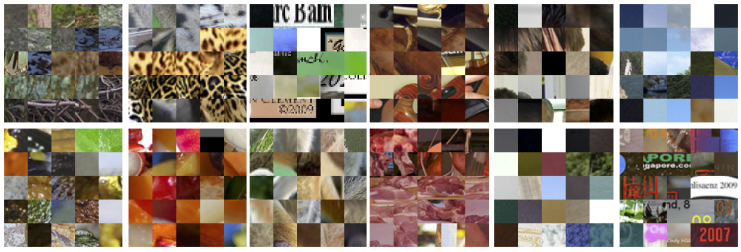


Figure 2: Positive examples of object latent clusters.

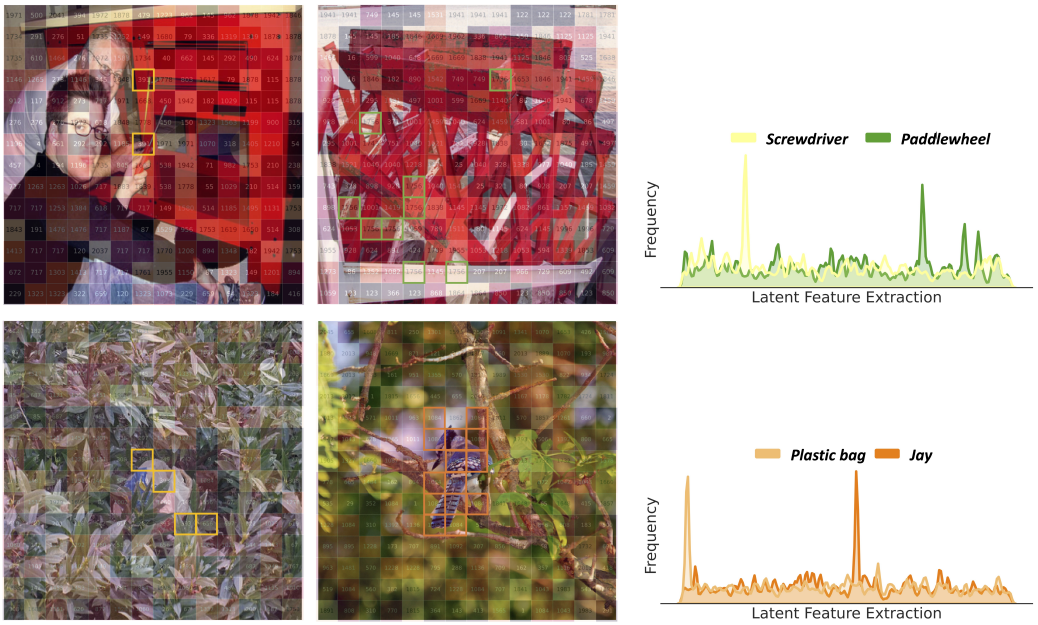


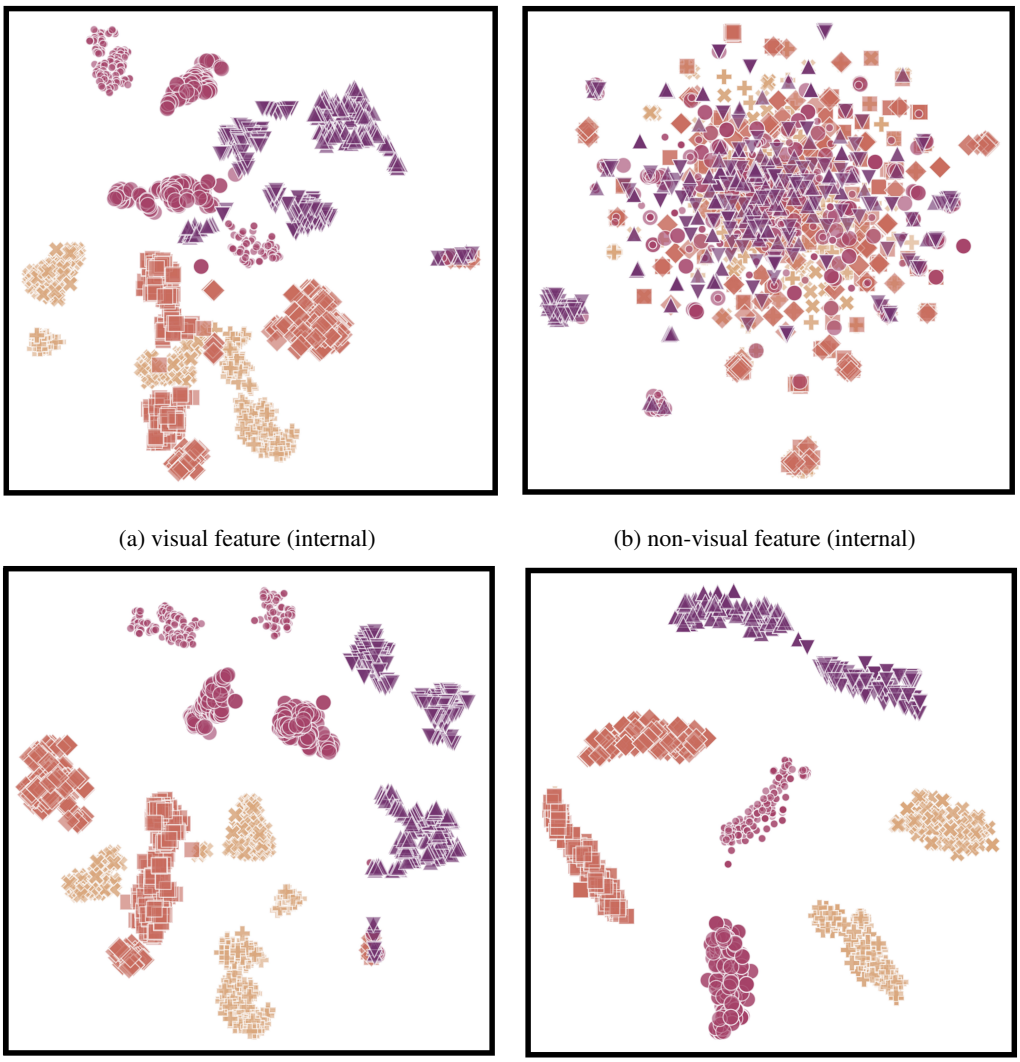
Figure 3: Additional example of object map on patches. In *left*, each tile shows an assigned object index to an image patch. *right* shows patch samples for the dominating objects and the frequency of the objects over all samples in each class.

and used parameters from open source[1] to reproduce a top-1 accuracy of 75.86% for IN1K (the reported performance from the open source is 76.13%). We also use ResNet50 pre-trained with self-supervised contrastive learning framework (Hadsell et al., 2006; Oord et al., 2018). Momentum Contrast (MoCo) (He et al., 2020) interpreted contrastive learning as dictionary look-up and built dynamic dictionaries with momentum-based moving average updates. MoCo v2 (Chen et al., 2020b) improved MoCo with the successes in (Chen et al., 2020a). We collected pre-trained ResNet50 weights using MoCo v2 from open source[2]. We then fine-tuned it using IN1K with the details described in (Chen et al., 2020b), and reproduced 77.01% top-1 accuracy in IN1K

C.1 Input-Domain focused Bias in CNN

Figure 5c shows the results of CNN in the classification benchmarks. In comparison with the bias in ViT 5a, semantically distinct classes ('840: Mop', '462: Broom', '764: Puck', and '523: Crutch') are still closely located, which is the shared input-focused inductive bias of the dataset. This observation is an evidence for the conflict of the input-domain focused bias even in CNN.

[1]ResNet50: https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html
[2]MoCo v2: https://github.com/facebookresearch/moco/tree/main



(a) visual feature (internal)

(b) non-visual feature (internal)

(c) visual feature (final)

(d) non-visual feature (final)

Figure 4: Feature distribution results.

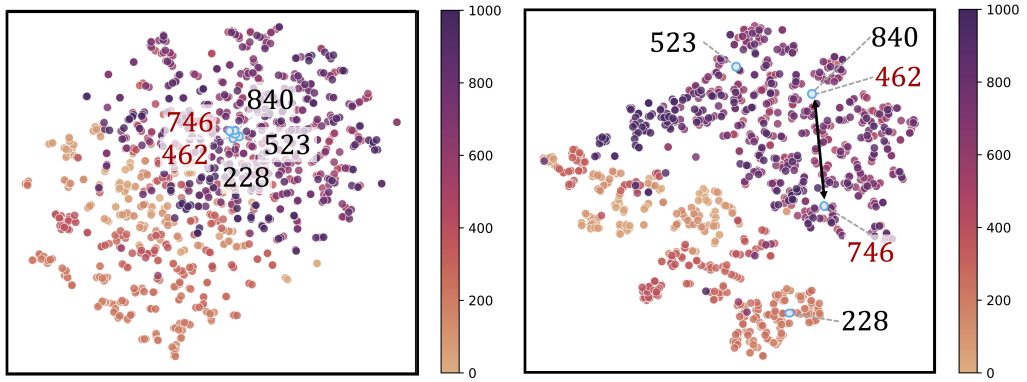
C.2 Input-Domain focused Bias in CNN with Contrastive Learning

Figure 5d shows the results of the CNN trained with contrastive learning. Using contrastive learning, the centroids of some classes (e.g. '523:Crutch' against '840: Mop', '462: Broom', '764: Puck') are slightly decoupled compared to supervised learning. However, this approach still fails to widen the gap between '462: Broom' and '746: Puck', where two class labels are visually similar in stick parts, but semantically distinguished by other objects. This observation shows that the input-domain focused bias still strongly used in determining the features.

C.3 Comparison with Label-focused Latent-object Biasing

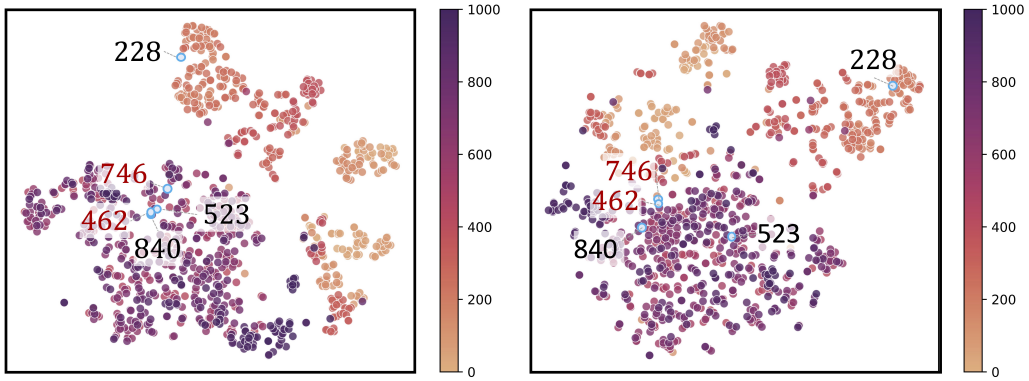
Figure 5b shows the results of LLB using the same classes in Figure 5e. Compared to ViT (Figure 5a), where the centroids of all features of five classes are closed located, LLB shows distant gaps between classes. Also, while other networks fail to widen the gap between '462: Broom' and '746: Puck', LLB placed them in a distant location.

Additionally, we can see that '840: Mop' and '462: Broom' are closed located in LLB. We hypothesise that, the way of structuring over components of mop and broom are similar, making LLB to generate their features in a close location. In contrast, the other methods placed '840: Mop' and '462: Broom' in relatively more distant locations. This observation implies that LLB can diminish the dominance of the visual input-domain focused bias, and introduce a distinct bias, regarded as the label-focused inductive bias.



(a) ViT

(b) LLB (Ours)



(c) CNN with supervised pre-training

(d) CNN with contrastive learning

(e) Image Samples for Adjacent Classes

Figure 5: Comparison of the distribution of centroids of all features in each class of ImageNet. Centroids of all output features from ViT: (a), LLB (Ours): (b), CNN with supervised pre-training: (c), CNN with contrastive learning: (d). We highlighted the dots of five classes in (e) with sky-blue color.

REFERENCES

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.