Any-Class Presence Likelihood for Robust Multi-Label Classification with Abundant Negative Data

Supplementary Material

6. Appendix

This section consists of theoretical justification of the loss, empirical justification of the likelihood surface, tuning of α and λ, time complexity during training, and additional experiments and discussions.

6.1. Learning signal on final layer neurons

Here, we explain how the redesigned loss function influences the learning behavior of the final layer neurons during training. We show that each output neuron receives the error between the synthesized any-class prediction and its target, in addition to the standard gradient during backpropagation. This highlights how the proposed formulation changes learning behavior in the presence of negative samples. To develop this understanding, we first revisit the formulation of conventional MLC with BCE, from input to loss calculation, then break down how each final layer neuron updates based on the error. The network takes an instance $\mathbf{x} \in \mathcal{X}$ as input and the final layer neurons produce M raw scores $\mathbf{Z} = (z_j)_{j=1}^{M}$. The sigmoid activation transforms raw scores into the class prediction probabilities $\mathbf{p} = (p_j)_{j=1}^{M}$, where

$$\begin{aligned} p_j &= \frac{1}{1+e^{-z_j}}, \text{which means} \\ e^{-z_j} &= \frac{1-p_j}{p_j} \quad \text{(for later derivations).} \end{aligned} \tag{17}$$

For simplicity of explanation, the BCE loss from Equation (1) is rewritten as:

$$\mathcal{J}_{bce} = -\sum_{j=1}^{M} \left[y_j \log(p_j) + (1-y_j) \log(1-p_j) \right]. \tag{18}$$

and the redesigned BCE loss from Equation (8) as:

$$\begin{aligned} \mathcal{J}_{any|bce} &= \mathcal{J}_{bce} + \alpha\, \mathcal{J}_{any}, \\ \mathcal{J}_{any} &= -\, y_a \log(p_a) - (1-y_a) \log(1-p_a). \end{aligned} \tag{19}$$

Therefore, the gradient propagated to each final layer neuron z_j can be formulated as follows:

$$\frac{\partial \mathcal{J}_{any|bce}}{\partial z_j} = \frac{\partial \mathcal{J}_{bce}}{\partial z_j} + \alpha \frac{\partial \mathcal{J}_{any}}{\partial z_j} \tag{20}$$

We now use the chain rule to derive the learning signal for each final layer neuron during backpropagation. The partial derivative of standard BCE loss term $\mathcal{J}_{bce}$ with respect to class j neuron is the product of the partial derivative of loss with respect to class j activation output p_j and the partial derivative of p_j with respect to pre-activation neuron z_j:

$$\frac{\partial \mathcal{J}_{bce}}{\partial z_j} = \frac{\partial \mathcal{J}_{bce}}{\partial p_j} \frac{\partial p_j}{\partial z_j} = \frac{p_j - y_j}{p_j(1-p_j)} \cdot p_j(1-p_j) = p_j - y_j \tag{21}$$

To derive the additional gradient term propagated from any class loss term $\mathcal{J}_{any}$ on each neuron, we further simplify the any class probability calculation p_a:

$$\begin{aligned} p_a &= \frac{1}{1 + \frac{\prod_{j=1}^{M} (1-p_j)^{w_j / \sum_j w_j}}{\prod_{j=1}^{M} p_j^{w_j / \sum_j w_j}}} \\ p_a &= \frac{1}{1 + \prod_{j=1}^{M} \left(\frac{1-p_j}{p_j} \right)^{w_j / \sum_j w_j}} \\ p_a &= \frac{1}{1 + \prod_{j=1}^{M} (e^{-z_j})^{w_j / \sum_j w_j}} \\ p_a &= \frac{1}{1+e^{-z^*}} \quad \text{where} \quad z^* = \frac{\sum_j w_j z_j}{\sum_j w_j} \end{aligned} \tag{22}$$

This further simplifies the any class presence probability p_a to the sigmoid of a synthesized raw score z^* which is the weighted mean of final layer raw scores $\mathbf{Z} = (z_j)_{j=1}^{M}$. Therefore, the learning signal on this synthesized raw score z^* is:

$$\frac{\partial \mathcal{J}_{any}}{\partial z^*} = p_a - y_a \tag{23}$$

Tracing this gradient back to each output neuron z_j is done as follows, which also explains how the gradient in each neuron depends on the positivity of the instance and the presence of class j:

$$\begin{aligned} \frac{\partial \mathcal{J}_{\text{any}}}{\partial z_j} &= \frac{w_j}{\sum_k w_k} (p_a - y_a), \\ &= \begin{cases} \dfrac{1}{\sum_k w_k} (p_a - 1), & \text{if } y_a = 1 \text{ and } y_j = 1, \\ \dfrac{\lambda}{\sum_k w_k} (p_a - 1), & \text{if } y_a = 1 \text{ and } y_j = 0, \\ \dfrac{1}{M}\, p_a, & \text{if } y_a = 0\,. \end{cases} \end{aligned} \tag{24}$$

When an instance is positive, each final layer neuron is encouraged to contribute towards keeping any class presence probability p_a close to 1. Neurons corresponding to the present classes receive a stronger influence due to higher weights, while neurons for absent classes receive a lower weight λ. When the instance is negative, each neuron is forced to keep any class probability p_a close to zero.

6.2. Geometric mean and likelihood surface

In Equation (4), the any-class probability is calculated with the normalized geometric mean of the predicted class probabilities. The geometric mean formulation fairly combines the individual probabilities, preserving the scale. To demonstrate this, Figure 2 plots the product and geometric mean of two probabilities and their normalized variants. The plots show that the product gets saturated near $p_1 = 0$ and $p_2 = 0$, while the geometric mean remains sensitive to changes. This confirms that the geometric mean is a better formulation for likelihood.

Additionally, Figure 3 presents the contour plots of BCE likelihood, the any-class likelihood, and the redesigned BCE likelihood for the two-class scenario. The first row of plots shows the scenario in which both classes are present ($y_1 = 1$ and $y_2 = 1$), while the second row depicts the case where only one class is present ($y_1 = 0$ and $y_2 = 1$). The two axes represent predicted probabilities p_1 and p_2. Here, λ is set to 0.05 for any-class presence likelihood. In the two-class scenario, adding any-class likelihood (Figure 3b) to BCE (Figure 3a) results in the redesigned loss (Figure 3c) pushing the decision boundary towards 1-1 value (close to the label), making the redesigned loss a better option. Similarly, for the one-class scenario (the second row of plots), the redesign loss pushes the likelihood surface towards the true value of 0-1 with some skew caused by the λ (Figure 3f).

6.3. Initial Tuning Experiment for α and λ

We conducted a prior systematic hyperparameter tuning experiment using the Optuna framework [1] to do an initial optimization of the parameters α and λ on a subset of the SewerML dataset of 100k images. The optimization objective was to maximize the validation F2 score.The search employed a Tree-structured Parzen Estimator (TPE) as the Bayesian optimization algorithm, configured with 5 startup trials before probabilistic modeling began. The search space was defined as $\alpha \in [0.0, 1.0]$ and $\lambda \in [0.0, 1.0]$, with both parameters sampled continuously using uniform distributions. The TPE algorithm modeled each parameter independently (univariate mode) and evaluated 24 candidate points per trial when selecting the next hyperparameter configuration to evaluate. To automatically terminate unpromising hyperparameter configurations, we implemented median pruning with early trial termination, which commenced after the first 5 trials, and thereafter, in effect after 22 steps for each trial. The pruner evaluates intervals of 4 steps, and a minimum of 5 trials required per step for the pruning decision.

After 50 trials, the Bayesian search identified optimal values of $\alpha = 0.85$ and $\lambda = 0.10$, which yielded the highest validation F2 score. Considering high α (close to 1), for subsequent experiments, we set $\alpha = 1.0$ to also maintain equal weighting with the individual likelihood terms from the original objective function, reflecting a principled balance between the base loss and meta-learning components. Given the critical association of λ with any-class presence likelihood, we conducted a comprehensive ablation study specifically for λ while fixing α at 1.0, and unevenly sampled values for λ between 0-1 with more attention to lower values as initial hyperparameter search revealed low values to be more performant.

6.4. Time Complexity during Training

While our method introduces additional operations during loss calculation, the computational impact is minimal and the operations can be efficiently implemented. As detailed in Appendix 6.1 (Equation 22), we show that the any-class presence probability p_a can be reformulated as:

$$p_a = \frac{1}{1 + e^{-z^*}} \quad \text{where} \quad z^* = \frac{\sum_j w_j z_j}{\sum_j w_j}$$

This demonstrates that p_a is simply the sigmoid of a weighted mean of the final layer raw scores (logits), which requires only:

- One weighted sum operation: $O(M)$
- One division by the sum of weights: $O(1)$
- One sigmoid evaluation: $O(1)$

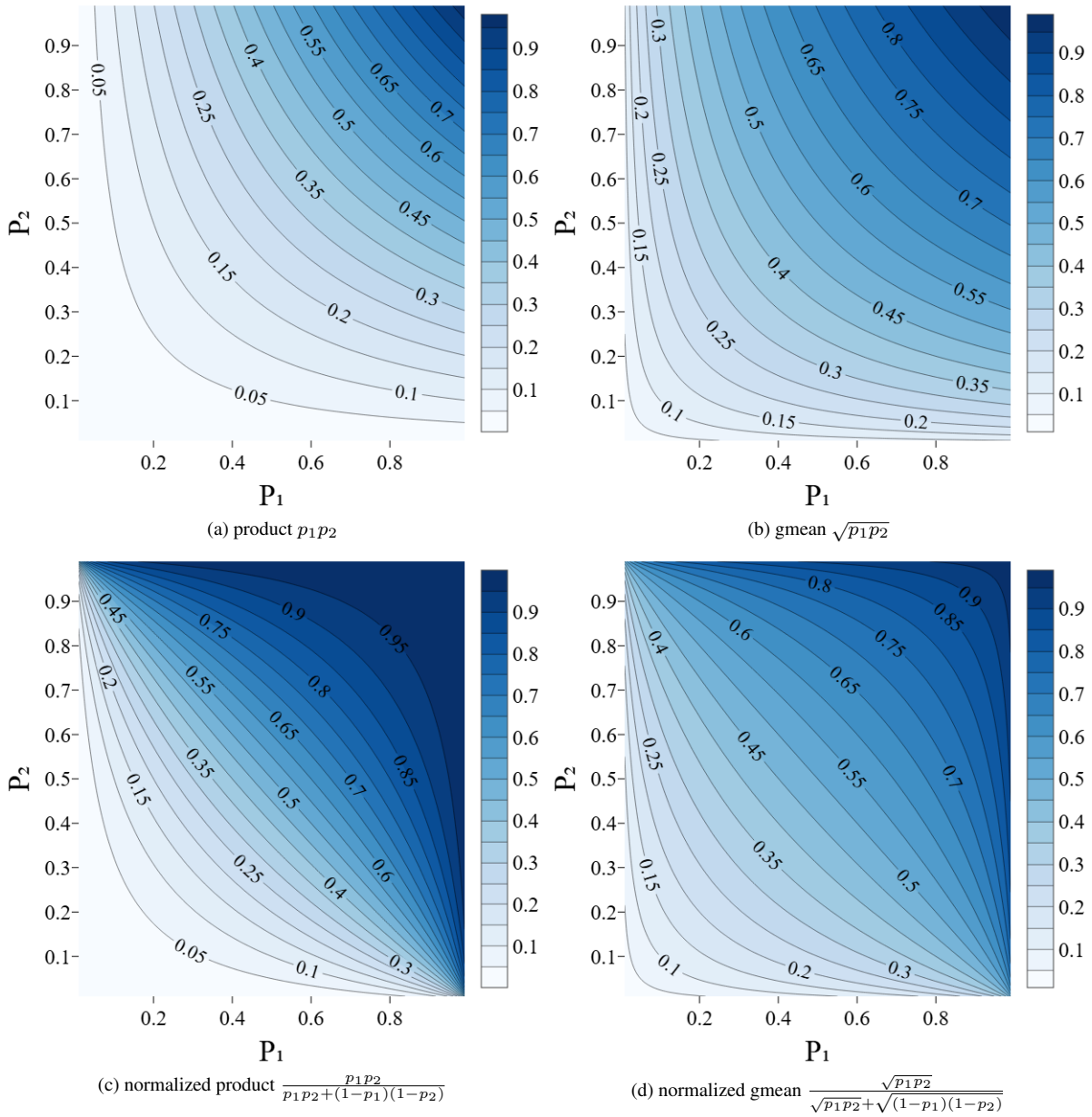
Thus, calculation of p_a simplifies to $O(M)$. The gradient computation (Equation 24) shows that backpropagation through our loss requires only element-wise operations proportional to the number of classes M, with the gradient for each neuron being:

$$\frac{\partial \mathcal{J}_{any}}{\partial z_j} = \frac{w_j}{\sum k w_k}(p_a - y_a)$$

The additional computational cost per training step is $O(M)$ for both forward and backward passes, which is the same complexity as the standard BCE loss computation. Since M (number of classes) is typically small compared to the network size, this represents a negligible overhead relative to the full network forward/backward pass. Modern deep learning frameworks effectively propagate the final error during backpropagation, which avoids any additional backpropagation overhead.

6.5. Additional experiments

λ with Positive to Negative Ratio: In real-world applications the positive–negative ratio can vary. We conducted additional experiments on SewerML subsets with 50k positive examples and varying positive-to-negative ratios (1:1, 1:2, and 1:5), using TresNet-L with focal loss. As shown in Table 5, the proposed objective consistently outperforms the baseline across all ratios. The optimal value of λ shifts with the class imbalance (approximately 0.02 at 1:1, 0.1 at 1:2, and 0.2 at 1:5), which is expected: as negative dominance



(a) product $p_1 p_2$

(b) gmean $\sqrt{p_1 p_2}$

(c) normalized product $\frac{p_1 p_2}{p_1 p_2 + (1-p_1)(1-p_2)}$

(d) normalized gmean $\frac{\sqrt{p_1 p_2}}{\sqrt{p_1 p_2} + \sqrt{(1-p_1)(1-p_2)}}$

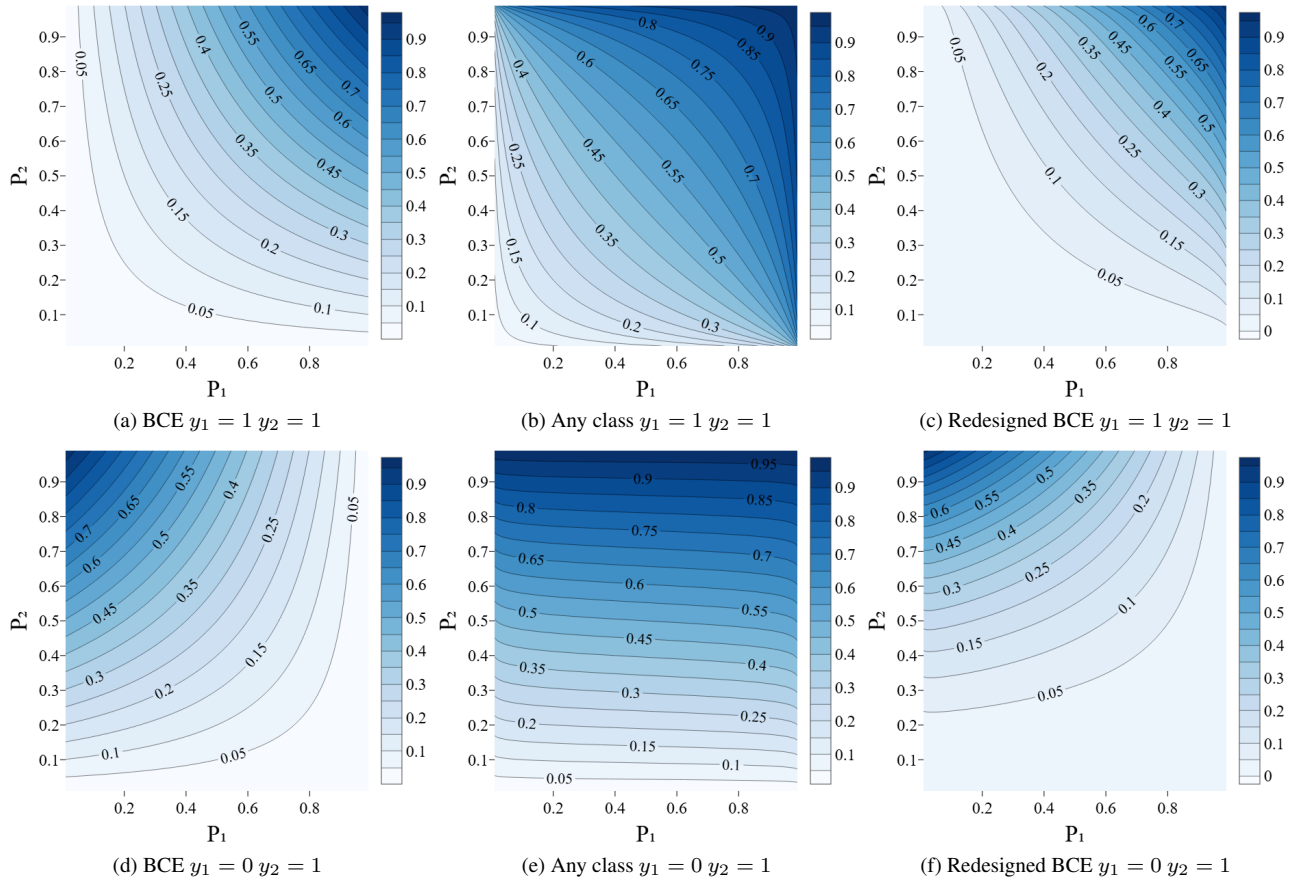
Figure 2. Surface plots of product and geometric mean between two probabilities, and their normalized surfaces. The geometric mean gives a better distribution on a robust scale for optimization.

increases, positive classification benefits more from stronger contrasting. Importantly, performance remains stable over a reasonably wide range of λ values. We are primarily interested in dominant negatives aligning more in real-world than non-dominant negatives. However, the observations can be projected to non-dominant negatives to have a low optimal λ (0.01-0.02).

Supporting Experiments to Main Ablation Study The main ablation study is presented in Subsection 4.4, while in this section, we report additional ablation experiments on the effect of varying λ. Table 6 and Figure 4 show the impact of λ on ChestX-ray14 with BCE Loss, and SewerML with Focal Loss.

6.6. Discussion

Adaptability to Other Works: The proposed loss is applied in conjunction with BCE and focal loss, as well as class-imbalanced variants. More generally, it is extendable to other multi-label classification (MLC) loss functions, such



(a) BCE $y_1 = 1$ $y_2 = 1$ (b) Any class $y_1 = 1$ $y_2 = 1$ (c) Redesigned BCE $y_1 = 1$ $y_2 = 1$

(d) BCE $y_1 = 0$ $y_2 = 1$ (e) Any class $y_1 = 0$ $y_2 = 1$ (f) Redesigned BCE $y_1 = 0$ $y_2 = 1$

Figure 3. Likelihood surface plots for two probabilities, the standard BCE likelihood, our any class likelihood and redesigned BCE likelihood. The first row depicts the case of both targets being 1, and in the second case, only one target is 1 and the any class likelihood is $\lambda = 0.05$

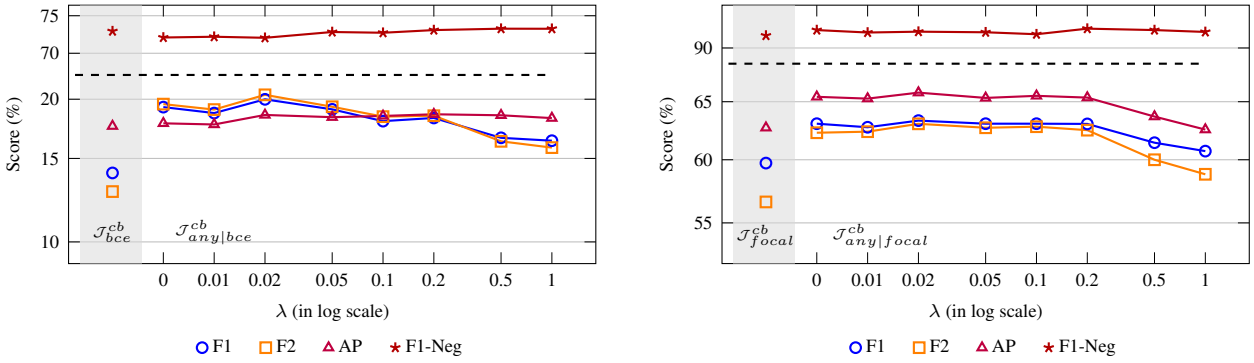
Table 5. TresNet-L - Focal loss - SewerML - Varying positive:negative ratio. Baseline $\mathcal{J}^{cb}_{focal}$ and proposed loss $\mathcal{J}^{cb}_{any|focal}$ with varying λ.

Pos:Neg	Metric	$\mathcal{J}^{cb}_{focal}$	$\mathcal{J}^{cb}_{any\|focal}$ with varying λ							
			0	0.01	0.02	0.05	0.1	0.2	0.5	1
1:1	F1	33.2	37.3	36.78	**37.94**	37.87	37.05	36.22	36.31	36.09
	F2	31.57	39.04	38.59	39.69	**40.09**	38.70	37.25	36.04	35.60
	AP	36.83	39.64	39.80	40.15	**40.55**	39.84	39.15	39.18	38.92
	F1-Neg	81.61	78.97	79.60	79.21	78.97	79.81	80.75	81.80	**82.48**
1:2	F1	32.72	37.72	36.78	37.42	35.87	**38.09**	35.70	37.11	36.43
	F2	31.91	39.98	39.52	40.00	38.53	**40.71**	37.55	38.12	36.67
	AP	35.1	38.76	37.94	38.67	38.22	**39.62**	37.68	38.93	38.11
	F1-Neg	88.29	86.69	86.78	87.18	86.73	87.63	87.60	88.78	**89.31**
1:5	F1	33.76	34.52	33.99	33.89	34.04	35.15	**35.81**	35.66	35.32
	F2	33.95	37.36	37.51	37.14	38.06	38.55	**38.80**	37.84	37.14
	AP	34.17	34.68	35.01	34.67	34.96	35.80	35.63	**36.90**	35.41
	F1-Neg	93.92	92.89	92.83	92.68	92.30	93.03	93.33	93.76	**94.00**

as Asymmetric Loss. This is because the any-class presence objective operates as a complementary term that contrasts positive instances in the presence of substantial negative data, while the base loss primarily enforces discrimination among

Table 6. TresNet-L performance with varying λ values for the ChestX-ray14 dataset with BCE loss and for the SewerML dataset with focal loss

Dataset	Metric	$\mathcal{J}^{cb}_{bce}$	$\mathcal{J}^{cb}_{any\|bce}$ with varying λ							
			0	0.01	0.02	0.05	0.1	0.2	0.5	1
ChestX-ray14	F1	13.98	19.25	18.7	**19.99**	19.04	17.98	18.26	16.58	16.35
	F2	12.77	19.53	19.02	**20.43**	19.28	18.38	18.46	16.3	15.81
	AP	17.55	17.79	17.69	18.53	18.33	18.44	**18.59**	18.5	18.25
	F1-Neg	72.83	71.99	72.09	71.95	72.72	72.62	72.98	**73.17**	73.16
Dataset	Metric	$\mathcal{J}^{cb}_{focal}$	$\mathcal{J}^{cb}_{any\|focal}$ with varying λ							
			0	0.01	0.02	0.05	0.1	0.2	0.5	1
SewerML	F1	59.73	63.06	62.75	**63.33**	63.06	63.06	63.04	61.43	60.72
	F2	56.61	62.28	62.37	**63.05**	62.71	62.80	62.50	60.00	58.81
	AP	62.71	65.44	65.28	**65.82**	65.34	65.53	65.37	63.68	62.55
	F1-Neg	91.21	91.75	91.51	91.59	91.53	91.34	**91.89**	91.75	91.56



(a) ChestX-ray14 - BCE variant

(b) SewerML - focal variant

Figure 4. Performance across varying λ values for TresNet-L network for setting in Table 6

positive labels. Analogous to its role at the loss-function level, the proposed any-class presence optimization can also be viewed as a complementary learning objective for MLC architectural solutions that aim to improve multi-label performance more broadly, including methods based on label correlation modeling and versatile classification heads.

Our current evaluation provides strong empirical evidence of the effectiveness and generalizability of the proposed loss across three diverse network architectures, three datasets, and two base losses, supported by theoretical grounding. These results highlight its complementary nature, and future work could further explore its integration with additional state-of-the-art methods.

Threshold Dependency for Classification: We use a threshold of 0.5 for classification. Since threshold can affect performance especially with imbalanced distributions and negative data, we make sure at least one metric is threshold independent, the mean average precision (mAP) which quantifies the ranking quality of predicted label probabilities which we also used as the validation metric during training.

Dynamics of α and λ In our experiments, we first tune α and λ together to verify preliminary operating point for α and further ablate λ in the paper. The reason for less attention to the parameter α is that given λ is non-zero, both α and λ contribute to a similar goal of contrasting positives from negatives where λ effects more in design while α contributes to the magnitude of effect. Therefore, after verifying with ablation that α fits higher values in 0-1 range, we set $\alpha = 1$ to reflect equal weight for the any-class presence term as the other entropy terms.

Tradeoff between False positives and False Negatives In many practical MLC applications, its more important to reduce false negatives (undiagnosed malignant tumor, undetected pipe fault) than false positives. Hence the proposed loss contrasts positives where there could be more false

positives, but the gain in reducing false negatives is substantial. Nevertheless, as shown in ablations, λ can be tuned to balance this tradeoff depending on the application's goals (higher λs give better F1-Neg than without in all three ablations).

Different Scales of Gain The varying gains across datasets reflect differences in data quality, scale, and overfitting risk. For example, ChestX-ray14 has limited and poor quality data, making models more prone to overfitting, where the regularizing effect of λ is more pronounced. In contrast, SewerML provides over one million training examples.

6.7. Label distributions and COCO dataset class filtering

We evaluate our models on three diverse datasets spanning infrastructure, general object, and medical domains: SewerML, COCO, and Chest X-ray. Each dataset is split into training, validation, and testing subsets, with a mix of samples containing at least one class label ("Any-Class") and samples with no target classes ("Negative"). Table 7 summarizes the distribution statistics across all splits for each dataset.

In its original form, the COCO dataset does not include negative images. This poses challenges for training multi-label models in contexts where object absence is meaningful. To create a dataset with negative classes, we devised a structured class-removing strategy based on statistical analysis of the dataset. By analyzing the raw annotation frequency of the 20 most occurring object classes in the dataset, as shown in Figure 5a, it is evident that the *person* class dominates the dataset with over 260,000 annotations. Thus, the *person* class was the first to be removed. The classes were then grouped into related categories such as *Furniture*, *Kitchenware*, *Vehicles*, and similar. The grouped distribution, shown in Figure 5b, confirmed that the *People* group remained the most dominant, supporting the decision to remove it.

To identify which categories are highly correlated with the *person* class, category-level co-occurrence heatmaps were examined, as seen in Figure 5c. These visualizations revealed strong co-occurrence of *person* with categories such as *Accessories*, *Sports*, and *Vehicles*. Based on these findings, we applied a filtering process to refine the label set. Following the removal of *person* class, classes that strongly correlate with *person* were removed, particularly those related to accessories, sports, and vehicles. To minimize semantic overlap and ambiguity, we also removed highly similar classes such as *bicycle* and *motorcycle* or *cup* and *wine glass*.

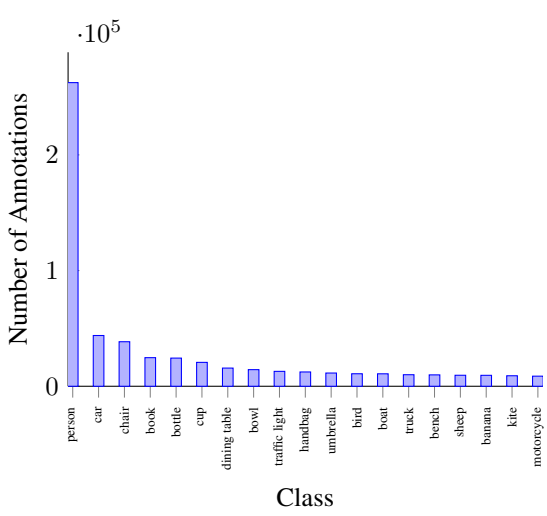
The final list of remaining classes is shown in Table 8. This systematic pruning allowed us to introduce class-absent examples, which are critical for evaluating models that can distinguish between the presence or absence of relevant objects in a scene.

Table 7. Summary of dataset distributions

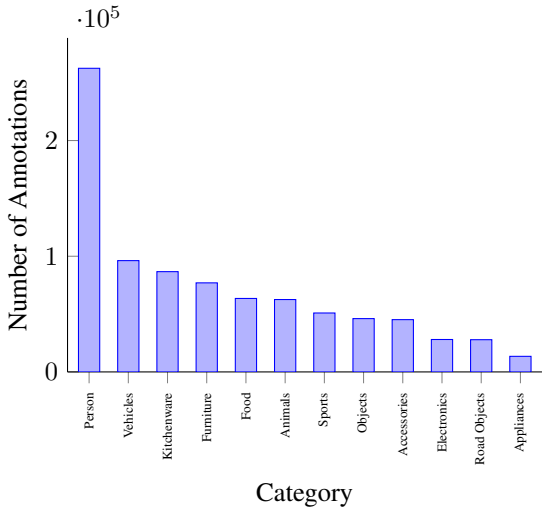
Dataset	Split	Number of Samples	Any-Class	Negative
Chest X-ray	Train	78,566	36,356	42,210
	Validation	17,063	7,821	9,242
	Test	16,491	7,582	8,909
	Total	112,120	51,759	60,361
COCO	Train	86,300	45,658	40,642
	Validation	18,493	9,784	8,709
	Test	18,494	9,784	8,710
	Total	123,287	65,226	58,061
Sewer	Train	1,040,129	487,309	552,820
	Validation	130,046	61,365	68,681

Table 8. COCO: remaining classes

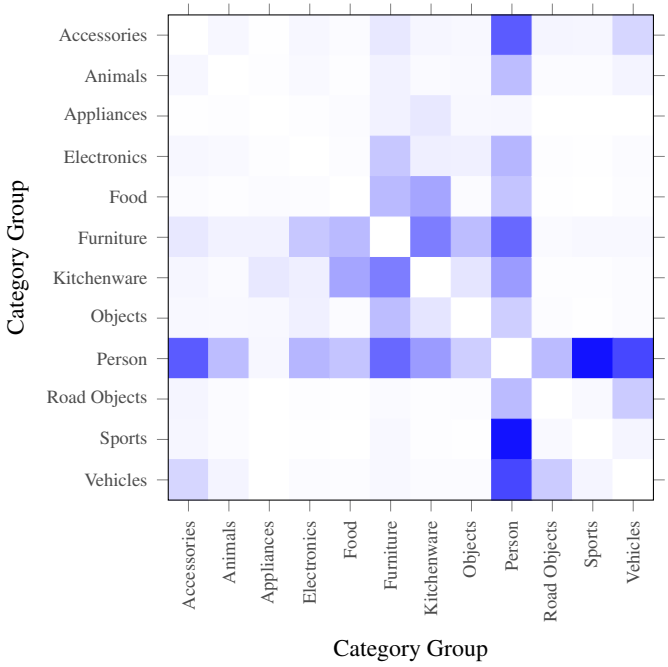
Classes					
boat	bird	cat	dog	bottle	fork
knife	spoon	banana	apple	sandwich	orange
broccoli	carrot	hot dog	pizza	donut	cake
chair	couch	bed	dining table	toilet	tv
laptop	mouse	remote	keyboard	cell phone	microwave
oven	toaster	sink	refrigerator	book	clock
vase	scissors	teddy bear	hair drier	toothbrush	



(a) COCO: 20 most occurring class distribution.



(b) COCO: grouped category distribution.



(c) Category-level co-occurrence heatmap.

Figure 5. COCO: 20 most occurring class distribution, abstract category distribution, and category co-occurrence heatmap. Person is the most frequent class and has the most co-occurrences with other classes.