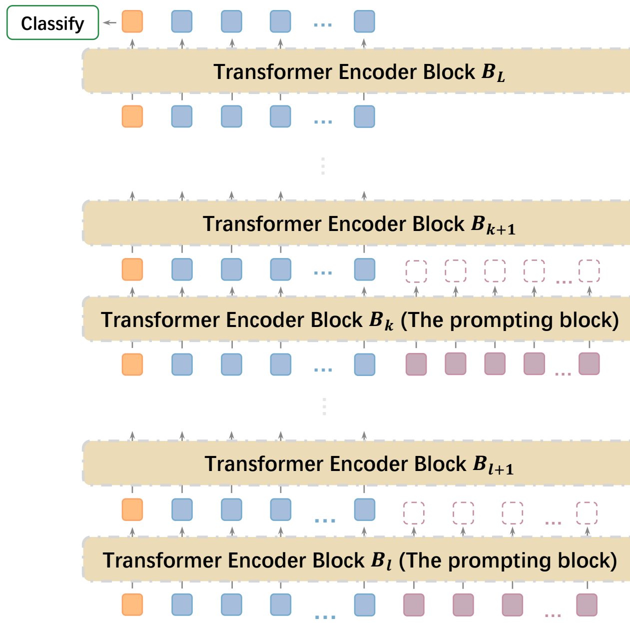

Figure 6: The illustration of TransHP with multiple layers of hierarchy. k and l are two insider layers, and L is the final layer.

Table 5: The balance parameters used for $\mathcal{L}_{coarse}$ of different levels (The last 1 is the balance parameter for the final classification.). "-" denotes that this transformer layer does not have prompt tokens.

λ	0	1	2	3	4	5	6	7	8	9	10	11
ImageNet	0.1	0.1	0.1	0.1	0.1	0.15	0.15	0.15	0.15	1	1	1
iNaturalist-2018	–	–	–	–	–	–	1	–	–	–	–	1
iNaturalist-2019	–	–	–	–	–	–	1	–	–	–	–	1
CIFAR-100	–	–	–	–	–	–	–	–	1	–	–	1
DeepFashion	–	–	–	–	–	–	0.5	–	1	–	–	1

A Multiple layers of hierarchy

We illustrate the TransHP in Fig. 6 when a dataset has multiple layers of hierarchy.

B Coarse-level classes of CIFAR-100

[0]: aquatic mammals, [1]: fish, [2]: flowers, [3]: food containers, [4]: fruit and vegetables, [5]: household electrical devices, [6]: household furniture, [7]: insects, [8]: large carnivores, [9]: large man-made outdoor things, [10]: large natural outdoor scenes, [11]: large omnivores and herbivores, [12]: medium mammals, [13]: non-insect invertebrates, [14]: people, [15]: reptiles, [16]: small mammals, [17]: trees, [18]: vehicles-1, and [19]: vehicles-2.

C Dataset details

The hierarchical labels of ImageNet are from WordNet [1], with details illustrated on Mike's website. Both the iNaturalist-2018/2019 have two-level hierarchical annotations: a super-category (14/6 classes) for the genus, and $8,142/1,010$ categories for the species. CIFAR-100 also has two-level hierarchical annotations: the coarse level has 20 classes, and the fine level has 100 classes. DeepFashion-inshop is a retrieval dataset with three-level hierarchy. To modify it for the classification task, we random select $1/2$ images from each class for training, and the remaining $1/2$ images for

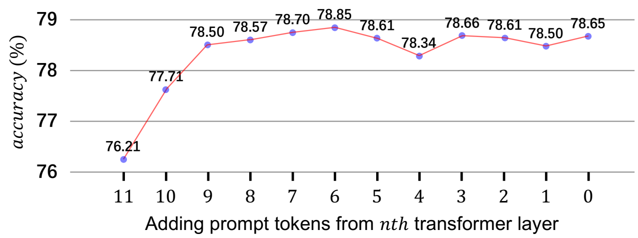

Figure 7: The top-1 accuracy on ImageNet *w.r.t* the transformer layer from which to add prompt tokens. The highest two transformer layers (which do not1 have too coarse-level labels) play an important role.

Table 6: The analysis of the number of coarse-level classes on the CIFAR-100 dataset. "N-class" denotes that there are N classes for the coarse-level classification.

Accuracy (%)	baseline	2-class	5-class	10-class	20-class
w/o Pre	61.77	63.34	63.12	64.47	67.09
w Pre	84.98	86.40	86.35	86.50	86.85

validation. Both the training and validation set contain 2 coarse classes, 17 middle classes, and $7,982$ fine classes, respectively.

D The balance parameters of different datasets

Please refer to Table 5 for the positions to insert prompt and corresponding balance parameters.

E Importance analysis of classification at different hierarchical levels

From Table 5 (Line 1), each transformer layer is responsible for one level classification. We remove the prompt tokens from the coarsest level to the finest level. In Fig. 7, n denotes that the prompt tokens are added from the nth transformer layer. We conclude that only the last two coarse level classifications (arranged at the 9th and 10th transformer layer) contribute most to the final classification accuracy. That means: (1) it is not necessary that the number of hierarchy and transformer layers are equal. (2) it is no need to adjust any parameters from too coarse level hierarchy. (Note that: though the current balance parameter for the 8th transformer layer is 0.15, when it is enlarged to 1, no further improvement is achieved.)

F Analysis of the number of coarse-level classes

As shown in Supplementary B, the CIFAR-100 dataset has 20 coarse-level classes. When we combine them into 10 coarse-level classes, we have ([0-1]), ([2-17]), ([3-4]), ([5-6]), ([12-16]), ([8-11]), ([14-15]), ([9-10]), ([7-13]), and ([18-19]). When we combine them into 5 coarse-level classes, we have ([0-1-12-16]), ([2-17-3-4]), ([5-6-9-10]), ([8-11-18-19]), and ([7-13-14-15]). When we combine them into 2 coarse-level classes, we have ([0-1-7-8-11-12-13-14-15-16]) and ([2-3-4-5-6-9-10-17-18-19]). The experimental results are listed in Table 6.

We observe that: 1) Generally, using more coarse-level classes is better. 2) Using only 2 coarse-level classes still brings over 1% accuracy improvement.

G The comparison with the "No prompts" baseline

In this section, we provide more experiments with the "No prompts" baseline. The detail of the "No prompts" baseline is shown in Fig. 4 (2). The experimental results are shown in Table 7. We find that

Table 7: Comparison between TransHP with the original baseline and the “No prompts" baseline.

Accuracy (%)	iNat-2018	iNat-2019	CIFAR-100	DeepFashion
Baseline (w/o Pre)	51.07	57.33	61.77	83.42
No prompts (w/o Pre)	51.88	58.45	63.78	84.23
TransHP (w/o Pre)	**53.22**	**59.24**	**67.09**	**85.72**
Baseline (w Pre)	63.01	69.31	84.98	88.54
No prompts (w Pre)	63.41	70.73	85.50	89.59
TransHP (w Pre)	**64.21**	**71.62**	**86.85**	**89.93**

Table 8: The top-1 accuracy of TransHP on some other datasets (besides ImageNet) with standard ViT-B/16 backbone. “w Pre” or “w/o Pre” denotes the models are trained from ImageNet pre-training or from scratch, respectively.

Accuracy (%)	iNaturalist-2018	iNaturalist-2019	CIFAR-100	DeepFashion
ViT-B/16 (w/o Pre)	52.96	58.24	62.91	84.28
TransHP (w/o Pre)	54.33	60.14	69.32	86.82
ViT-B/16 (w Pre)	64.10	70.22	87.13	89.14
TransHP (w Pre)	66.43	73.14	88.76	90.31

though “No prompts" baseline surpasses the original baseline, our TransHP still shows significant superiority over this baseline.

H More experiments with the ViT-B/16 backbone

In this section, we provide more experiments with the standard ViT-B/16 backbone. The experimental results are shown in Table 8. We find that no matter with pre-trained models or without, the TransHP achieves consistent improvement on all these datasets.

I Additional L_{coarse} with DeiT.

We introduce the experimental results by only adopting L_{coarse} in DeiT. Note that the L_{coarse} is imposed on the class token as shown in Fig. 4 (2). We find that the TransHP still shows performance improvement compared with only using L_{coarse} on DeiT-S and DeiT-B: compared with DeiT-S (79.82%) and DeiT-B (81.80%), “only with L_{coarse}" achieves 79.98% and 81.76% while the TransHP achieves 80.55% and 82.35%, respectively.

J Efficiency Comparison

Due to the increase of parameters ($+2.7\%$ on our baseline and $+1.4\%$ on ViT-B for ImageNet) and the extra cost of the backward of several $L_{coarses}$, the training time increases by 15% on our baseline and 12% on ViT-B for ImageNet. For inference, the computation overhead is very light. The baseline and TransHP both use around 50 seconds to finish the ImageNet validation with 8 A100 GPUs.