Learning to Reason: Targeted Knowledge Discovery and Fuzzy Logic Update for Robust Image Recognition

Supplementary Material

6. Synthetic Rule Base Generation

Algorithm 1: Constructing Rule Base $\mathcal{R}$

Input: T: Number of implicit concepts, K: Target classes, S: Concepts $\{s_1, \ldots, s_T\}$, Y: Classes $\{y_1, \ldots, y_K\}$, l: Rules per category (*i.e.* $concepts \Rightarrow class$), and q_{min}, q_{max}: min/max concept combination set size

Procedure $S, Y, l, q_{min}, q_{max}$
- $R_a \leftarrow \emptyset$, $R_b \leftarrow \emptyset$;
- $O \leftarrow \{s \mapsto 0 \mid \forall s \in S\}$;
- `// Phase 1: Core phase`
- **forall** $y_i \in Y$ **do**
  - **for** $j \leftarrow 1$ **to** l **do**
    - $q \leftarrow \texttt{RandomInt}(q_{min}, q_{max})$;
    - $A \leftarrow \texttt{Sample}(S, q)$; `// Sample concepts`
    - **foreach** $s \in A$ **do** $O[s] \leftarrow O[s] + 1$;
    - $R_a \leftarrow R_a \cup \{A \implies \{y_i\}\}$;
    - $R_b \leftarrow R_b \cup \{\{y_i\} \implies A\}$;
    - **if** $P_{neg} > 0$ **then**
      - $Y_{other} \leftarrow Y \setminus \{y_i\}$;
      - $N_{neg} \leftarrow \lfloor P_{neg} \cdot |Y_{other}| \rfloor$;
      - $\hat{Y}_{neg} \leftarrow \texttt{Sample}(Y_{other}, N_{neg})$;
      - **forall** $y_{neg} \in \hat{Y}_{neg}$ **do**
        - $R_a \leftarrow R_a \cup \{A \implies \{\neg y_{neg}\}\}$;
        - $R_b \leftarrow R_b \cup \{\{\neg y_{neg}\} \implies A\}$;
- `// Phase 2: concept-coverage phase`
- $S_{unused} \leftarrow \{s \in S \mid O[s] = 0\}$;
- **forall** $s^* \in S_{unused}$ **do**
  - $q \leftarrow \texttt{RandomInt}(q_{min}, q_{max})$;
  - $A' \leftarrow \texttt{Sample}(S \setminus \{s^*\}, \max(0, q-1))$;
  - $A \leftarrow A' \cup \{s^*\}$;
  - $y^* \leftarrow \texttt{Choice}(Y)$;
  - $R_a \leftarrow R_a \cup \{A \implies \{y^*\}\}$;
  - $R_b \leftarrow R_b \cup \{\{y^*\} \implies A\}$;
  - **foreach** $s \in A$ **do** $O[s] \leftarrow O[s] + 1$
- $\mathcal{R} \leftarrow R_a \cup R_b$
- **return** $\texttt{Shuffle}(\mathcal{R})$

To provide a logical inductive bias for the proposed methodology, we synthesize a knowledge base, denoted as $\mathcal{R}$. This rule base guides the network to learn implicit support concepts by enforcing meaningful relationships between abstract concepts and downstream classes. The synthesis procedure, outlined in Algorithm 1, is divided into two phases: a **Core Phase** to establish foundational class-concept links, and a **Concept-Coverage Phase** to ensure semantic grounding of all available concepts.

Let $S = \{s_1, s_2, \ldots, s_T\}$ denote the set of abstract concepts and $Y = \{y_1, y_2, \ldots, y_K\}$ denote the set of target classes. The final rule base is the union of compositional rules and their converses, $\mathcal{R} \leftarrow R_a \cup R_b$.

6.1. Phase 1: Core Rule Generation

In the initial phase, we generate a minimum of l distinct positive rules for each class $y_k \in Y$. For the l-th rule of class y_k, we determine the antecedent size q by sampling uniformly from the integer interval $[q_{min}, q_{max}]$. In our experiment setup, we fix $[q_{min} = 2, q_{max} = 4]$. We then sample a support set of concepts $\mathcal{A}_{k,l} \subset S$ such that $|\mathcal{A}_{k,l}| = q$, strictly without replacement.

This support set $\mathcal{A}_{k,l}$ forms the basis for **compositional rules** (R_a), as illustrated in Figure 2 (left and middle panels). First, we define the *positive implication* that maps the conjunction of selected concepts to the specific category:

$$\left(\bigwedge_{s \in \mathcal{A}_{k,l}} s \right) \implies y_k. \tag{12}$$

To enforce mutual exclusivity among classes—ensuring a specific concept combination maps uniquely to one category—we explicitly construct *negative associations* against all other classes $y_m \in Y \setminus \{y_k\}$:

$$\left(\bigwedge_{s \in \mathcal{A}_{k,l}} s \right) \implies \neg y_m, \quad \forall m \neq k. \tag{13}$$

Furthermore, to impose a bidirectional logical structure, we generate **converse rules** (R_b). For every positive and negative implication defined above, we add its converse to the knowledge base. For instance, the converse of the positive association in Eq. (12) is defined as:

$$y_k \implies \left(\bigwedge_{s \in \mathcal{A}_{k,l}} s \right). \tag{14}$$

Converse rules for the negative associations are generated analogously.

6.2. Phase 2: Concept-Coverage

A stochastic sampling strategy may leave a subset of concepts in S unselected, rendering them semantically ungrounded. To mitigate this, the second phase identifies all unused concepts and explicitly generates rules for them. We repeat the logic of the Core Phase (generating positive, negative, and converse implications) for these specific concepts, ensuring that every $s \in S$ is grounded in at least one positive compositional rule within $\mathcal{R}$.

7. Qualitative Results

7.1. Positive Cases

Table 6 highlights the qualitative benefits of integrating the KLUE bottleneck into the WideResNet-101 (WRN-101) architecture for multi-label classification. We compare the baseline WRN-101 against two KLUE variants: the standard KLUE and KLUE + $\mathcal{L}_{SAT}$. Both variants utilize WRN-101 as their backbone. The KLUE + $\mathcal{L}_{SAT}$ variant is trained with an additional loss term applied to a subset of rules R_a (cf. Sec. 4 in the main paper). All models were hyperparameter-optimized using the Optuna framework on the PASCAL VOC Train set; general performance metrics are detailed in the main paper (cf. Table 4). For this qualitative evaluation, we selected the best-performing model from each category. As shown in Table 6, KLUE variants demonstrate strong performance gains over the baseline. For example, in image (a), the best KLUE model improves the probability for the Ground-Truth (GT) class "dog" by approximately 32%, while simultaneously suppressing probabilities for the Non-GT "bottle" class. A similar trend is observed in images (b) and (c). Notably, in image (c), the KLUE + $\mathcal{L}_{SAT}$ variant yields a significant improvement for the "pottedplant" class. This suggests that the additional $\mathcal{L}_{SAT}$ term induces a bidirectional logical structure, providing better learning signals during optimization. This allows the model to learn meaningful implicit concepts, effectively treating the learned rules as targeted knowledge that enhances final classification performance.

7.2. Failure Cases

Table 7 presents an evaluation of the models on "hard" examples where all the models struggles. We observe that in certain scenarios, the KLUE variants are unable to correct the predictions. For instance, in image (d), the KLUE models fail to improve the probability for the "motorbike" class. Similarly, in image (e), the probability for the "person" class remains low, likely due to the visual ambiguity of the subject in the scene; however, it is worth noting that the model significantly improves confidence for the "car" class in the same image. A similar limitation is seen in image (f) for the "pottedplant" class. We attribute this to partial visibility, where the learned concepts fail to sufficiently validate the class presence to refine the probability.

Conclusion. In general, KLUE shows limitations when dealing with heavy occlusion or partial visibility in a multi-label classification setting. However, we hypothesize that this could be overcome in an object detection framework. In detection tasks, the model receives additional learning signals from the object's spatial location, which KLUE could leverage to direct its attention more effectively. Validating this hypothesis remains a promising direction for future work.

The corresponding tables for these assessments are presented on the subsequent pages.

7.3. Activation Comparison

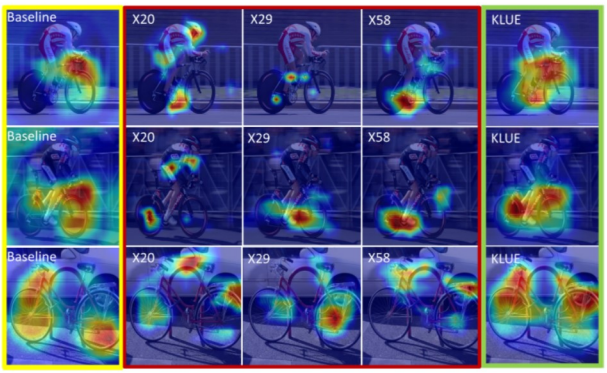


Figure 6. Activation comparison: Baseline WRN-101 (yellow), KLUE concepts (red), KLUE final (green).

Fig. 6, recurring concepts (e.g., x20/x29/x58) activate consistently for semantically similar images and differ for a distinct one; compared to the baseline, KLUE yields more localized, refined activations.

7.4. Scalability / Overhead

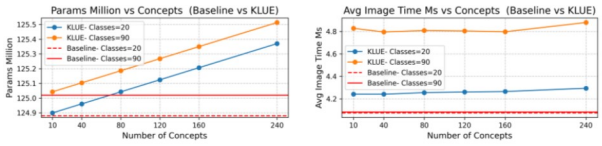
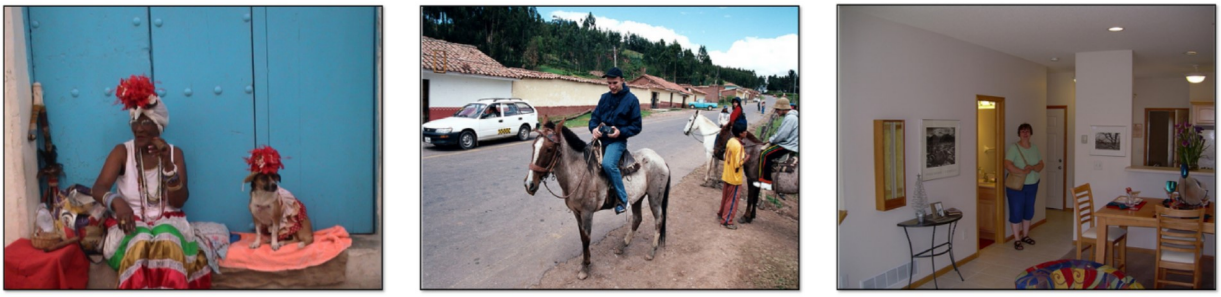


Figure 7. Minimal parameter overhead (left) and constant latency (right) across concept scales. Backbone: WRN-101

Fig. 7 shows KLUE is lightweight: $< 0.5\%$ parameter overhead as classes/concepts grow and **nearly constant inference latency** from 10–240 concepts.



(a) (b) (c)

Image	Label Type	Label	WRN-101 (Baseline)	KLUE			KLUE + $\mathcal{L}_{SAT}$		
			$p_{cls}{}^*$	p_{cls}	p_Δ	$p_\Delta - p_{cls}{}^*$	p_{cls}	p_Δ	$p_\Delta - p_{cls}{}^*$
(a)	GT	person	0.81	0.82	0.99 ↑ +0.17	↑ +0.18	0.81	1.00 ↑ +0.19	↑ +0.19
	GT	dog	0.59	0.77	0.98 ↑ +0.21	↑ +0.39	0.59	0.91 ↑ +0.32	↑ +0.32
	Non-GT	bottle	0.37	0.44	0.31 ↓ −0.13	↓ −0.06	0.37	0.00 ↓ −0.37	↓ −0.37
(b)	GT	person	0.75	0.82	1.00 ↑ +0.18	↑ +0.25	0.92	1.00↑ +0.08	↑ +0.25
	GT	car	0.22	0.77	0.99↑ +0.22	↑ +0.55	0.63	0.97↑ +0.34	↑ +0.61
	GT	horse	0.65	0.97	1.00	↑ +0.35	0.85	1.00 ↑ +0.15	↑ +0.35
	Non-GT	bottle	0.04	0.34	0.09↓ −0.25	↑ −0.05	0.28	0.00 ↓ −0.28	↓ −0.04
(c)	GT	person	0.36	0.99	1.00	↑ +0.64	0.92	1.00↑ +0.08	↑ +0.64
	GT	chair	0.59	0.99	1.00	↑ +0.41	0.69	0.99↑ +0.30	↑ +0.40
	GT	diningtable	0.24	0.76	0.98↑ +0.22	↑ +0.74	0.61	0.95↑ +0.34	↑ +0.71
	GT	pottedplant	0.16	0.29	0.09↓ −0.20	↓ −0.07	0.55	0.79 ↑ +0.24	↑ +0.63

Table 6. Comparison of Baseline WideResNet-101 (WRN-101), the WRN-101-KLUE variant, and WRN-101-KLUE+$\mathcal{L}_{SAT}$ trained with additional loss. All models are trained on PASCAL VOC train set. The top row displays input images (a–c) from PASCAL VOC Val set. We denote the baseline class probability as p^*_{cls}. For KLUE variants, we report the initial probability (p_{cls}) and the refined probability (p_Δ). The values adjacent to arrows within the (p_Δ) column represent the internal refinement magnitude ($p_\Delta - p_{cls}$). The separate column $p_\Delta - p_{cls}{}^*$ quantifies the absolute performance gain/loss of the final refined probability over the original baseline. Rows denote Ground-Truth (GT) labels and Non-GT labels (shown in gray). Green and red text indicate positive and negative effects, respectively. For clarity, absolute difference values smaller than 0.05 (5%) are omitted from the table.

(d)

(e)

(f)

Image	Label Type	Label	WRN-101 (Baseline)	KLUE			KLUE + $\mathcal{L}_{SAT}$		
			$p_{cls}{}^*$	p_{cls}	p_Δ	$p_\Delta - p_{cls}{}^*$	p_{cls}	p_Δ	$p_\Delta - p_{cls}{}^*$
(d)	GT	person	0.98	1.00	1.00	↑ +0.02	0.98	1.00	↑ +0.02
	GT	bicycle	0.89	1.00	1.00	↑ +0.11	0.99	1.00	↑ +0.11
	GT	motorbike	0.18	0.05	0.00 ↓ −0.05	↓ −0.18	0.20	0.00↓ −0.20	↓ −0.20
(e)	GT	person	0.30	0.32	0.06↓ −0.26	↓ −0.30	0.45	0.23↓ −0.22	↓ −0.07
	GT	bus	0.88	1.00	1.00	↑ +0.12	0.88	1.00↑ +0.12	↑ +0.12
	GT	car	0.62	0.97	1.00	↑ +0.38	0.85	1.00 ↑ +0.15	↑ +0.38
(f)	GT	chair	0.82	1.00	1.00	↑ +0.18	0.96	1.00	↑ +0.18
	GT	diningtable	0.32	0.85	1.00↑ +0.15	↑ +0.68	0.82	0.99↑ +0.17	↑ +0.67
	GT	pottedplant	0.18	0.19	0.00↓ −0.19	↓ −0.18	0.35	0.02↓ −0.33	↓ −0.16

Table 7. Comparison of Baseline WideResNet-101 (WRN-101), the WRN-101-KLUE variant, and WRN-101-KLUE+$\mathcal{L}_{SAT}$ trained with additional loss. All models are trained on PASCAL VOC train set. The top row displays input images (d–f) from PASCAL VOC Val set. We denote the baseline class probability as p^*_{cls}. For KLUE variants, we report the initial probability (p_{cls}) and the refined probability (p_Δ). The values adjacent to arrows within the (p_Δ) column represent the internal refinement magnitude ($p_\Delta - p_{cls}$). The separate column $p_\Delta - p_{cls}{}^*$ quantifies the absolute performance gain/loss of the final refined probability over the original baseline. Rows denote Ground-Truth (GT) labels, whereas green and red text indicate positive and negative effects, respectively. For clarity, absolute difference values smaller than 0.05 (5%) are omitted from the table.