

LTGC: Long-tail Recognition via Leveraging LLMs-driven Generated Content

Supplementary Material

A1. Experiments Details

A1.1. Dataset

We conduct the experiments on three popular long-tailed benchmarks, including ImageNet-LT [5], Places-LT [5], and iNaturalist 2018 [3]. We present these datasets in detail below.

ImageNet-LT. The ImageNet-LT [5] is the long-tailed version of the dataset ImageNet-2012 [1]. Overall, ImageNet-LT has 115.8K images from 1000 categories with an imbalanced factor $\beta = 1280/5$.

Places-LT is a long-tailed version of the large-scale scene classification dataset Places [13]. It consists of 62.5K images from 365 categories with class cardinality ranging from 5 to 4,980.

iNaturalist 2018. The iNaturalist 2018 [3] is a large-scale dataset for long-tailed visual recognition rooted in real-world scenarios and exhibits a highly imbalanced distribution. It encompasses 437.5K training images and 24.4K validation images spread across 8142 categories. The dataset’s fine-grained nature further intensifies its complexity.

A1.2. More Results on ImageNet-LT and Place-LT

As shown in Tab. A1 and Tab. A2, we present additional experimental results of our proposed LTGC on ImageNet-LT and Place-LT datasets, including results on the ‘Many’ and ‘Medium’ split sets.

Table A1. Comparison with SOTA methods on ImageNet-LT.

Method	Many	Medium	Few	All
CLIP Zero [9]	60.8	59.3	58.6	59.8
CLIP Finetune [9]	74.4	56.9	34.5	60.5
VL-LTR [9]	77.8	67.0	50.8	70.1
LTGC(Ours)	83.0	79.8	70.5	80.6

Table A2. Comparison with SOTA methods on Places-LT.

Method	Many	Medium	Few	All
CLIP Zero [9]	37.5	37.5	40.1	38.0
CLIP Finetune [9]	50.8	38.6	22.7	39.7
VL-LTR [9]	54.2	48.5	42.0	50.1
RAC [6]	48.7	48.3	41.8	47.2
LPT [2]	49.3	52.3	46.9	50.1
LTGC(Ours)	55.8	54.2	52.1	54.1

A1.3. Prompts for LMMs comparison.

For the experiments on LMMs (Sec. ??), it’s worth noting that providing a labels list significantly improves the model’s performance compared to querying without a

list (i.e., directly asking ‘What species is in the image?’). The labels list provides prior information to the model, helping it narrow down the decision space within its extensive knowledge base. Therefore, on ImageNet-LT, we used the query template ‘Please classify this image. Choose from the following classes: Class 1, Class 2, ..., Class Y.’.

However, it is essential to note that the iNaturalist 2018 dataset encompasses a substantial number of classes, totaling 8,141, which is approximately 184 KB of text. Currently, no LLMs (including the most advanced GPT-4) are capable of processing such a lengthy input. Inputting all categories at once would lead to an error due to the limitations in text length. Dividing the categories into multiple inputs would cause the model to forget previous categories, leading to inaccurate judgments based on the most recent inputs. Therefore, for the iNaturalist dataset, we adopted the query template ‘What species is in the image?’ for testing purposes. We then reviewed the entire response, considering the LLMs’ classification as correct if it included any vocabulary matching the Ground Truth class, and as incorrect otherwise.

Table A3. Effectiveness of the Self-reflection module. **NC**: Number-checking. **RC**: Repetition-checking.

Dataset	NC	RC	Top-1 Accuracy
ImageNet-LT	-	-	60.5
	✓	-	71.7
	-	✓	73.2
	✓	✓	80.6
iNaturalist 2018	-	-	71.6
	✓	-	78.4
	-	✓	71.8
	✓	✓	82.5

A2. Additional Ablation Studies

Here, we conduct additional ablation experiments. Unless otherwise noted, we report the top-1 accuracy averaged over three runs on the ImageNet-LT evaluation protocol.

Effectiveness of the iterative evaluation module. To guarantee the accurate representation of the desired classes by the images produced via T2I, we have integrated an iterative evaluation module within our architecture for the progressive refinement of images. To assess the effectiveness of this module, we contrasted it with three distinct image generation strategies: 1) w/o iterative evaluation: the im-

ages are fed directly into our framework’s training process without any preliminary detection or refinement. 2) Detection and exclusion: the CLIP model evaluates the generated images, selectively forwarding only the ones that align closely with the intended class criteria to the training phase. Images that fail to meet the detection threshold are excluded, bypassing the refinement step entirely. As illustrate in Tab. A4, the performance of the two variants is worse than our method. This suggests that our proposed iterative evaluation module incorporating filtering and refinement of the design is more effective.

Table A4. Evaluation on the effectiveness of the iterative evaluation.

Method	ImageNet-LT	iNaturalist 2018
w/o iterative evaluation	55.8	64.9
Detection and exclusion	71.5	77.4
Ours	80.6	82.5

Effectiveness of the Self-reflection module. The self-reflection module consists of two elements: a number check for images and a repetition check. We separately investigated these two checks, and the results are shown in Tab. A3. When only performing the quality check, LLMs generate highly repetitive descriptions, leading to a decrease in textual diversity and, consequently, a decline in image diversity. When only performing the repetitiveness check, the LLMs are posed the two questions mentioned in Section ?? only once. This results in a limited number of generated samples, thereby leading to limited performance gains. As shown in Tab. A3, our proposed method incorporates both checks and consistently outperforms all variants. This attests to (1) the effectiveness of the repetitiveness design, enabling LLMs to generate comprehensive text descriptions, and (2) the efficacy of the self-reflection design, allowing LLMs to increase the number of samples and simulate a more balanced set of classes.

Impact of using different versions of ChatGPT of LLM. Besides, we also test using different versions of ChatGPT with different capabilities, including GPT 3.0, GPT-3.5 Turbo 16K, and the latest GPT-4 Turbo 128K. As shown in Tab. A5, our framework with different versions of GPT used can achieve different results. This shows that the performance of our framework is affected by the version of ChatGPT.

Table A5. Evaluation on using different versions of ChatGPT.

Version	ImageNet-LT	iNaturalist
GPT 3.0	72.0	74.1
GPT 3.5	76.9	79.3
GPT 4.0	80.6	82.5

The effectiveness of different maximum image num-

bers for LLM’s self-reflection module. In our framework, the maximum number of generated and original images for each class y , denoted as K_y . For LLM’s self-reflection module, the K_y is set to 100 for iNaturalist 2018, 300 for ImageNet-LT, and 800 for Place-LT, respectively. We explore alternative caps for the maximum image number and show the findings in Fig. A1. The results indicate that for each dataset when the cap is set below 100, 300, and 800, respectively, there’s a noticeable improvement in our framework’s performance as the limit on generative images increases. This might be because, by setting the maximum number of images to be a larger number, LLM can better cover the diverse distinctive features and backgrounds (scenes). Moreover, it’s observed that increasing the maximum image number beyond 100, 300, and 800 does not lead to further improvements in performance. However, surpassing these maximum numbers of 100, 300, and 800 doesn’t yield further performance gains. Therefore, balancing performance and efficiency, we determined these respective limits as the optimal settings for our framework.

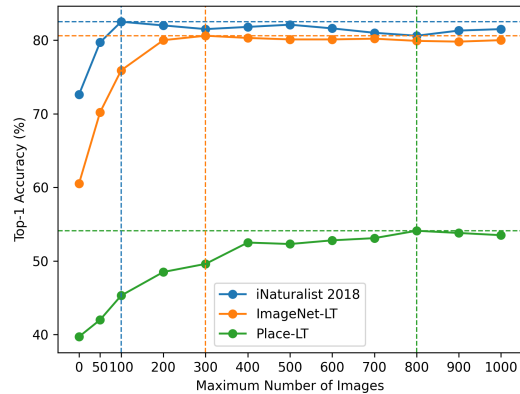


Figure A1. The effectiveness of different maximum image numbers for LLM’s self-reflection module.

Impact of the maximum number of cycle times for iterative evaluation module. In our framework’s iterative evaluation module, we initially cap the cycle count at three. To explore the impact of varying this limit, we experimented with different maximum cycle thresholds and presented the outcomes in Table A6. The results indicate an enhancement in our framework’s performance as we expand the maximum cycle count for iterative evaluation. This improvement is likely due to the more thorough refinement of generated images achieved with a higher cycle limit. However, the gains in performance diminish when the cycle count exceeds three, leading us to establish three as the optimal maximum for the iterative evaluation module.

Check the reasonability of the generated descriptions. In our framework, we use ChatGPT to generate text descriptions for the evaluated classes, and we further design

Table A6. Evaluation on the maximum number of cycle times for iterative evaluation.

Maximum number of cycle time	ImageNet-LT
1	71.5
2	78.1
3	80.6
4	79.9

a iterative evaluation module in which we guide ChatGPT to modify its generated descriptions. Here, we perform a check w.r.t. the reasonability of the generated descriptions before and after passing into the iterative evaluation module. Specifically, we find that, before passing into the iterative evaluation module, 4% of descriptions are checked to be unreasonable, but 0.2% of the descriptions output by the module is checked to be unreasonable. This shows that ChatGPT has small probability of generating unreasonable descriptions, while our iterative evaluation module can further mitigate this problem. The above check is done by inviting 3 volunteers and passing the same 1000 descriptions to them. The 3 volunteers first make decisions independently and then discuss disagreed decisions.

Preparation time. In our framework, before entering the fine-tuning process, we first prepare the generated images by extending the descriptions of tail classes and then generate diverse images for each class, filtering and regenerating the low-quality images. We here report the preparation time of our framework. For generating images, the speed of image generation is limited by DALL-E’s API, about 50 images/minute [8]. For generating descriptions, the speed is about 300 items/minute. Note that the entire preparation process of our framework can be automated by a script with multi-threaded accelerated generation.

The effectiveness of fine-tuning with LoRA. In the fine-tuning process, we fine-tune the visual encoder of the CLIP model using Low-Rank Adaptation (LoRA) [4], LoRA layers are typically added to the Transformer layers of the CLIP model. The main idea of LoRA is to make low-rank modifications to existing weight matrices, rather than updating all parameters directly. This method is often used in large-scale models to achieve effective and parameter-efficient fine-tuning.

Each Transformer layer in the CLIP model consists of two main components: the Multi-Head Self-Attention and the Feedforward Neural Network. LoRA can be applied to the weight matrices of both these components.

Specifically, in the Multi-Head Self-Attention, LoRA can be applied to the Query, Key, and Value matrices. For the Feedforward Network, LoRA can be integrated into the first linear transformation, which is right before the ReLU activation function.

Moreover, other competitive methods of parameter fine-

tuning have emerged in recent years, such as VQT [11]. In Tab. A7 we compare LoRA with other fine-tuning methods such as Linear-probing [7] and VQT [11]. We observe that LoRA outperforms other fine-tuning methods, demonstrating the effectiveness of our fine-tuning with LoRA

Table A7. The effectiveness of fine-tuning with LoRA.

Method	ImageNet-LT	iNaturalist
Linear-probing [7]	71.4	66.3
VQT [11]	77.3	78.1
LoRA (Ours)	80.6	82.5

A3. Additional Visualizations

Visualisation of the iterative evaluation module. In this work, we propose an iterative evaluation module for our framework. This module aims to enhance the quality of generated images, particularly those that are not highly accurate. Fig A2 illustrates the refinement process carried out by our iterative evaluation module. As depicted, this module effectively improves images that initially do not align well with their intended classes, ensuring they more accurately represent these classes. This highlights the efficiency and utility of our iterative evaluation module in image refinement.




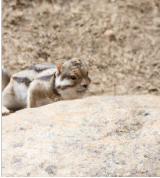


Domain gap between generated images and original images. Some previous methods have shown that there is a domain gap [10] between the generated data and the original data, to better illustrate the domain gap we visualize the t-SNE [12] results for the part of the original data and the part of generated data. In Fig. A3, we observe that there is a domain gap between the generated data and the original data. This motivated us to propose the BalanceMix method and significantly helped the model fine-tuning to improve the recognition results.

A4. Licenses



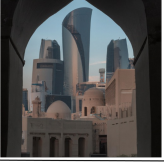





Large model licenses. We use ChatGPT and DALL-E by following the terms of using the services of OpenAI. We use CLIP by following the MIT License.

References

- [1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1
- [2] Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. Lpt: Long-tailed prompt tuning for image classification. *arXiv preprint arXiv:2210.01033*, 2022. 1
- [3] Grant Van Horn, Oisín Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2017 dataset. *CoRR*, abs/1707.06642, 2017. 1

Desired class:	Swimming Pool		Tamias Amoenus	
	Description	Image	Description	Image
Dataset Images	-		-	
The first cycle of iterative evaluation module	A photo of the class Swimming Pool, with a kids' pool area, featuring slides and playful water fountains.		A photo of the class Tamias Amoenus, showcasing its small, agile body, scampering across a rocky terrain.	
The second cycle of iterative evaluation module	A photo of the class Swimming Pool, showcasing a family-friendly design with a shallow kids' pool area, colorful slides, and interactive water fountains, all within a larger swimming complex setting.		A photo of the class Tamias Amoenus, highlighting its distinctive fur pattern with dark and light stripes, agilely navigating through a rocky mountainous habitat typical of its natural environment.	

(a)

Desired class:	Triumphal Arch		Lepus Townsendii	
	Description	Image	Description	Image
Dataset Images	-		-	
The first cycle of iterative evaluation module	A photo of the class Triumphal Arch, capturing the contrast between its ancient design and the modern cityscape surrounding it.		A photo of the class Lepus Townsendii, illustrating its nocturnal activity, foraging in the dim light of dusk on a grassy plain.	
The second cycle of iterative evaluation module	A photo of the class Triumphal Arch, showcasing its imposing and detailed neoclassical architecture, prominently positioned against the backdrop of a modern cityscape, illustrating the historical significance and enduring legacy of these structures amidst contemporary urban development.		A photo of the class Lepus Townsendii, capturing its large, powerful hind legs and oversized ears, which are key adaptations for its fast and agile movement, as it forages in the twilight hours on a vast, open grassland.	
The third cycle of iterative evaluation module	A photo of the class Triumphal Arch, highlighting its grand scale and elaborate sculptural details, with a focus on the ornate reliefs and inscriptions that commemorate historical events, standing proudly as a timeless monument in a historical city square.		A photo of the class Lepus Townsendii, highlighting its unique white-tipped tail and brown-grey fur, along with its large, alert ears, captured foraging in the dim light of dusk on a vast, open grassy plain.	

(b)

Figure A2. Visualization about the iterative evaluation module. The images in (a) do not align most closely with their intended classes until the third cycle, whereas the images in (b) achieve their closest alignment with the intended classes in the second cycle. The classes of *Tamias Amoenus* and *Lepus Townsendii* are derived from the iNaturalist dataset, while *Swimming Pool* and *Triumphal Arch* are from the Place-LT and ImageNet-LT datasets, respectively.

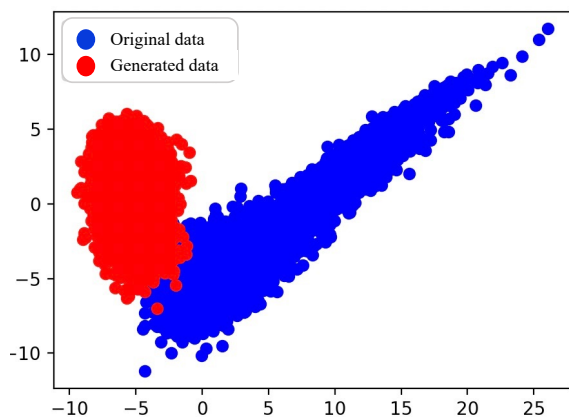


Figure A3. **Domain gap between generated images and original images.**

- [4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3
- [5] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 1
- [6] Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6959–6969, 2022. 1
- [7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3
- [8] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3
- [9] Changyao Tian, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. VI-ltr: Learning class-wise visual-linguistic representation for long-tailed visual recognition. In *European Conference on Computer Vision*, pages 73–91. Springer, 2022. 1
- [10] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023. 3
- [11] Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7725–7735, 2023. 3
- [12] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 3
- [13] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 1