Towards Continual Expansion of Data Coverage: Automatic Text-guided Edge-case Synthesis

Kyeongryeol Go
Superb AI
Seoul, South Korea
krgo@superb-ai.com

Abstract

The performance of deep neural networks is strongly influenced by the quality of their training data. However, mitigating dataset bias by manually curating challenging edge cases remains a major bottleneck. To address this, we propose an automated pipeline for text-guided edge-case synthesis. Our approach employs a Large Language Model, fine-tuned via preference learning, to rephrase image captions into diverse textual prompts that steer a Text-to-Image model toward generating difficult visual scenarios. Evaluated on the FishEye8K object detection benchmark, our method achieves superior robustness, surpassing both naive augmentation and manually engineered prompts. This work establishes a scalable framework that shifts data curation from manual effort to automated, targeted synthesis, offering a promising direction for developing more reliable and continuously improving AI systems. Code is available at github repository.

1. Introduction

The performance of deep neural networks is fundamentally dependent on the quality and diversity of training data. However, real-world datasets are often fraught with noisy labels and inherent biases, which can significantly impair model generalization and robustness [11]. To mitigate these issues, prior research has largely pursued two complementary directions: model-centric methods [9, 18, 19, 22, 34], such as robust optimization and regularization techniques, and data-centric methods [1, 17, 26], including data augmentation and re-sampling strategies.

These methods help mitigate deficiencies in existing datasets. However, they remain largely remedial in nature. A growing trend instead focuses on synthesizing new data to capture model weaknesses and underrepresented edge-case scenarios. Traditionally, identifying such edge-cases has been a labor-intensive process. It typically requires domain experts to manually analyze model failure modes, scrutinizing confusion matrices, class-wise performance metrics, and specific false positives and negatives, to guide subsequent data collection [15]. This manual intervention renders the process not only time-consuming and expertise-dependent but also difficult to automate, scale, and reproduce.

To overcome these limitations, we introduce a novel, automated pipeline for targeted edge-case synthesis. Our pipeline employs a Large Language Model (LLM) fine-tuned with preference learning. This LLM is trained to systematically rephrase image captions from an existing dataset into textual descriptions that characterize challenging edge-case scenarios. These newly generated captions are then fed as prompts to a pre-trained text-to-image (T2I) model, yielding a curated set of diverse and challenging training examples designed to bolster model robustness.

A key distinction of our work lies in its approach to identifying edge-cases at the caption level, rather than through the analysis of image or instance embeddings as in prior work [5, 10, 33]. This design choice is motivated by two key insights. First, we find that simple semantic modifications to captions can induce substantial and meaningful diversity in the generated images, affecting not only object attributes and arrangements but also the broader scene context and background. Second, by building upon the standard T2I generation framework, our pipeline is poised to readily incorporate future advancements in high-fidelity generative models and parameter-efficient fine-tuning techniques.

By automating the discovery and synthesis of edge-cases through generative captioning, our work provides a scalable and efficient solution for improving model performance on challenging, real-world scenarios. Our main contributions are summarized as follows:

- We propose a novel pipeline that automates the generation of targeted edge-case data by leveraging a preference-tuned LLM to create challenging prompts for a T2I model.
- We demonstrate that training with our synthesized data enhances model robustness and generalization on critical edge-case scenarios compared to existing methods.

Figure 1. Comparison of synthetic data generation strategies. Each row displays examples generated from the same real image base caption using three different approaches: naive, manual, and automatic (ours). This row-wise arrangement isolates the effect of the generation strategy on the resulting visual output and its associated labels. Note that pseudo annotations are marked with different colors indicating different classes (Bus, Bike, Car, Pedestrian, Truck). Best viewed in color.

2. Related work

Synthetic data from generative models. Recent advancements in generative models, particularly diffusion-based approaches, have led to a remarkable improvement in the quality of synthetic data generation. This progress has spurred active research into leveraging these models for data augmentation, with generated data enhancing the performance of various visual recognition tasks such as classification [3], detection [4], and segmentation [23].

While generative models offer a promising solution to imperfect data, their application is still in challenges. A primary concern is their tendency to generate data centered around the main modes of the training distribution [31], a phenomenon often referred to as mode-collapse. This issue is particularly acute when dealing with real-world datasets that inherently exhibit a long-tail distribution, where over-reliance on typical samples can be detrimental to the model's ability to handle infrequent but critical cases [32]. In essence, beyond a certain threshold, the utility of synthetic data diminishes as it fails to capture the diversity of rare, edge-case scenarios [6]. Therefore, to effectively enhance the robustness and accuracy of discriminative models, a more strategic approach is required—one that progressively expands the data distribution coverage, with a specific focus on generating synthetic samples in regions where the model exhibits high uncertainty.

Automated synthesis of challenging data. A significant body of research has focused on automatically generating challenging or out-of-distribution (OOD) data to improve model generalization. Early approaches, such as DREAM-OOD [5], learn a compact, text-conditioned latent space from in-distribution (ID) data. Outliers are then synthesized by sampling new embeddings from the low-likelihood regions of this learned manifold and then decoding them into

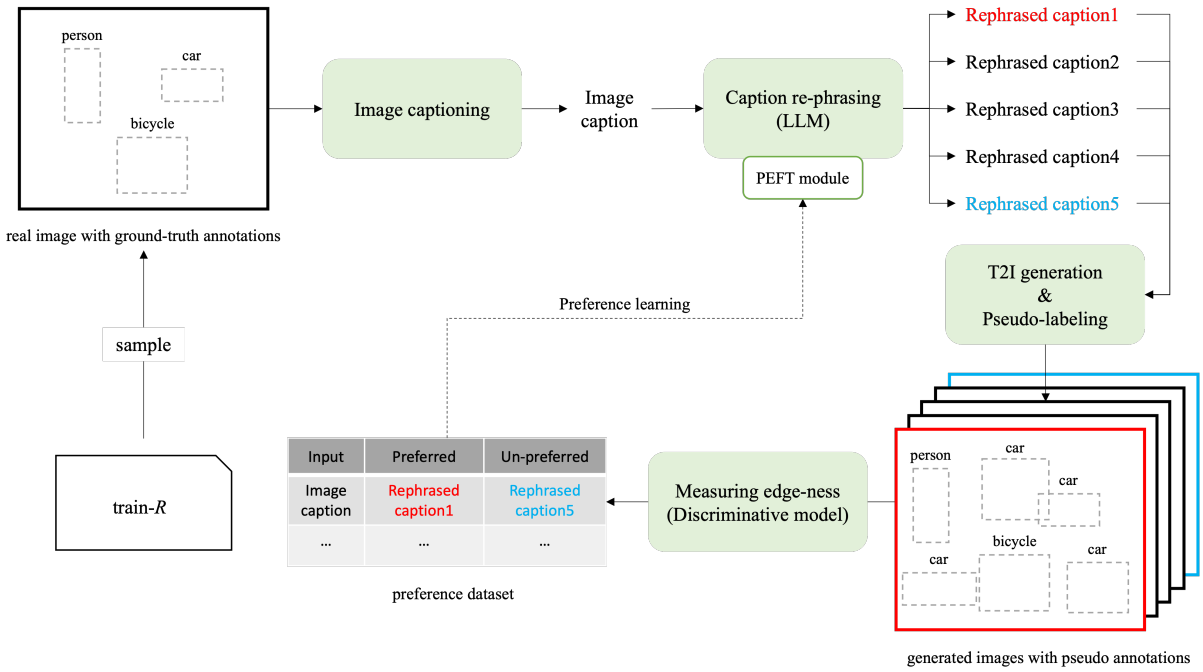


Figure 2. Training pipeline of the rephrasing LLM to understand and generate edge-case-aware captions. This cyclical process teaches the LLM to generate prompts that are more likely to produce challenging images for the discriminative model. Best viewed in color.

photorealistic images. This method targets the boundaries of the known data distribution to produce novel samples.

Subsequent research shifted from generating generic outliers to creating more informative samples specifically tailored for model training. The Guided Imagination Framework (GIF) [33], for instance, expands small-scale datasets by performing optimized perturbations in the latent space of a pre-trained generative model. The optimization is guided by dual objectives: maximizing the information content of samples (e.g., by increasing classification entropy) while preserving their original class identity and promoting sample diversity. More recently, Longtail-Guided Diffusion (LTG) [10] proposed to condition the generation process directly on a target model's blind spots. LTG introduces a differentiable, model-based longtail signal via an Epistemic Head, which guides a latent diffusion model during inference to synthesize instances where the model is uncertain.

Despite their ingenuity, these methods often rely on navigating complex latent spaces and can be constrained by restrictive structural assumptions. Their performance is frequently sensitive to hyperparameter tuning, and the interpretability of the generated data can be limited, posing challenges for scalability and practical deployment.

Data-centric and prompt-based generation. In parallel with automated, model-centric techniques, another line of work has demonstrated the efficacy of a more hands-on, data-centric pipeline, particularly in specialized domains. A notable example is found in fisheye object detection [15], where researchers conducted a detailed manual error analysis to identify critical edge-cases, such as confusing class pairs or objects distorted at the image periphery. These qualitative insights were then translated into systematically crafted text prompts, which guided a fine-tuned T2I model to synthesize images replicating these specific failure modes.

While this approach has proven highly effective for targeted problem-solving, its core limitation is its dependence on domain expertise and manual prompt engineering. This reliance makes the process difficult to scale and adapt to new domains or models without significant human intervention.

Our work aim to combine the scalability and automation of model-conditioned guidance with the semantic precision of targeted, prompt-based generation. Instead of navigating latent embeddings or requiring manual prompt crafting, we propose an automated method to discover and formulate edge-cases at the caption level, thereby creating a scalable yet highly targeted data synthesis pipeline.

3. Methodology

The overall pipeline is illustrated in Figure 2. We aim to automatically synthesize high-value, edge-case data, thereby progressively expanding the coverage and robustness of a training dataset. The core of our methodology is a self-improving feedback loop wherein a LLM is fine-tuned to

rephrase image captions, guiding a generative model toward producing images that are challenging for a given discriminative model. This process leverages preference learning to align the LLM's rephrasing strategy with a quantitative measure of edge-ness.

3.1. Initial data preparation

The process commences with an image x from the original training dataset D. First, a pre-trained image captioning model, $Cap(\cdot)$, generates a concise descriptive caption, $c_{base} = Cap(x)$. This base caption serves as a factual, high-level summary of the image content.

Next, to explore the semantic space around this description, c_{base} is fed into a rephrasing LLM, denoted as π_θ, where θ are its parameters. The LLM is prompted to produce a set of N diverse semantic variations, $C = \{c_1, c_2, ..., c_N\}$, where each $c_i \sim \pi_\theta(\cdot|c_{base})$. These rephrased captions are designed to alter contextual details, object attributes, or narrative perspectives of the original scene, forming the foundation for generating a diverse range of visual outputs. While increasing N enhances the diversity of the rephrased captions, it also incurs additional computational cost, as each caption must be synthesized and pseudo-labeled in the following steps. For practicality, we fix $N = 5$.

3.2. Quantifying edge-ness for preference labeling

Each of the N rephrased captions $c_i \in C$ is passed to a pre-trained T2I model, $G(\cdot)$, to synthesize a corresponding image, $x'_i = G(c_i)$. Since these synthetic images lack ground-truth annotations, we employ a high-performance, pre-trained model as a pseudo-labeler, $PL(\cdot)$, to generate pseudo-annotations $y'_i = PL(x'_i)$.

To evaluate which rephrased captions are most effective at generating challenging scenarios, we introduce the concept of edge-ness. We define edge-ness as a quantifiable measure of the difficulty a synthetic image poses to a discriminative model, M_ϕ, which is pre-trained on the original training dataset D. Specifically, for each generated pair (x'_i, y'_i), we compute the task-specific training loss:

$$s_i = \mathcal{L}_{task}\left(M_\phi(x'_i), y'_i\right) \tag{1}$$

A higher value of s_i is interpreted as an indicator of edge-ness, as it signifies a greater discrepancy between the model's predictions and the pseudo annotations.

3.3. Edge-case-aware rephrasing

To align the LLM with the objective of generating challenging scenarios, we construct a preference dataset D_{pref} consisting of pairs (c_w, c_l), where c_w denotes the caption that induced the maximum task loss (preferred) and c_l denotes the caption that induced the minimum task loss (un-preferred) among the N variations generated for a given base caption. This dataset is used to fine-tune the rephrasing LLM π_θ, effectively aligning its behavior toward generating captions that are more likely to produce high edge-ness images. For this preference-based fine-tuning, we adopt Direct Preference Optimization (DPO) [28], which optimizes the model to increase the relative log-probability of c_w over c_l using the following objective:

$$\begin{aligned} \mathcal{L}_{DPO} &= \log\sigma\left(\beta\big(r_\theta(c_w \mid c_{base}) - r_\theta(c_l \mid c_{base})\big)\right), \\ \text{where} \quad r_\theta(c \mid c_{base}) &= \log\frac{\pi_\theta(c \mid c_{base})}{\pi_{ref}(c \mid c_{base})}. \end{aligned} \tag{2}$$

Here, the reference model π_{ref} is a frozen copy of the initial language model, σ denotes the sigmoid function, and β controls the strength of the preference signal.

Once the LLM is aligned, the newly fine-tuned model $\pi_{\theta'}$ is used to generate a set of edge-case-aware captions. These captions are used to synthesize new images, which, along with their pseudo annotations, are used to augment the original training set: $D_{aug} = D \cup \{(x'_{new}, y'_{new})\}$. Please refer to Appendix A for the data augmentation pipeline.

Notably, this entire pipeline can be executed iteratively. By retraining the discriminative model M_ϕ on D_{aug} and using the updated model as the next-stage edge-ness scorer, the system can progressively discover and synthesize increasingly complex edge-cases, continually expanding the data coverage.

4. Experiment

We aim to validate whether our proposed pipeline effectively generates edge-case data that addresses the blind spots of a given discriminative model. We primarily compare the efficacy of different prompt generation strategies for augmenting a training dataset via the T2I model. Due to space constraint, details on model architectures and training configurations are provided in Appendix B.

4.1. Setup

Comparison groups. To evaluate the effectiveness of our proposed method as a prompt generation strategy, we compare it against the two baselines. For clarity, we present both the baselines and our method below:

- **naive**: This baseline directly uses the base captions (c_{base}) generated by InternVL3-38B [35] from the training images. It generates new images without any further rephrasing or modification to these captions, varying only the random seed of the T2I model for diversity.
- **manual**: Following prior work [15], this baseline relies on manually-engineered prompts. These prompts are derived from detailed error analyses by AI researchers and are designed to elicit specific, known failure modes.

Table 1. YOLOv11-small performance under iterative data augmentation strategies, comparing naive, manual, and automatic pipelines, with evaluation using mAP and mAP w/o TP.

Train dataset	Count	mAP	mAP w/o TP
train-*D*	3,187	0.335 ± 0.005	-
train-*D* + naive v0	19,122	0.376 ± 0.004	-
train-*D* + naive v0 + naive v1	35,057	0.378 ± 0.002	0.361 ± 0.002
train-*D* + naive v0 + manual v1	35,057	0.374 ± 0.003	0.357 ± 0.003
train-*D* + naive v0 + automatic v1	35,057	0.381 ± 0.003	0.363 ± 0.003
train-*D* + naive v0 + naive v1 + naive v2	50,992	0.380 ± 0.002	0.362 ± 0.003
train-*D* + naive v0 + manual v1 + manual v2	50,992	0.380 ± 0.001	0.362 ± 0.001
train-*D* + naive v0 + automatic v1 + automatic v2	50,992	**0.384 ± 0.001**	**0.366 ± 0.001**

Table 2. Performance comparison for cross-scale transferability of synthetic data (small → medium/large).

(a) YOLOv11-medium performance.

Train dataset	Count	mAP	mAP w/o TP
train-*D*	3,187	0.366 ± 0.004	-
train-*D* + naive v0	19,122	0.426 ± 0.002	-
train-*D* + naive v0 + naive v1	35,057	0.429 ± 0.004	0.415 ± 0.005
train-*D* + naive v0 + manual v1	35,057	0.428 ± 0.005	0.414 ± 0.005
train-*D* + naive v0 + automatic v1	35,057	**0.429 ± 0.003**	**0.416 ± 0.005**

(b) YOLOv11-large performance.

Train dataset	Count	mAP	mAP w/o TP
train-*D*	3,187	0.371 ± 0.001	-
train-*D* + naive v0	19,122	0.419 ± 0.001	-
train-*D* + naive v0 + naive v1	35,057	0.431 ± 0.003	0.414 ± 0.004
train-*D* + naive v0 + manual v1	35,057	0.429 ± 0.007	0.411 ± 0.008
train-*D* + naive v0 + automatic v1	35,057	**0.432 ± 0.005**	**0.415 ± 0.006**

- **automatic**: Our method employs a preference-tuned LLM to automatically rephrase captions into descriptions of edge-case scenarios, removing the need for manual intervention.

Dataset. We conduct our experiments on the FishEye8K dataset [8], a benchmark adopted to the AI-City Challenge. The dataset consists of images from 20 fisheye cameras with annotations for five categories: Bus, Bike, Car, Pedestrian, and Truck. Given its inherent scene biases across diverse urban environments, FishEye8K serves as an optimal testbed for evaluating the effectiveness of our automated edge-case synthesis.

To rigorously assess model robustness, we partition the official training set into two disjoint subsets: train-*D* and train-*R*. To simulate a restricted training environment, train-*D* is used for training the discriminative model while train-*R* is used for the preference learning of the rephrasing LLM and contains a more diverse distribution. This setup allows the LLM to learn under-represented characteristics and generate corresponding synthetic data to fill the gaps in train-*D*. Please refer to Appendix C for more details.

To prevent potential data leakage, the LLM is restricted from accessing train-*D* during the preference learning phase and only interacts with it at inference time for data augmentation. This protocol ensures that the synthesized samples reflect the LLM's generalization capability rather than simple memorization. For completeness, the test set is held out exclusively for final evaluation.

Evaluation metrics. We evaluate the effectiveness of the prompt generation strategies using the mAP gain of the discriminative model after augmenting the dataset with synthesized samples. This evaluation protocol allows us to directly quantify how much each augmentation strategy contributes to improving detection performance beyond the original training data. To explicitly assess how well each method addresses the model's blind spots, we introduce a custom

metric, termed *mAP w/o TP*, described in Appendix D. In brief, this metric computes mAP only on ground-truth annotations that do not overlap with the True Positive predictions of the discriminative model trained on train-*D* and naive v0, thereby focusing exclusively on previously misrecognized or challenging instances. A higher mAP w/o TP therefore indicates that the synthetic data more effectively improves the model's initial blind spots, rather than merely reinforcing patterns that the model has already learned.

4.2. Visual diversity

Figure 1 provides a controlled comparison of how different prompt generation strategies influence synthetic visual outputs. By utilizing the same base caption for each row, we can directly observe the impact of rephrasing on synthesis.

While the automatic approach does not necessarily generate a higher number of object instances than the manual variant, it exhibits significantly greater background diversity, encompassing a wider range of lighting conditions, camera viewpoints, and scene contexts. This suggests that the preference-tuned LLM successfully explores a broader distribution of plausible scenarios rather than sticking to the main modes of the training distribution. In contrast, the manual baseline often produces repetitive scenes with limited background variation. Consequently, the automatic method is more effective at generating novel samples that cover underrepresented regions of the data distribution, thereby helping to alleviate dataset bias and improve model robustness

4.3. Model performance

As summarized in Table 1, our proposed method consistently outperforms both baselines across all experimental settings. To establish a competitive baseline, we define v0 as the initial iteration of data augmentation, where the train-*D* is expanded five-fold using the naive strategy. This naive v0 dataset serves as the common starting point for all subsequent augmentation experiments. When augmenting the dataset further in the next iteration (v1), our method achieves the highest mAP and, more importantly, the highest mAP w/o TP.[1]

This result demonstrates that the data generated by our pipeline is more effective at correcting the baseline model's specific weaknesses compared to both naively generated data and data from manually engineered prompts. In contrast, other baselines show diminishing returns, suggesting that simply increasing data volume without targeting specific weaknesses is a less efficient strategy. For a class-wise and tag-wise analysis of the model performance to inspect the effect on dataset bias, please refer to Appendix E and F, respectively.

[1]Note that the model trained on train-*D* + naive v0 serves as the reference for calculating the mAP w/o TP. Accordingly, we measure this metric exclusively for subsequent iterations of data augmentations.

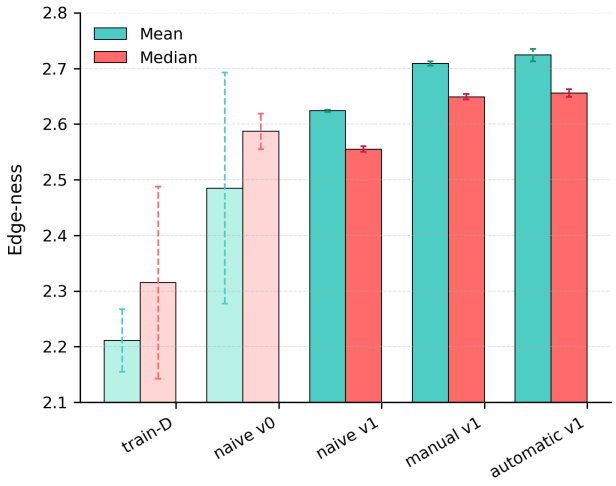


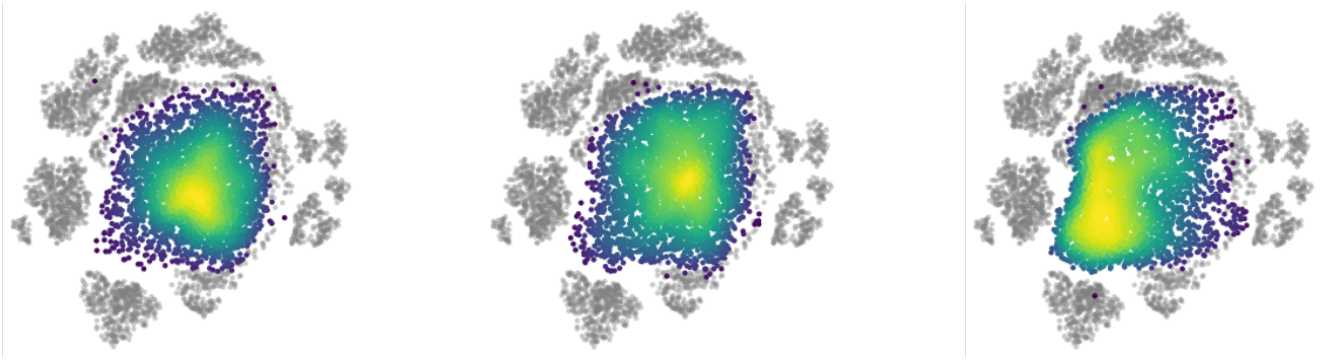
Figure 3. Comparison of edge-ness statistics across different data augmentation strategies. The proposed automatic pipeline produces samples with the highest mean and median loss, indicating more challenging training examples for the discriminative model.

4.4. Transferability

To assess whether a dataset deemed effective for one model remains valid for another, a core concern in data-centric evaluation, we conducted transferability experiments. This approach aligns with recent findings [7, 12] that emphasize robust evaluation designs, acknowledging that instance informativeness may vary across model configurations and necessitating experimental methods to identify broadly transferable data. To examine the transferability of synthetic data across model scales, we trained YOLOv11-medium and YOLOv11-large using datasets generated by the proposed pipeline with YOLOv11-small. As summarized in Table 2, the automatic still achieves the highest mAP and mAP w/o TP.

4.5. Edge-ness

We empirically validate the core mechanism of our pipeline: its ability to generate images with high edge-ness. We measure the training loss $\mathcal{L}_{task}$ of the discriminative model on data generated by each method. Following the standard training objective of our YOLOv11-small discriminator, $\mathcal{L}_{task}$ is specifically defined as the weighted sum of the bounding box regression loss, classification loss, and distribution focal loss. [13]. As shown in Figure 3, data from our proposed pipeline consistently induces the highest mean and median loss. This confirms that our preference learning framework successfully steers the LLM to generate captions that translate into verifiably more difficult examples for the discriminative model, thereby creating effective training signals for improving robustness.



(a) naive (b) manual (c) automatic

Figure 4. Comparison of naive, manual, and automatic by UMAP of CLIP embeddings. Gray points represent real data, while the colored ones indicate synthetic data, with colors closer to yellow denoting higher density.

4.6. Embedding

To better understand the underlying properties of the generated data, we visualize the image embeddings obtained by CLIP [27] for each generation strategy using UMAP [20], as shown in Figure 4. This visualization provides an intuitive view of how different strategies explore the feature space induced by a strong vision-language model. Compared to the naive approach, the manual strategy exhibits a slightly broader spread, indicating a modest increase in diversity. In contrast, the automatic method not only achieves a significantly greater spread, but also shifts its mode toward a sparser region of the real data embedding space. This shift suggests that the automatic strategy is more effective at generating novel and less frequently observed samples, potentially covering underrepresented regions of the data distribution. Such behavior is consistent with our objective of synthesizing edge-case examples that complement the existing training data.

4.7. Linguistic analysis

A closer inspection of the prompts and images of high-density regions reveals that our preference-tuned LLM learns a sophisticated rephrasing strategy. Compared to the factual source captions, its outputs tend to: (1) employ more descriptive, literary language; (2) emphasize actions and dynamic situations over static object descriptions; and (3) transform objective metadata (e.g., time, weather) into evocative descriptions of mood and atmosphere. This contrasts sharply with the manual baseline, where manual prompts are narrowly focused on object attributes. We interpret this as an automated strategy to enrich background context and scene diversity, which directly counteracts the inherent biases of the FishEye8K dataset. Please refer to Appendix G for more structured analysis.

5. Conclusion

In this paper, we propose a novel pipeline for automated, text-guided edge-case synthesis. This framework facilitates the creation of more diverse and challenging synthetic datasets, providing a scalable pathway to improve both training efficiency and generalization. We anticipate that the benefits of our pipeline will become even more pronounced as more advanced T2I models and pseudo-labelers emerge. Furthermore, considering dataset bias even during the fine-tuning of T2I models and pseudo-labelers could have improved performance. For future work, we plan to extend our study to a broader range of datasets and explore alternative uncertainty measures for edge-ness.

References

[1] Sumyeong Ahn, Seongyoon Kim, and Se-Young Yun. Mitigating dataset bias by using per-sample gradient. *arXiv preprint arXiv:2205.15704*, 2022. 1

[2] AI@Meta. Llama 3 model card. 2024. 9

[3] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023. 2

[4] Kai Chen, Enze Xie, Zhe Chen, Yibo Wang, Lanqing Hong, Zhenguo Li, and Dit-Yan Yeung. Geodiffusion: Text-prompted geometric control for object detection data generation. *arXiv preprint arXiv:2306.04607*, 2023. 2

[5] Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models. *Advances in Neural Information Processing Systems*, 36:60878–60901, 2023. 1, 2

[6] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7382–7392, 2024. 2

[7] Kyeongryeol Go and Kye-Hyeon Kim. Transferable candidate proposal with bounded uncertainty. In *NeurIPS 2023 Work-*

shop on Adaptive Experimental Design and Active Learning in the Real World, 2023. 6

[8] Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Erkhembayar Ganbold, Jun-Wei Hsieh, Ming-Ching Chang, Ping-Yang Chen, Byambaa Dorj, Hamad Al Jassmi, Ganzorig Batnasan, Fady Alnajjar, et al. Fisheye8k: A benchmark and dataset for fisheye camera object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5305–5313, 2023. 5

[9] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*, 2017. 1

[10] David S Hayden, Mao Ye, Timur Garipov, Gregory P Meyer, Carl Vondrick, Zhao Chen, Yuning Chai, Eric Wolff, and Siddhartha S Srinivasa. Generative data mining with longtail-guided diffusion. *arXiv preprint arXiv:2502.01980*, 2025. 1, 3

[11] Max Hort, Zhenpeng Chen, Jie M Zhang, Mark Harman, and Federica Sarro. Bias mitigation for machine learning classifiers: A comprehensive survey. *ACM Journal on Responsible Computing*, 1(2):1–52, 2024. 1

[12] Yilin Ji, Daniel Kaestner, Oliver Wirth, and Christian Wressnegger. Randomness is the root of all evil: more reliable evaluation of deep active learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3943–3952, 2023. 6

[13] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. 6, 9

[14] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. 10, 12

[15] Seunghyeon Kim and Kyeongryeol Go. Edge-case synthesis for fisheye object detection: A data-centric perspective. *arXiv preprint arXiv:2507.16254*, 2025. 1, 3, 4, 9, 10

[16] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024. 9

[17] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *Advances in Neural Information Processing Systems*, 34:25123–25133, 2021. 1

[18] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 1

[19] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020. 1

[20] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. 7

[21] MMDetection Contributors. OpenMMLab Detection Toolbox and Benchmark, 2018. 10

[22] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684, 2020. 1

[23] Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. *Advances in Neural Information Processing Systems*, 36:76872–76892, 2023. 2

[24] OpenAI. Gpt-4.1, 2024. Accessed: 2025-07-05. 9

[25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 11

[26] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020. 1

[27] 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. 7, 11

[28] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023. 4

[29] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformer Reinforcement Learning. 9

[30] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics, 2020. 9

[31] Shin'ya Yamaguchi and Takuma Fukuda. On the limitation of diffusion models for synthesizing training datasets. *arXiv preprint arXiv:2311.13090*, 2023. 2

[32] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45 (9):10795–10816, 2023. 2

[33] Yifan Zhang, Daquan Zhou, Bryan Hooi, Kai Wang, and Jiashi Feng. Expanding small-scale datasets with guided imagination. *Advances in neural information processing systems*, 36:76558–76618, 2023. 1, 3

[34] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 1

[35] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 4, 9

[36] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023. 9

A. Data augmentation pipeline

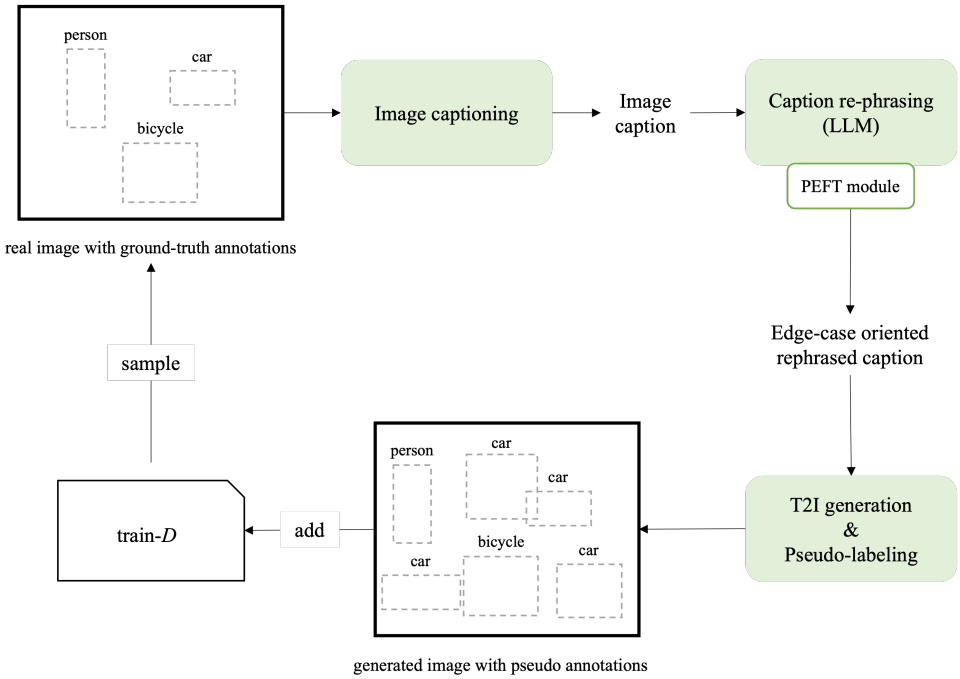


Figure 5. Data augmentation pipeline by inference of the preference-tuned LLM.

This diagram depicts the data augmentation process using the preference-tuned LLM obtained from the training phase. An image is sampled from the train-*D* split and a base caption is generated. The preference-tuned LLM then produces the optimized "edge-case oriented" captions based on its learned preference. This targeted captions are used to synthesize the new, high-value images via the T2I model. After being assigned pseudo-annotations by a pseudo-labeler, this new synthetic data is added to the train-*D* set, thereby augmenting the original training data with a sample specifically designed to address a model weakness.

B. Model architectures and training configurations

We summarize the details of the model architectures and training procedures employed in our paper.

Models. We used five distinct models covering captioning, re-phrasing, generation, pseudo-labeling, and discriminative tasks:

- Image captioning: InternVL3-38B [35]
- Caption re-phrasing: Llama-3-8B-Instruct [2]
- Text-to-image generation: Flux.1-dev [16]
- Pseudo-labeling: Co-DETR [36]
- Discriminative: YOLOv11-small [13]

It is worth noting that the model selection closely aligns with the manual baseline [15], except for the caption re-phrasing component. Unlike the baseline, which relied on the closed-source GPT-4.1-mini [24], we employed an open-source model to allow fine-tuning for our task.

Training strategies. The image captioning model (InternVL3-38B) was directly adopted from its publicly available pre-trained checkpoint in `transformers` [30] without further fine-tuning. The caption re-phrasing model (Llama-3-8B-Instruct) was fine-tuned with the preference dataset constructed from train-*R* using the `trl` [29] with a Low-Rank Adaptation (LoRA) scheme. The text-to-image generation model (Flux.1-dev) was trained with train-*D* via the `simpletuner` library, employing Low-Rank Kronecker product (LoKr) scheme. For the pseudo-labeling stage, we fine-tuned Co-DETR with train-*D* using the

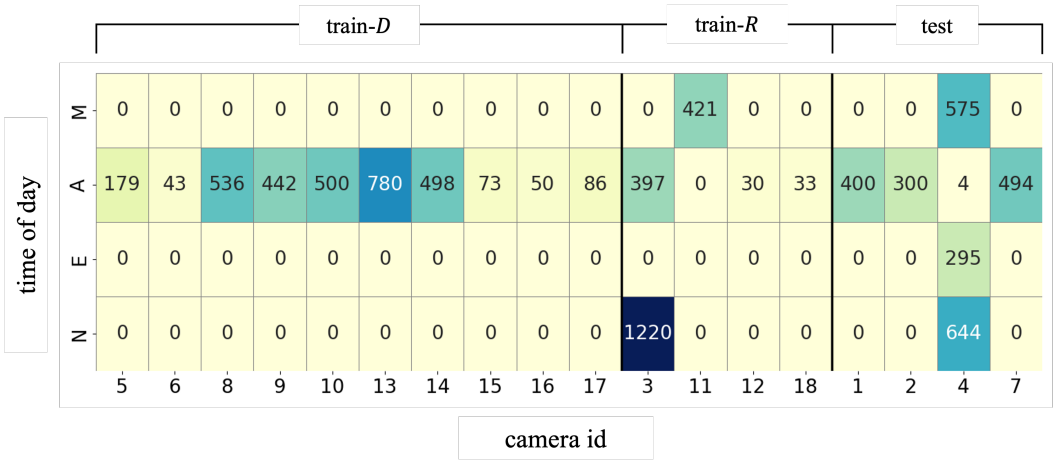


Figure 6. Number of images per camera ID and time-of-day in FishEye8K. The dataset is split into train-*D*, train-*R*, and test sets based on camera IDs, intentionally creating different levels of bias across splits.

`mmdetection` [21] with full model finetuning. Finally, the discriminative model (YOLOv11-small) was trained with train-*D* and synthetic datasets through full finetuning using the `ultralytics` [14].

Configurations. All training was performed by the default configurations provided by each library, unless otherwise specified. We adopt most settings from prior work [15], and release all configuration files, bash scripts, and step-by-step guides for reproducing the experiments at the github repository.

Hardware. All training and inference were performed on four NVIDIA A100 GPUs. The heaviest GPU usage occurred during image captioning with InternVL3-38B, which required approximately 75GiB VRAM per device.

C. Analysis of bias in data splits

We divided the FishEye8K dataset into three subsets based on camera IDs to intentionally introduce dataset biases and to study their impact on discriminative and generative modeling. The splits are as follows:

- train-*D*: camera ids {5, 6, 8, 9, 10, 13, 14, 15, 16, 17}
- train-*R*: camera ids {3, 11, 12, 18}
- test: camera ids {1, 2, 4, 7}

As shown in Figure 6, the distribution of images across time-of-day and camera IDs reveals distinct patterns of bias. Specifically, the train-*D* split contains images exclusively from the afternoon (A), with no images from morning (M), evening (E), or night (N). In contrast, the train-*R* split exhibits a moderate bias with images from multiple times of day, while the test split is relatively unbiased.

This deliberate separation allows us to train a discriminative model on the highly biased train-*D* split, highlighting its limitations when generalizing to other times of day. Meanwhile, the train-*R* split serves to allow the rephrasing LLM to automatically capture under-represented characteristics of train-*D* split and generate such data.

D. Custom evaluation metric proposal

D.1. Background and motivation

In data-centric AI, a critical task is to identify a model's weaknesses and subsequently verify whether new algorithms or data effectively address them. In object detection, these weaknesses often manifest as instance-level failures, such as objects that are occluded, truncated, or small. These *weak instances* are typically rare compared to easily detectable ones. Consequently, global performance metrics like mean Average Precision (mAP) are often insensitive to improvements on these specific blind spots, as the metric's overall fluctuation is dominated by the vast number of *easy* instances. This makes it challenging to determine if a new approach genuinely targets and resolves the model's key deficiencies.

D.2. Limitations of existing analysis methods

Several methods are commonly used to identify model weaknesses, but each has its limitations:

Class-wise AP While useful for identifying underperforming classes, it is not sufficiently informative when a class has large intra-class variation and the model consistently fails on specific patterns within that class.

False positive/negative analysis Visualizing the correction of false positives and false negatives provides qualitative evidence of improvement. However, there is no standard quantitative metric to measure this change systematically across the entire dataset.

Embedding and uncertainty analysis Techniques that identify sparse regions in an embedding space or high-uncertainty samples are often used as proxies for data difficulty. However, these methods may require separate, large-scale models [25, 27]. While effective in general domains, these models often require additional fine-tuning for specialized domains (e.g., manufacturing defects, medical imaging), limiting their direct applicability.

D.3. Proposal: mAP w/o TP

To address these limitations, we propose **mAP w/o TP** (mean Average Precision without True Positives of a baseline model). This metric provides a clear, quantitative way to measure how effectively a new model (Model $\mathcal{B}$) improves upon the specific weaknesses identified by a baseline model (Model $\mathcal{A}$).

The core idea is to first define *non-weak instances* as the set of annotations that the baseline Model $\mathcal{A}$ already detects correctly (its true positives (TP)). Then, when evaluating Model $\mathcal{B}$, we exclude its correct predictions on these non-weak instances from the calculation. This focuses the evaluation squarely on the subset of data that was challenging for the baseline model, making any improvements on these blind spots much more apparent. We provide an example of the filtered ground-truth annotations and predictions in Figure 7.

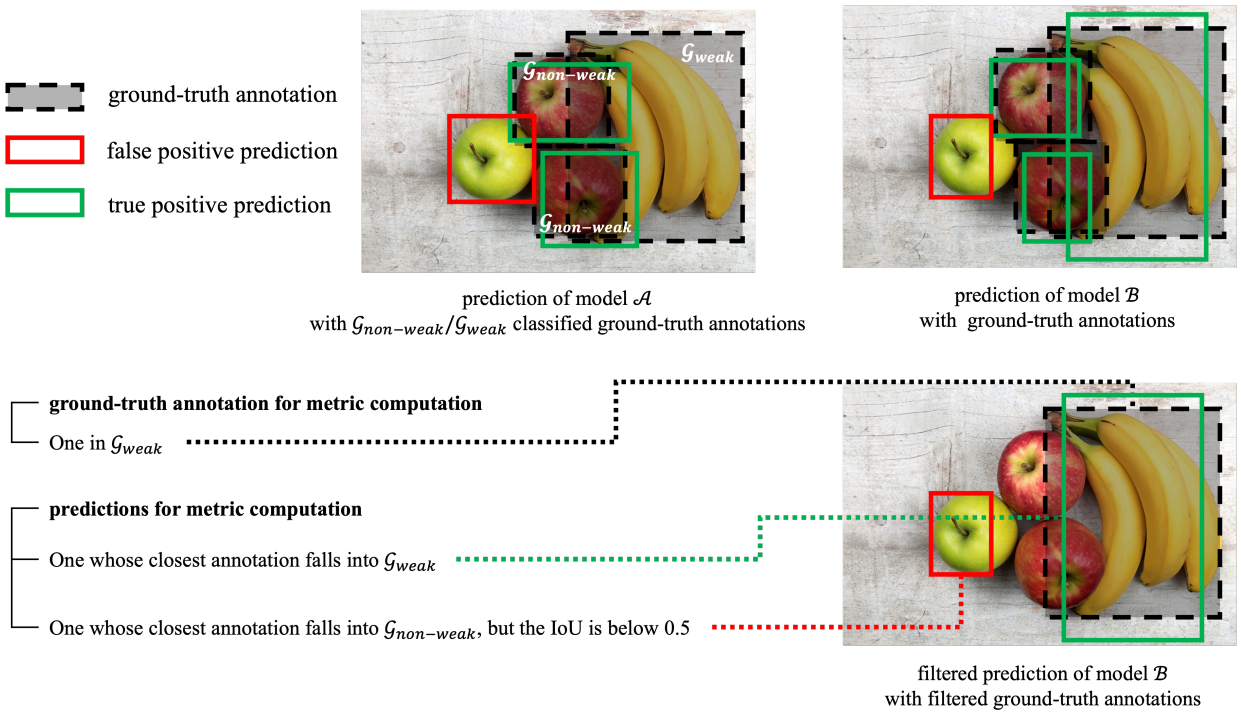


Figure 7. Filtered ground-truth annotations and predictions to compute mAP w/o TP.

D.4. Step-by-Step calculation

The mAP w/o TP for a new Model $\mathcal{B}$ relative to a baseline Model $\mathcal{A}$ is calculated as follows:

1. Inference with baseline model: First, run inference with Model $\mathcal{A}$ on the evaluation dataset to obtain its set of predictions.
2. Partition ground-truth annotations: Categorize all ground-truth annotations in the evaluation set into two disjoint sets based on Model $\mathcal{A}$'s predictions:
   - $\mathcal{G}_{\text{non-weak}}$: The set of annotations correctly identified by a TP prediction from Model $\mathcal{A}$ (based on the IoU greater than 0.95 and class label matching).
   - $\mathcal{G}_{\text{weak}}$: Other annotations, which correspond to Model $\mathcal{A}$'s FN or were not associated with any high-confidence prediction.
3. Inference with new model: Run inference with the new Model $\mathcal{B}$ on the same evaluation dataset to obtain its predictions.
4. Filter predictions of new model: Filter the predictions from Model $\mathcal{B}$ based on their relationship with the partitioned annotation sets. A prediction from Model $\mathcal{B}$ is kept for evaluation only if it meets one of the following criteria:
   - Its highest-IoU corresponding annotation belongs to $\mathcal{G}_{\text{weak}}$ (i.e., it addresses a weak instance).
   - Its highest-IoU corresponding annotation belongs to $\mathcal{G}_{\text{non-weak}}$, but the IoU is less than 0.5. This penalizes cases where Model $\mathcal{B}$ fails on an easy instance that Model $\mathcal{A}$ handled correctly.
5. Calculate final metric: Calculate the mAP, class-wise AP, or other relevant metrics using only the filtered set of predictions from the previous step and ground-truth annotations in $\mathcal{G}_{\text{weak}}$.

D.5. Significance and benefits

The mAP w/o TP metric offers several key advantages:

- It provides a **quantitative and direct** measure of a model's ability to resolve the specific blind spots of a baseline.
- By isolating the evaluation to weak instances, it makes the impact of targeted algorithms or data augmentation strategies **more interpretable**.
- It is **model-agnostic** and universally applicable. It does not depend on a model's architecture or training methodology and can be used to compare any two models.

E. Analysis in a class-wise manner

Table 3. Per-class statistics and evaluation performance.

Setting	Bus	Bike	Car	Pedestrian	Truck
train-D + naive v0 + naive v1	8,904	247,519	**205,875**	60,201	4,064
train-D + naive v0 + manual v1	**11,916**	232,935	202,669	69,967	**7,367**
train-D + naive v0 + automatic v1	9,457	**250,853**	203,388	**86,666**	4,230

(a) Per-class instance counts

Setting	Bus	Bike	Car	Pedestrian	Truck
train-D + naive v0 + naive v1	0.509	0.303	**0.507**	0.189	**0.368**
train-D + naive v0 + manual v1	**0.527**	0.308	0.503	0.218	0.332
train-D + naive v0 + automatic v1	0.510	**0.320**	0.505	**0.221**	0.343

(b) Per-class evaluation performance (mAP)

We report the class-wise instance counts and detection performance of YOLOv11-small [14] in Table 3. The results indicate that the automatic provides more balanced improvements across classes compared to the naive and manual baselines. In particular, the Pedestrian class, originally underrepresented and difficult to detect, shows the largest gap. Similarly, the Bike class, which had a large number of training samples but relatively low accuracy, also benefits from the automatic strategy. In short, the automatic effectively mitigates this trade-off by improving rare and difficult classes (Pedestrian, Bike) while maintaining stable performance on other categories.

In contrast, the naive primarily increases data quantity, which boosts performance for dominant classes such as Car, but fails to consistently improve underrepresented categories like Pedestrian and Bike. This suggests that quantity alone does not guarantee higher performance when the additional data lacks diversity and edge-ness. While the manual improves Pedestrian accuracy, it comes at a cost, markedly reducing the Truck score despite a substantial increase in its instance count.

F. Analysis in a tag-wise manner

To assess the impact on dataset bias, we conduct a tag-wise analysis with respect to the time of day: morning, evening, and night. These categories are particularly critical as they represent an underrepresented tag in the dataset. As shown in Table 4, although naive and manual strategies yield gains in certain cases, the automatic approach consistently achieves equal or superior performance across the time of days. It is worth noting that the manual baseline serves as a strong baseline in this evaluation, given its design for fine-grained error analysis, even though it often underperforms the naive baseline in other quantitative results. In contrast, the automatic not only adapts more effectively to rare conditions but also delivers more robust improvements overall. These findings highlight the automatic strategy as a more effective solution for alleviating dataset bias in underrepresented tags.

Table 4. Performance comparison across time of day.

Setting	M (morning)	E (evening)	N (night)
train-D + naive v0 + naive v1	$\mathbf{0.402 \pm 0.002}$	0.497 ± 0.016	0.177 ± 0.013
train-D + naive v0 + manual v1	0.387 ± 0.014	$\underline{0.502 \pm 0.011}$	$\underline{0.182 \pm 0.020}$
train-D + naive v0 + automatic v1	$\underline{0.398 \pm 0.009}$	$\mathbf{0.509 \pm 0.007}$	$\mathbf{0.183 \pm 0.007}$

G. Analysis of rephrasing patterns

We analyze 20 caption–rephrased pairs located in high-density regions of Figure 4c. This analysis reveals several consistent patterns that highlight the strategies used to enrich the original descriptions and make them more vivid and informative. The six most prominent trends are as follows:

1. *Scene atmosphere enhancement*: Rephrased captions frequently introduce emotive and descriptive adjectives, such as "serene", "peaceful", "bustling", or "vibrant", to convey the mood of the scene more vividly than the original caption. This trend appears in all 20 pairs, indicating its universal role in enhancing expressiveness.
2. *Perspective specification*: Many rephrasings explicitly indicate the camera viewpoint (e.g., "front-view", "side-view", "low-angle"), adding spatial context that is often implicit in the original captions. This occurs in the majority of pairs (15/20), highlighting the importance of visual perspective in rephrasing.
3. *Temporal and weather visualization*: Neutral references to weather or time (e.g., "clear sky", "daytime") are often expanded into more expressive phrases, such as "golden morning light", "sun-drenched afternoon", or "warm glow", enhancing the visual imagery. This trend is consistently applied across all pairs, showing a clear preference for temporal and environmental enrichment.
4. *Intersection type clarification*: Generic mentions of "intersection" are frequently replaced with more precise terms like "T-junction", "Y-junction", or "street corner", providing clearer spatial context. Applied in 14/20 pairs, this trend improves spatial specificity without altering the core scene description.
5. *Action emphasis*: Rephrased captions tend to highlight the dynamics of pedestrians and vehicles, converting simple existence statements into descriptive actions such as "stroll", "zip by", or "weave through", thereby increasing narrative engagement. This occurs in 16/20 pairs, emphasizing the narrative benefit of describing movement.
6. *Urban context enrichment*: Additional details about shops, cafes, buildings, and signage are often added to create a richer urban setting, making the scene more specific and relatable. This trend is applied in 17/20 pairs, reflecting a strong tendency to augment environmental details.

Overall, these trends consistently appear across the majority of caption pairs, suggesting that effective rephrasing not only preserves the factual content but also enhances *expressiveness*, *spatial awareness*, and *narrative dynamics*.

H. Prompt templates

We present the exact prompts used in our experiments. In particular, we list them in sequence: starting with the prompt used by InternVL3-38B to generate captions from real images, followed by those applied with Llama-3-8B-Instruct for caption rephrasing under the manual baseline, for constructing the preference dataset, and with the preference-tuned LLM.

H.1. Image captioning prompt

You are an expert in generating high-quality image captions. Please analyze the provided fish-eye image in detail.

Please adhere to the following format for the caption:
- Start with "A photo of".
- Limit the total length to 40-50 words.
- Focus on bus, bike, car, pedestrian and truck.
- Describe time of day, weather and location.
- Focus on the scene inside the fish-eye lens.
- Use grammatically correct and clear sentences.

H.2. Rephrasing prompt by the manual baseline

You are an expert at visually-grounded image caption rewriting. Your task is to rewrite image captions taken with a fisheye camera so that:
- The objects Bus, Bike, Car, Pedestrian, and Truck are small in scale and located near the edges of the image, where fisheye distortion is strong.
- Object placement and interaction should be plausible within real-world traffic scenes.

Please adhere to the following format for the caption:
- Start with ”A photo of”.
- Limit the total length to 40-50 words.
- Use grammatically correct and clear sentences.

Apply the following object descriptions:
- Bus: large public passenger vehicles
- Truck: heavy-duty vehicles like dump trucks or semi-trailers
- Car: compact vehicles such as sedans, SUVs, or vans
- Bike: bicycles, motorcycles, or scooters, either parked or with riders
- Pedestrian: visible people walking, standing, or crossing

Avoid visual ambiguity between:
- Bus vs Truck $\rightarrow$ contrast size, function, silhouette
- Car vs Truck $\rightarrow$ emphasize bulk and height differences
- Pedestrian vs Bike $\rightarrow$ distinguish by motion, vehicle presence, posture

Ensure variation across scene conditions:
- Camera angles: side-view or front-view
- Intersection types: T-junctions, Y-junctions, cross-intersections, mid-blocks, pedestrian crossings, or irregular layouts
- Lighting: morning, afternoon, evening, or night
- Traffic flow: free-flowing, steady, or busy

When choosing scene elements, slightly favor the following (but still maintain diversity overall):
- Categories prominently shown: Pedestrian and Truck
- Time of day: Day, Afternoon, Night, or general Daytime
- Weather: Clear
- Location type: Urban areas such as City streets

Preserve the core content of the original caption, but rewrite it to reflect the above constraints. If none of the specified categories are present, you may subtly introduce one or more at the distorted outer edges of the image. Always maintain natural, fluent language, and don’t make the added objects the main focus unless already emphasized.

H.2.1. Rephrasing prompt to construct preference dataset

You are an expert in generating diverse yet realistic image captions for road scenes captured by a fish-eye surveillance camera. Your task is to rewrite the caption I give you into 5 diverse and realistic variants.

Please adhere to the following format for the caption:

- Start with "A photo of".
- Limit the total length to 40–50 words.
- Use grammatically correct and clear sentences.
- While preserving the core elements (bus, bike, car, pedestrian, truck) of the original caption, vary:
  - Camera angles: side-view or front-view
  - Intersection types: T-junctions, Y-junctions, cross-intersections, mid-blocks, or pedestrian crossings
  - Lighting: morning, afternoon, evening, or night
  - Traffic flow: free-flowing, steady, or busy
  - Scene content: object count/placement and background features (e.g., buildings, shops, trees, signs, utility poles)
- Ensure each caption includes at least one distinct variation that differentiates it from the others.
- Output the 5 captions as a numbered list, using the format:
  - Caption 1
  - Caption 2
  - Caption 3
  - Caption 4
  - Caption 5
- Do not include any explanation or extra text outside of the list.

H.2.2. Rephrasing prompt with the preference-tuned LLM

You are an expert in generating diverse yet realistic image captions for road scenes captured by a fish-eye surveillance camera. Your task is to rewrite the caption I give you into a realistic variant.

Please adhere to the following format for the caption:

- Start with ”A photo of”.
- Limit the total length to 40-50 words.
- Use grammatically correct and clear sentences.
- While preserving the core elements (bus, bike, car, pedestrian, truck) of the original caption, vary:
  - Camera angles: side-view or front-view
  - Intersection types: T-junctions, Y-junctions, cross-intersections, mid-blocks, or pedestrian crossings
  - Lighting: morning, afternoon, evening, or night
  - Traffic flow: free-flowing, steady, or busy
  - Scene content: object count/placement and background features (e.g., buildings, shops, trees, signs, utility poles)