



# MLT-Trans: Multi-level Token Transformer for Hierarchical Image Classification

Tanya Boone Sifuentes<sup>1</sup>, Asef Nazari<sup>1</sup>, Mohamed Reda Bouadjenek<sup>1</sup>,  
and Imran Razzak<sup>2</sup>

<sup>1</sup> Deakin University, Waurin Ponds, VIC 3216, Australia

{tboonesifuentes, asef.nazari, reda.bouadjenek}@deakin.edu.au

<sup>2</sup> University of New South Wales, Sydney, NSW 2052, Australia

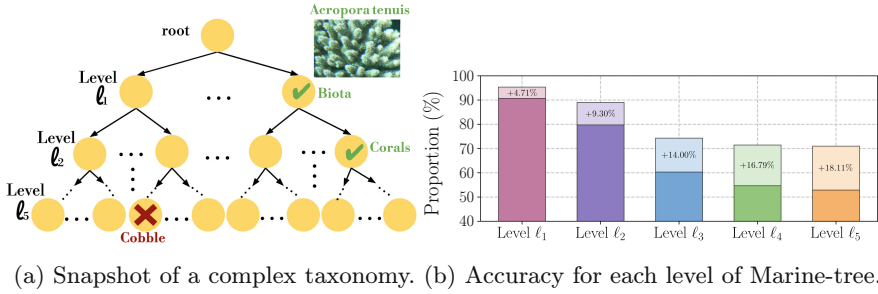
imran.razzak@unsw.edu.au

**Abstract.** This paper focuses on Multi-level Hierarchical Classification (MLHC) of images, presenting a novel architecture that exploits the “[CLS]” (classification) token within transformers – often disregarded in computer vision tasks. Our primary goal lies in utilizing the information of every [CLS] token in a hierarchical manner. Toward this aim, we introduce a Multi-level Token Transformer (MLT-Trans). This model, trained with sharpness-aware minimization and a hierarchical loss function based on knowledge distillation is capable of being adapted to various transformer-based networks, with our choice being the Swin Transformer as the backbone model. Empirical results across diverse hierarchical datasets confirm the efficacy of our approach. The findings highlight the potential of combining transformers and [CLS] tokens, by demonstrating improvements in hierarchical evaluation metrics and accuracy up to 5.7% on the last level in comparison to the base network, thereby supporting the adoption of the MLT-Trans framework in MLHC.

**Keywords:** Hierarchical classification · Image processing · Transformer · Class tokens · Hierarchy taxonomy

## 1 Introduction

Traditional image classification models often employ flat classification schemes, treating each category independently and disregarding potential hierarchical relationships that may exist between classes [1]. This lack of hierarchical consideration hinders the ability of algorithms to capture and leverage the underlying semantic structure of complex visual datasets [16]. This limitation is particularly influential in domains such as fine-grained categorization, where classes often exhibit nested relationships [12]. Recent works such as HERBS [6] and Metaformer [7] display the importance of incorporating supplementary information, such as background cues or object attributes, improving classification accuracy. However, HERBS and Metaformer are designed for flat classification,



**Fig. 1.** (a) An image of an “*Acropora tenuis*” classified by five independent classifiers as a “Biota”, “Corals”, and on the last level, incorrectly classified as a “Cobble” for the Marine-tree dataset. (b) Proportion of correctly classified images (depicted in a darker color) for each level of the taxonomy for Marine-tree dataset and the proportion of images incorrectly classified (depicted in a light color) but for which the other levels in the taxonomy were correctly identified.

ignoring the potential benefits that could come from incorporating hierarchical relationships as supplementary information.

MLHC has been attracting attention over the past few years mainly because it can provide order and structure to complex real-life datasets. In contrast to flat classification, Multi-level hierarchical classification (MLHC) aims to correctly classify objects organized in a tree-based taxonomy [15]. Moreover, it significantly contributes to the development of recommendation systems, image captioning, annotation, scene graph generation, and visual question answering (VQA) leading to more accurate and interpretable data analysis [19]. To the best of our knowledge, the majority of approaches for MLHC frequently neglect the incorporation of taxonomy structures and employ backbone networks whose outputs lack mutual constraint, leading to potential inconsistency in predictions [21,25,27]. While efforts have been made within the MLHC field, there is still considerable scope for future research in developing architectures that address these limitations. To illustrate the aforementioned advantages of MLHC, Fig. 1 provides a concrete example featuring the classification of an image by five distinct independent classifiers. Figure 1b displays the proportions of accurate classifications at each hierarchical level, represented in a darker color. Furthermore, these figures include the proportion of images that were misclassified at that particular level but were correctly identified in previous levels represented in a lighter color. If we hypothesize that coarser levels of the hierarchy exhibit higher accuracy compared to finer-grained levels, it becomes evident that promoting communication between these coarser levels can enhance the classification accuracy of the fine-grained levels. This highlights our belief that MLHC warrants further exploration as an appealing solution for addressing such issues.

## 2 Related Work

There have been several methods proposed for MLHC, and they can be categorized according to how the hierarchical structure is explored [22]. In particular, we distinguish: (i) *flat classification approaches*, typically involve a complete disregard for the class hierarchy, wherein predictions are made exclusively at the leaf nodes, with the implicit assumption that all ancestor classes are also attributed to the given instance. This provides an indirect solution to the hierarchical classification problem that could become suboptimal when dealing with a high number of classes, images with low intra-class, low inter-class separability or a combination [5, 12]; (ii) *local classification approaches*, a multi-class classifier is trained for every parent node within the class hierarchy. In the context of CNNs, we can distinguish HD-CNN [25] as one of the first attempts to embed CNNs using a taxonomy. As other predecessors, it lacks versatility and efficiency as it requires training several classifiers; and (iii) *global classification approaches* where in contrast, a single classifier is employed to manage the complete class hierarchy structure. Within this category, we can identify what is commonly referred to as “branch methods,” exemplified by models such as Branch-CNN (B-CNN) [27] and Hierarchical-CNN (H-CNN) [21]. These approaches incorporate a backbone network that produces hierarchical-level outputs (“branches”) at specific layers. While these methods improved performance, they are still susceptible to inconsistent predictions due to the absence of inter-level penalization.

As transformers continue to gain efficiency and importance in the field of Machine Learning, there have been some efforts to incorporate a taxonomy within these models. For instance, the Coarse-to-Fine Transformer, as proposed in [5], aims to enhance final classification accuracy. However, similar to other global approaches, it does not account for the hierarchical structure within the data. In contrast, the Nested Hierarchical Transformer (NesT) [26] and the ViT neural tree decoder (ViT-NeT) [14] introduce architectural designs that embrace the concept of taxonomy by generating decision trees. While these models demonstrate promising results, their effectiveness as MLHC methods remains challenging to evaluate because they do not present their results in hierarchical metrics.

The success of transformers can be attributed to several factors, one of them is their robust ability to model long-range dependencies. The Vision Transformer (ViT) proposed by Alexey Dosovitskiy et al. [9] introduced a special token, denominated [CLS] token or classification token, which was used to aggregate information from the entire sequence of the patch tokens from the image. This token was later replaced by average or global pooling as it would serve the same purpose. Within the domain of image segmentation, recent studies [8, 24] have emphasized the significance of incorporating [CLS] tokens to exploit class-specific information, ultimately yielding state-of-the-art results. This insight has served as motivation for harnessing class-specific information in the context of MLHC. However, our approach extends beyond the singular use of [CLS] tokens; we also seek to model the interrelationships between multiple levels of [CLS] tokens. In this work, we argue that the majority of existing MLHC transformer-

based models do not capture attention between different hierarchical levels and therefore, do not learn interactions between them. To tackle this issue, we propose MLT-Trans, which incorporates [CLS] token interactions between several levels of hierarchy. In summary, our main contributions are: (i) We propose MLT-Trans to exploit class-specific transformer attention for MLHC, (ii) We suggest a simple but effective transformer framework denominated Multi-level Token Transformer (MLT-Trans) which includes a multi-level token strategy to learn class-specific interactions between levels, (iii) We trained our proposed framework with sharpness-aware minimization and a hierarchical loss function based on the knowledge distillation loss, and finally, (iv) we performed extensive experimental evaluation on multiple baselines across diverse hierarchical datasets reporting our results in terms of hierarchical metrics.

### 3 Mathematical Notation and MLT-Trans

**Multi-level Hierarchical Classification:** The MLHC problem is defined by [2] as learning a mapping function  $f : \mathbb{X} \rightarrow \mathcal{Y}$ , which assigns to each feature vector  $\mathbf{x}^{(i)}$  a prediction vector  $\hat{\mathbf{y}}^{(i)} = (\hat{y}^{[ \ell_1 ]}, \hat{y}^{[ \ell_2 ]}, \dots, \hat{y}^{[ \ell_n ]})$  s.t.  $\hat{y}^{[ \ell_i ]}$  is the class label that  $f$  assigns for each level  $\ell_i$ .

**Taxonomy Encoding:** In contrast to flat classification, where classes are perceived as unrelated entities, hierarchical classification entails the organization of classes within a taxonomic structure. In this paper, our focus is only on *tree* taxonomies, which are organized with a hierarchy structure of  $n$  levels  $\ell_i$ , such that  $\ell_i \subset \mathcal{Y}$ ,  $\ell_1 \cup \ell_2 \dots \cup \ell_n = \mathcal{Y}$ , and  $\forall y_j \in \ell_{i+1}, \exists y_k \in \ell_i$  s.t.  $y_k \prec y_j$ , where  $\prec$  is the “subclass-of” relationship [2]. Lastly, we encode the relationship between two successive levels  $\ell_i$  and  $\ell_{i+1}$  in a taxonomy using an  $|\ell_i| \times |\ell_{i+1}|$  matrix  $M^{[ \ell_i, \ell_{i+1} ]}$ , where the binary value  $M_{y_k, y_j}^{[ \ell_i, \ell_{i+1} ]} \in \{0(y_k \not\prec y_j), 1(y_k \prec y_j)\}$ , with  $y_k \in \ell_i$  and  $y_j \in \ell_{i+1}$ .

**MLT-Trans:** Figure 2 shows an overview of our proposed MLT-Trans. Given an image  $\mathbf{I}$ , we first follow the conventional SwinTransformer-L [17] to obtain the patch embedding from the last transformer block  $\mathbf{Z}$ . According to [8], plugging cross-attention mechanisms after the last transformer block avoids large computational costs and additionally, there is no proof that performing these operations in the initial transformer blocks leads to a better performance. To calculate the attention between patch tokens  $\mathbf{Z}$  and class tokens for every level  $\ell_n$ , we first define class tokens as a matrix  $\mathbf{T}_{\ell_i}$  for each level of the taxonomy. The size of this matrix is the number of classes of each  $\ell_i$  and the embedding dimension of the transformer. To capture interactions between tokens  $\mathbf{T}_{\ell_i}$ , we calculate the multihead cross-attention [8] between each pair of tokens. Consider  $H$  as the number of heads where  $h \in \{1, \dots, H\}$ . We have the attention between  $\mathbf{T}_{\ell_i}$  and  $\mathbf{Z}$  for the  $h^{\text{th}}$  head as:

$$AT_h = \delta \left( \frac{Q_h(\mathbf{T}_{\ell_{i+1}})K_h(\mathbf{T}_{\ell_i})^T}{\sqrt{d_{K_h}}} \right) V_h(\mathbf{T}_{\ell_i}) \quad (1)$$

where  $\delta$  corresponds to the Softmax function and  $\sqrt{d_{K_h}}$  is the dimension of the key  $K_h$ . This attention mechanism occurs for each pair of subsequent levels. To calculate the attention between all levels, we decided to concatenate these attention outputs and use them to calculate the attention between them and the patch tokens  $\mathbf{Z}$  similar to Eq. (1). To update our multi-class tokens  $\mathbf{T}_{\ell_i}$ , we take the concatenated attention output  $AT$  and obtain the top- $k$  tokens with the highest activation where  $k = \ell_i$ . Consequently, we average it on the batch dimension to update our trainable token  $\mathbf{T}_{\ell_i}$ . Finally, to update our patch tokens  $\mathbf{Z}$ , we concatenate the norm of the concatenated attention that we calculated before and we represent this as  $\tilde{\mathbf{Z}}$ .

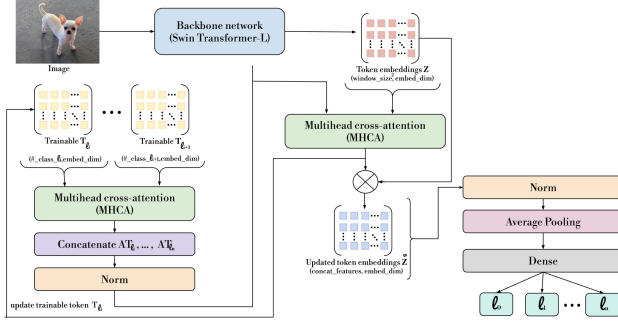


Fig. 2. MLT-Trans architecture.

**Hierarchical Loss Function:** In the realm of hierarchical image classification, the hierarchical loss function has become a foundational element [1]. Consequently, in this work we propose the computation of a loss, which is the summation of classification losses across all levels  $\ell_i$ . Our strategy takes inspiration from knowledge distillation loss proposed by Hinton et al. [11]. Essentially, this loss function is a pedagogical method in Machine Learning wherein a smaller model (denominated student model) learns from a larger, more complex model (teacher model) by capturing the teacher’s learned knowledge, typically in the form of softened probability distributions over classes. This process allows for model compression, enabling more efficient and compact models to retain the insights and generalization abilities of their larger counterparts, which can be especially beneficial for tasks demanding resource-efficient solutions. The knowledge distillation loss ( $KD_{Loss}$ ) [11] is defined as  $KD_{Loss} = CE(y, \delta(z_S)) + \lambda KL(\delta(z_T/\tau) || \delta(z_S/\tau))$ , where  $z_S$  are the logits coming from the student model and  $z_T$  are the logits from the teacher model. Also,  $CE$  stands for cross-entropy, and  $\delta$  stands for the softmax function; the Kullback-Leibler (KL) divergence loss is used to minimize the discrepancy between the soft outputs probabilities of the student and teacher model. These soft probabilities are obtained by dividing these logits by a temperature factor  $\tau$ . By transferring  $KD_{Loss}$  to the context of MLHC, we can consider the teacher model as the coarser level  $\ell_i$

**Table 1.** Summary of the hierarchical datasets employed in our experiments.

Dataset	Stanford Cars	CUB 200-2011	Marine tree	Stanford Dogs	FGVC Aircraft	Food 101
Training set	8,144	5,944	118,260	9,600	3,334	60,600
Validation set	4,020	3,000	16,126	2,400	3,333	15,150
Test set	4,021	2,071	26,798	8,580	3,333	25,250
#classes	196	200	60	120	100	101
<b>Taxonomy</b>						
#classes $\ell_1$	13	39	2	8	30	3
#classes $\ell_2$	113	123	10	120	70	15
#classes $\ell_3$	196	200	38	—	100	101
#classes $\ell_4$	—	—	46	—	—	—
#classes $\ell_3$	—	—	60	—	—	—
#classes $\ell_3$	—	—	60	—	—	—

and the student as the model coming from  $\ell_{i+1}$ . Specifically, we achieve this by replacing student logits by the prediction  $\hat{y}^{(j)[\ell_i]}$  and its taxonomic parent  $\tilde{y}^{(j)[\ell_{i-1}]}$  as the teacher. This will encourage penalization when there is a mismatch between these predictions. Performing this loss for every pair of levels, we obtain  $HKD_{Loss}$  as the summation of  $\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m CCE(y^{(j)[\ell_i]}, \delta(\hat{y}^{(j)[\ell_i]}))$  and  $\sum_{i=2}^n \sum_{j=1}^m \lambda K L(\delta(\tilde{y}^{(j)[\ell_{i-1}]}) / \tau) || \delta(\hat{y}^{(j)[\ell_i]} / \tau)$ , where CCE denotes the categorical cross-entropy function.  $\lambda$  and  $\tau$  are hyperparameters that need to be tuned to calibrate the importance of  $HKD_{Loss}$  and the effect of temperature on the probabilities. Note that to obtain the taxonomic parent of prediction  $\hat{y}^{(j)[\ell_i]}$  we use the taxonomy  $M^{[\ell_i, \ell_{i+1}]}$  explained in this section.

## 4 Experimental Evaluation

Our experiments draw upon a set of six hierarchical datasets and eight models which are detailed below. Our results are expressed in terms of hierarchical metrics [15] including Hierarchical Recall (HRecall), Hierarchical Precision (HPrecision), and Hierarchical F1 Score (HF1 Score). The other metrics are Consistency and Exact Match (proportion of examples that are correct and consistent) were also reported by Boone et al. [2].

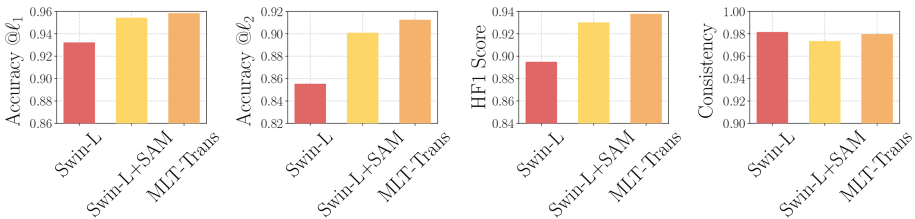
**Datasets:** We selected publicly accessible datasets, mainly employed for the purpose of coarse-to-fine classification, and subsequently transformed them into hierarchical datasets. This transformation involved utilizing existing semantic taxonomies wherever available, while also formulating our own taxonomic structures when none were provided. We’ve summarized this process in Table 1. Specifically, we performed experiments on FGVC-Aircraft [18], hierarchical versions of CUB-200-2011, Stanford Cars, and Marine-tree [3] where you can find their construction details used in [2].

- **Stanford Dogs** [13]: Following the guidelines established by the American Kennel Club (AKC), our classification scheme was aligned to include an additional hierarchical level that corresponds to the AKC’s seven distinct dog groups: “hound group”, “herding group”, “working group”, “sporting group”, “non-sporting group”, “toy group”, and “terrier group”. We added an additional super-class “wild dogs” for classes such as “dingo”, “dhole”, and “African hunting dog”.
- **Food-101** [4]: We transformed this dataset into a non-overlapping three-tier hierarchical structure. The initial level comprises three categories: “side dish,” “main dish,” and “dessert”. Subsequently, the second level is comprised of 15 classes: “beef dish”, “pork dish”, “rice dish”, “egg dish,” “poultry dish,” “pasta dish”, “seafood dish”, “sandwich”, “spread”, “appetizer”, “salad”, “soup”, “cake”, “pastry,” and “cup confections”.

**Implementation Details:** For our augmentation and training strategy we followed [6]. The entire experimental setup was implemented using Tensorflow, and was executed on an NVIDIA Tesla V100 GPU with a batch size of 8. Equal weights were assigned to all output components for the loss function. Additionally, we experiment with three different temperature and  $\lambda$  configuration similar to [20] and we selected the best of the three settings **Low(L)**( $\tau = 2.5, \lambda = 500$ ), **Medium(M)**( $\tau = 5, \lambda = 1000$ ), **High(H)**( $\tau = 10, \lambda = 2000$ ). Additionally, our MLT-Trans is trained using sharpness aware minimization (SAM) [10] with default settings. We trained eight state-of-the-art networks that display the highest accuracy on every datasets on their non-hierarchical representation. These baselines are: Swin Transformer, EfficientNetV2, InceptionNext, Uniformer, CoAtNet, CAFormer, TinyViT and MaxViT. You can find more details about the implementation of each baseline on the documentation from KerasCV [23].

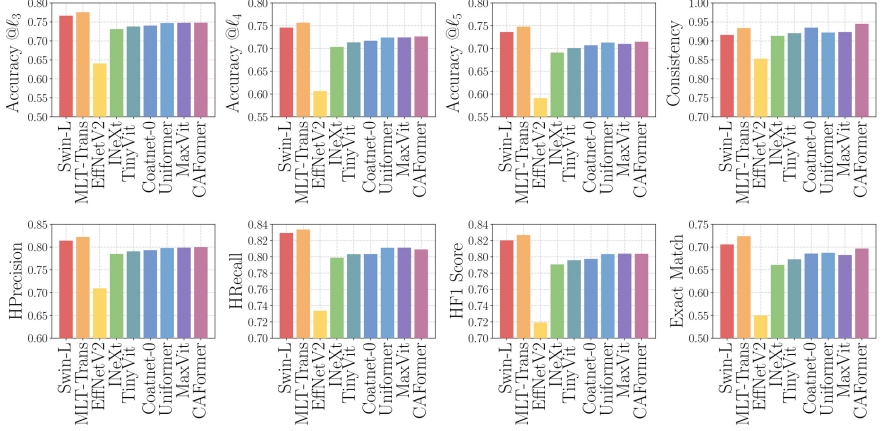
**Performance:** Figure 4 shows the comparison between our proposed architecture against the baseline models. From the obtained results, we make the following key observations: (i) Our proposed MLT-Trans improved their backbone network Swin-L in all datasets for almost all metrics. Especially accuracy  $\ell_i$  and Hierarchical F1-Score. (ii) The biggest increase in accuracy is observed at the last level while  $\ell_0$  is the level with the least increase in accuracy. This indicates that it is easier to improve the finer levels than the coarser levels even though we have a smaller number of classes. This problem could be related to over-fitting and top-down approaches because there is no feedback from the bottom levels to the top level. (iii) Stanford Dogs is the dataset that benefited the most from our proposed transformer, it improved 5.7% on its last level  $\ell_2$ . (iv) Stanford Cars presents an improvement of 3%, 2.7% and 1.7% respectively compared to Swin-L. Also, there is a 2.5% in HRecall meaning that the model is able to provide examples more relevant to the class. Although there is a small increase in consistency, there is a 3% increase in Exact Match due to the increase in accuracy for every level. Of all the baseline models, MLT-Trans is the model with the best

performance in all metrics. (v) Similarly, for CUB-210-2011 MLT-Trans is the model with the best performance in all metrics but with a smaller increase due to CUB-210-2011 being a dataset that is considered harder to train. The final accuracy is 92% which is only 1% less than the current state-of-the-art models HERBS [6] which also utilizes Swin-L as baseline model with the difference that we are not using extra training information such as background datasets. CAFormer presents slightly higher consistency than our method but our method outperforms their accuracy for all  $\ell_i$ . (vi) MLT-Trans for Food-101 presents the highest accuracy against all models for the last two levels with the exception of  $\ell_0$  where there is a small decrease and lastly (vii) Marine-tree, the most complex dataset to train, presents a 74.58% on its last level ( $\ell_5$ ) establishing a state-of-the-art performance. The performance in all metrics is superior by a large margin with respect to the baseline models. The closest model in performance is its baseline model Swin-L. (vii) The number of parameters for Swin-L is  $\approx 195\text{M}$  parameters, adding SAM minimization adds another 3M parameters and the complete MLT-trans architecture contains 270M parameters. Additionally, we want to point out that for datasets with a 3-level hierarchy, we only calculated the attention between the first two levels, and for Marine-tree we used 3 levels. Using all the levels would cause the attention calculation to slow the training process without a dramatic increase in performance. In conclusion, we observe that our proposed MLT-Trans manages to improve the results of their baseline model for most datasets but other datasets show that Swin-L might not be the best pick. For instance, for Stanford Dogs our InceptionNext implementation manages to beat the state-of-the-art performance by more than 1%, which opens the possibility of using InceptionNext as the backbone network for our proposed architecture. In terms of ablation study, Fig. 3 shows the performance of the backbone network Swin-L, the backbone network plus SAM, and finally our proposed MLT-Trans. The combination of Swin-L plus SAM increases both  $\ell_1$  and  $\ell_2$  and MLT-Trans increases them slightly more. We have observed that in some datasets, SAM causes a small decrease in Consistency compared to the backbone network. MLT-Trans is able to recover such loss and in some cases, it improves it. Nevertheless, the increase in accuracy that SAM provides is worth the small decrease in Consistency.

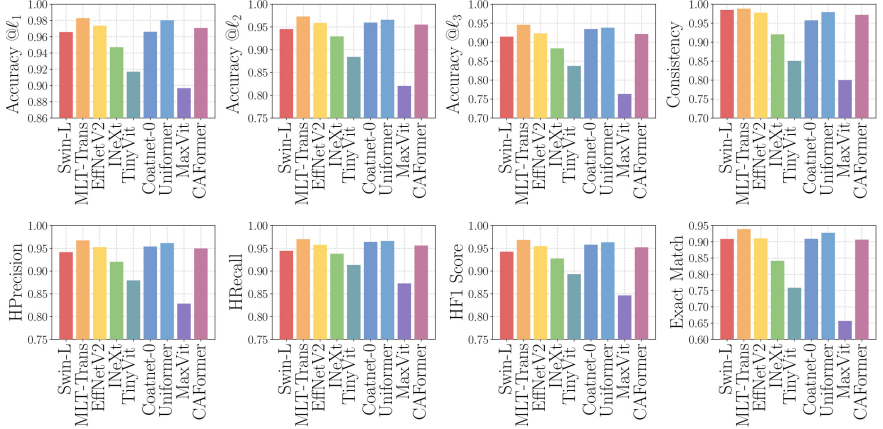


**Fig. 3.** Ablation study on Stanford Dogs.

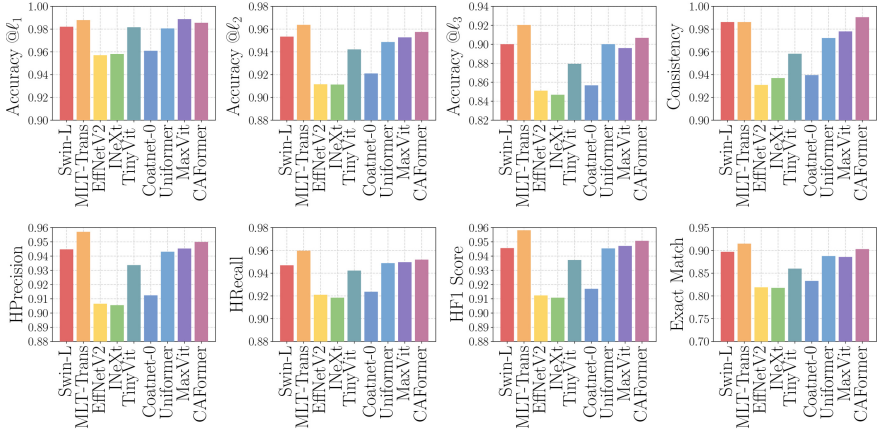




(a) Marine-tree

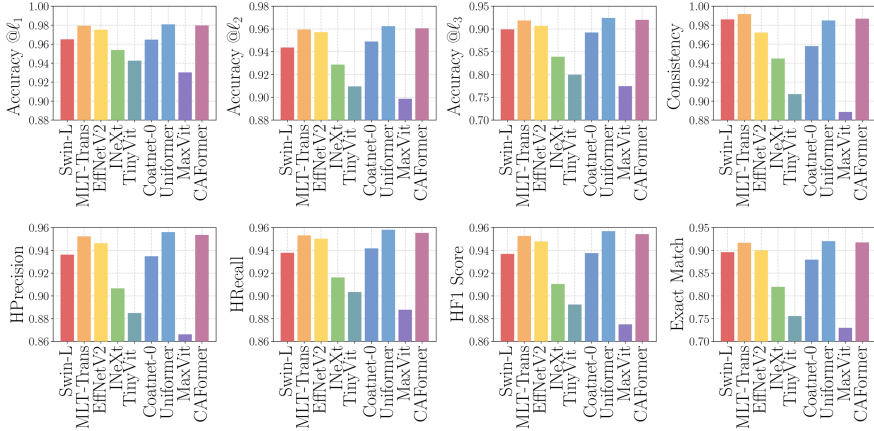


(b) Stanford Cars

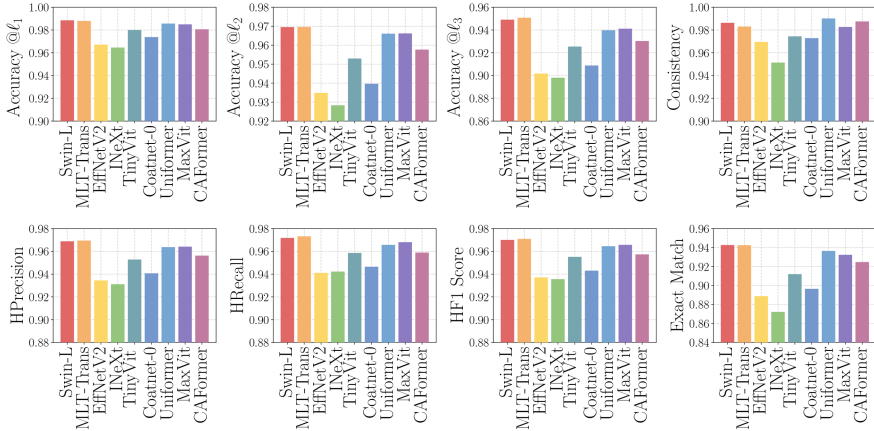


(c) CU-Birds-200-2011

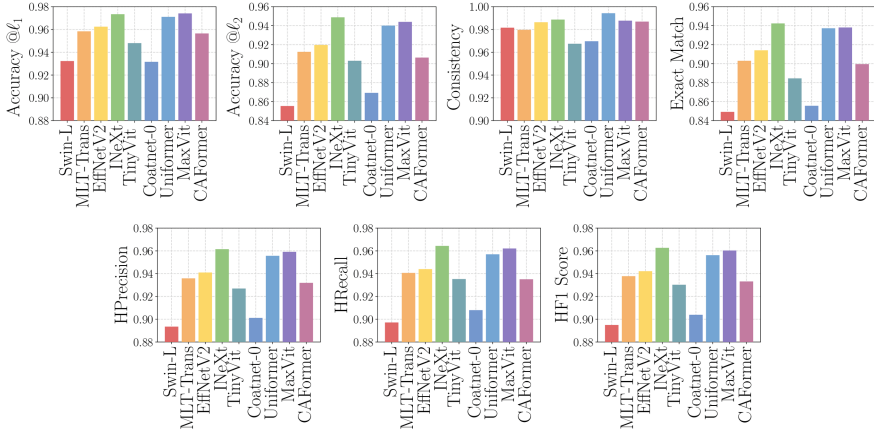
Fig. 4. Performance comparison.



(d) FGVC Aircraft



(e) Food-101



(f) Stanford Dogs

Fig. 4. (continued)

## 5 Conclusions and Future Work

In this work, we introduced a new transformer for Multi-level Hierarchical Classification we denoted Multi-level Token Transformer (MLT-Trans). Our proposed transformer implements an attention mechanism between trainable classification tokens ([CLS] tokens) per level of a predefined taxonomy plus the addition of a hierarchical loss based on knowledge distillation loss. Focusing on image classification, we presented a thorough experimental evaluation of the performance of our method on six different datasets varying from two levels of hierarchy up to five. Experiments on these datasets show that our proposed architecture improves several hierarchical metrics, most importantly accuracy, demonstrating that MLHC could be a strategy to beat state-of-the-art performances without doubling or triplicating the number of parameters. Future work includes testing our proposed methodology on different backbone networks as well as different hierarchical losses. Additionally, we could improve our attention mechanism to take into account all levels of hierarchical datasets with a large number of classes without being computationally expensive.

## References

1. Bertinetto, L., Mueller, R., Tertikas, K., Samangooei, S., Lord, N.A.: Making better mistakes: leveraging class hierarchies with deep networks. In: Proceedings of the IEEE/CVF Conference, pp. 12506–12515 (2020)
2. Boone-Sifuentes, T., Bouadjenek, M.R., Razzak, I., Hacid, H., Nazari, A.: A mask-based output layer for multi-level hierarchical classification. In: CIKM’22, pp. 3833–3837 (2022)
3. Boone-Sifuentes, T., et al.: Marine-tree: large-scale marine organisms dataset for hierarchical image classification. CIKM ’22, New York, NY, USA (2022)
4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 446–461. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10599-4\\_29](https://doi.org/10.1007/978-3-319-10599-4_29)
5. Chen, M., et al.: Coarse-to-fine vision transformer. arXiv preprint [arXiv:2203.03821](https://arxiv.org/abs/2203.03821) (2022)
6. Chou, P.Y., Kao, Y.Y., Lin, C.H.: Fine-grained visual classification with high-temperature refinement and background suppression. arXiv preprint [arXiv:2303.06442](https://arxiv.org/abs/2303.06442) (2023)
7. Diao, Q., Jiang, Y., Wen, B., Sun, J., Yuan, Z.: MetaFormer: a unified meta framework for fine-grained recognition. arXiv preprint [arXiv:2203.02751](https://arxiv.org/abs/2203.02751) (2022)
8. Dong, B., Zhou, P., Yan, S., Zuo, W.: Towards class interpretable vision transformer with multi-class-tokens. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), pp. 609–622. Springer (2022). [https://doi.org/10.1007/978-3-031-18913-5\\_47](https://doi.org/10.1007/978-3-031-18913-5_47)
9. Dosovitskiy, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
10. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. arXiv preprint [arXiv:2010.01412](https://arxiv.org/abs/2010.01412) (2020)

11. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531) (2015)
12. Huo, Y., Lu, Y., Niu, Y., Lu, Z., Wen, J.R.: Coarse-to-fine grained classification. In: Proceedings of the ACM SIGIR Conference, pp. 1033–1036. SIGIR'19 (2019)
13. Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: Stanford dogs. In: Proceedings of CVPR Workshop on Fine-Grained Visual Categorization (FGVC). vol. 2. Citeseer (2011)
14. Kim, S., Nam, J., Ko, B.C.: ViT-NeT: interpretable vision transformers with neural tree decoder. In: International Conference on Machine Learning, pp. 11162–11172. PMLR (2022)
15. Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., Androutsopoulos, I.: Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Min. Knowl. Disc.* **29**(3), 820–865 (2015)
16. Liu, Y., Dou, Y., Jin, R., Qiao, P.: Visual tree convolutional neural network in image classification. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 758–763. IEEE (2018)
17. Liu, Z., et al.: Swin Transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF Conference, pp. 10012–10022 (2021)
18. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013)
19. Parag, T., Wang, H.: Multilayer dense connections for hierarchical concept classification. arXiv preprint [arXiv:2003.09015](https://arxiv.org/abs/2003.09015) (2020)
20. Schmid, F., Masoudian, S., Koutini, K., Widmer, G.: Knowledge distillation from transformers for low-complexity acoustic scene classification. In: Proceedings of the Detection and Classification of Acoustic Scenes and Events 2022 Workshop (2022)
21. Seo, Y., Shin, K.S.: Hierarchical convolutional neural networks for fashion image classification. *Expert Syst. Appl.* **116**, 328–339 (2019)
22. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. *Data Min. Knowl. Disc.* **22**(1), 31–72 (2011)
23. Wood, L., Tan, Z., Stenbit, I., Bischof, J., Zhu, S., Chollet, F., et al.: Kerascv. <https://github.com/keras-team/keras-cv> (2022)
24. Xu, L., Ouyang, W., Bennamoun, M., Boussaid, F., Xu, D.: Multi-class token transformer for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference, pp. 4310–4319 (2022)
25. Yan, Z., et al.: HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In: Proceedings of the IEEE ICCV Conference (2015)
26. Zhang, Z., Zhang, H., Zhao, L., Chen, T., Arik, S.Ö., Pfister, T.: Nested hierarchical transformer: towards accurate, data-efficient and interpretable visual understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 3417–3425 (2022)
27. Zhu, X., Bain, M.: B-CNN: branch convolutional neural network for hierarchical classification. arXiv preprint [arXiv:1709.09890](https://arxiv.org/abs/1709.09890) (2017)