NeuroRule: Bridging Vision and Logic with Differentiable Rule Induction (Supplementary Material)

This supplementary document provides additional details that complement the main paper: (i) explicit equations for the rule regularizer $\Omega(\mathrm{Rul})$ used in Eq. (8), (ii) expanded implementation and training protocol, (iii) computational complexity details, and (iv) additional guidance on extracting rule-based reasoning chains.

1. Expanded Implementation Details

In implementation, NeuroRule in PyTorch (v2.0) with CUDA (v11.8) and train on 4 NVIDIA A100 GPUs (80GB each). Unless otherwise stated, images are resized to 512×512. We use batch size 8 with gradient accumulation every 4 steps (effective batch size 32), fix random seeds, and report averages over 5 runs when mean$\pm$std.

The NeuroRule implementation features a modular architecture for scene graph generation, comprising a Mask2Former Feature Extractor that processes images to extract entity features and spatial relationships (currently using dummy implementations for compatibility), a Rule Learner module that applies differentiable logical rules through attention-based predicate selection and tensor operations for multi-relation prediction, and the main NeuroRule model that integrates visual feature extraction with rule-based reasoning using PyTorch's auto grad for end-to-end training. The training pipeline utilizes a custom Scene Graph Dataset for loading JSON-formatted scene graph annotations, employs BCE with Logits Loss for multi-label relation prediction, and implements early stopping with model checkpointing. At the same time, utility functions handle configuration management, metrics computation, and knowledge graph export, all optimized for CPU/GPU execution with configurable hyperparameters for rule complexity and feature dimensions.

Further hardware details are available in the Compute Report PDF file.

1.1. Training Protocol

The training follows a three-phase protocol: initial Mask2Former fine-tuning, joint neural-symbolic optimization, and rule refinement We use a three-phase optimization schedule

Table 1. Key implementation details (consistent with Sec. 4.2 of the main paper).

Component	Setting
Backbone	Mask2Former (Swin-L)
Input resolution	512×512
Optimizer (rule module)	AdamW
Learning rate	5×10^{-5}
Weight decay	1×10^{-4}
Rule template repository size	≤ 1000
Max rule body length	$L = 4$
Rule regularization weight	$\lambda = 0.1$
Batch size / effective batch	8 / 32 (grad accum.$\times 4$)

1. **Mask2Former warm-up.** Fine-tune the Mask2Former backbone to stabilize pixel-precise entity masks and features.
2. **Joint neural-symbolic training.** Optimize the full pipeline end-to-end (visual backbone + predicate scoring + differentiable rule induction).
3. **Rule refinement.** Freeze visual components and continue optimizing the rule module (rule weights and selection logits) under $\Omega(\mathrm{Rul})$ to encourage compact, stable rules.

1.2. Hyperparameters

Table 1 summarizes key hyperparameters used throughout the experiments.

2. Datasets Detailed Descriptions

We evaluate NeuroRule on three complementary benchmarks, VG, PSG, and Open-PSG, chosen to support different tasks of pixel-precise neuro-symbolic reasoning. A comparative overview of these three benchmarks is illustrated with an example in Fig. 1.

Visual Genome (VG). VG is a representative SGG benchmark with 108K annotated images. We follow the standard split and evaluate under the **SGDet** setting using **R@K, mR@K, and AP50**. This benchmark is a strong testbed for NeuroRule compositional rule induction because rela-

Table 2. Dataset summary as specified in Sec. 4.1 of the main paper.

Dataset	Annotation	Primary Evaluation
VG	SGG triplets	SGDet: R@K, mR@K, AP_{50}
PSG	Mask-level panoptic	Cont@R, Supp@R, Cont@
Open-PSG	Open-vocab + NL	OE/OR + comp./transfer

tion prediction must be learned jointly with object localization and contextual reasoning.

Panoptic Scene Graph (PSG). PSG extends COCO with pixel-precise panoptic annotations over 48.7K images, enabling *mask-grounded* spatial understanding. We follow the standard PSG protocol and report spatial relation metrics **Cont@R, Supp@R, Cont@**, which directly test geometric reasoning. This dataset provides strong evidence for NeuroRule because our rules are grounded on mask-level entity representations produced by Mask2Former, aligning the symbolic reasoning chain with precise spatial evidence rather than coarse boxes.

Our current supplementary discussion also describes PSG as containing approximately **49,000** images with pixel-level spatial relation annotations, which is consistent up to rounding.

Open Panoptic Scene Graph (Open-PSG). Open-PSG builds on PSG by adding open-vocabulary annotations and natural-language descriptions, enabling evaluation of *zero-shot* generalization to unseen categories. We evaluate open-world capabilities using the Open-Entity **(OE)** and Open-Relations **(OR)** settings defined in the main paper, together with compositional/transfer metrics reported in Tab.2.

Our current supplementary discussion further characterizes the Open-PSG test set as containing **15,000 test images** with **30%** novel relation combinations and **15%** completely unseen relation types for zero-shot evaluation. These properties make Open-PSG a strong benchmark for NeuroRule, because explicit compositional rules can transfer across contexts and support robust reasoning beyond the training distribution.

2.1. Open-world evaluation settings on Open-PSG

To make the open-world settings explicit, we follow the definitions from Sec. 4.1 of the main paper:

- **Open-Relations (OR):** object classes are seen during training while the *predicate* is unseen at test time.
- **Open-Entity (OE):** predicate classes are seen during training while the *object classes* are unseen at test time.

These settings directly measure whether NeuroRule learned compositional rules and chaining can generalize to novel entities/predicates rather than only memorizing dataset-specific co-occurrences.

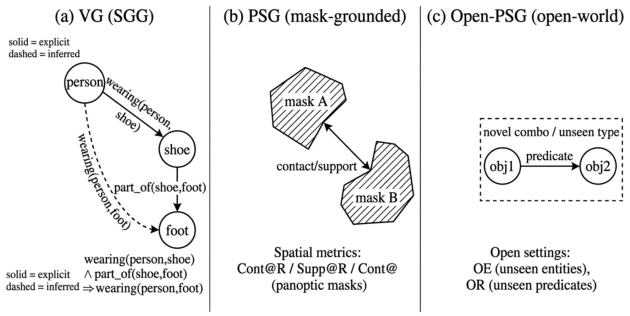

Figure 1. A comparison view of distinct SGG for the dataset. (a) Traditional SGG (VG): Focuses on object-level nodes and utilizes logical reasoning to deduce inferred relations. (b) Mask-Grounded PSG: Shifts from bounding boxes to panoptic masks, capturing precise spatial interactions evaluated via mask-level metrics. (c) Open-World SGG (Open-PSG): Expands beyond fixed vocabularies to discover novel compositional triplets, including unseen entities (OE) and unseen predicates (OR).

3. Details of the Rule Regularization

The main paper defines the training loss (Eq. (8)) and states that $\Omega(\mathrm{Rul})$ contains an L_1 sparsity term Ω_{sparsity} and a semantic term Ω_{semantic} that maximizes rule body–head similarity. Here we provide explicit equations.

Let $\boldsymbol{\alpha} = \{\alpha_r\}_{r=1}^{R}$ be the learnable rule weights (Eq. (4)). We define:

$$\Omega_{\text{sparsity}}(\mathrm{Rul}) = \|\boldsymbol{\alpha}\|_1 = \sum_{r=1}^{R} |\alpha_r|. \tag{1}$$

This encourages a small *active* set of high-weight rules, improving interpretability and reducing spurious rule firing.

3.1. Semantic coherence via body–head similarity

Each predicate $P \in \mathcal{P}$ has an embedding $\mathbf{W}_P \in \mathbb{R}^d$ (main paper, "Predicate Embedding"). For a rule r, denote the head predicate as $P_{\text{head}}^{(r)}$ and the (softly selected) body predicates at position l by weights $\beta_{l,P}^{(r)}$ (Eqs. (5)-(6)). We define the head embedding and a soft body embedding as:

$$\mathbf{e}_{\text{head}}^{(r)} = \mathbf{W}_{P_{\text{head}}^{(r)}}, \tag{2}$$

$$\mathbf{e}_{\text{body}}^{(r)} = \frac{1}{L} \sum_{l=1}^{L} \sum_{P \in \mathcal{P}} \beta_{l,P}^{(r)} \mathbf{W}_P. \tag{3}$$

We then define a cosine-similarity penalty:

$$\Omega_{\text{semantic}}(\mathrm{Rul}) = \frac{1}{R} \sum_{r=1}^{R} \left(1 - \cos\left(\mathbf{e}_{\text{body}}^{(r)}, \mathbf{e}_{\text{head}}^{(r)}\right)\right), \tag{4}$$

where,

$$cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}. \tag{5}$$

This favors semantically aligned body/head predicates and helps avoid logically incoherent clauses. Finally,

$$\Omega(\mathrm{Rul}) = \Omega_{\text{sparsity}}(\mathrm{Rul}) + \Omega_{\text{semantic}}(\mathrm{Rul}). \tag{6}$$

The total loss as in Eq. (8) of the main paper is;

$$\mathcal{L} = \sum_{(I,T)} \sum_{(s,r,o)\in T} (\lambda\, \Omega(\mathrm{Rul}) - \log P(r(s,o))) \tag{7}$$

4. Differentiable Existential in Rule Chaining

The main paper computes multi-hop body truth values with an existential max over intermediate entities (Eq. (7)) and notes a differentiable approximation. We implement the approximation with a temperature-controlled log-sum-exp:

$$t^r_{\text{body}}(s,o) = \max_{\mathbf{e}\in\mathcal{E}^k} \prod_{i=1}^{k} t_{P_i}(e_i, e_{i+1}), \tag{8}$$

$$\max_j x_j \approx \tau \log \sum_j \exp(x_j/\tau), \tag{9}$$

where $\tau > 0$ controls smoothness ($\tau \downarrow 0$ approaches hard max). This yields stable gradients while preserving the intended existential semantics.

4.1. Efficiency and Computational Complexity

We provide a complementary complexity analysis to the inference-time plots in the main paper.

Let N be the number of detected entities, $M \leq N^2$ the number of candidate subject–object pairs after pruning, $P = |\mathcal{P}|$ the number of predicates, R the number of candidate rules (templates), L the maximum body length, and H the maximum chaining depth as shown in Tab. 3.

Predicate scoring. Computing all predicate truth values for M pairs scales as $O(M \cdot P)$ (Eq. (2)).

Rule evaluation (single hop). Evaluating R rules of length L over M pairs is $O(M \cdot R \cdot L)$ (Eqs. (3)-(4)).

Multi-hop chaining. In the depth H, chaining scales as $O(H \cdot M \cdot R \cdot L)$ in the vectorized implementation, with the existential aggregation realized by softmax/log-sum-exp.

Memory. Storing predicate truth tensors and rule activations is $O(MP + MR)$. In practice, the visual backbone dominates computation; the rule module is controlled by bounded (L, H), sparsity in $\boldsymbol{\alpha}$, and pair pruning.

5. Reasoning Chains and Rule Outputs

NeuroRule outputs relation triplets (s, r, o) (SGG edges) and can also provide an explicit *reasoning chain* as an explanation. For a predicted relation $r(s, o)$, we compute each rule's contribution:

Table 3. Asymptotic complexity of the differentiable rule module.

Stage	Time	Memory
Predicate scoring	$O(MP)$	$O(MP)$
Rule eval. (1-hop)	$O(MRL)$	$O(MR)$
Chaining (depth H)	$O(HMRL)$	$O(MR)$

$$\mathrm{score}_r(s,o) = \alpha_r \cdot t^r_{\text{body}}(s,o), \tag{10}$$

The top-K rules by $\mathrm{score}_r(s,o)$, together with the intermediate entities selected by the (soft) existential operator. This yields a step-wise trace, such as the example shown in Fig.3 of the main paper: `wearing(person,shoe)`$\wedge$`part_of(shoe,foot)` $\Rightarrow$ `wearing(person,foot)`.

In this example:

- The target relation and confidence $P(r(s,o))$.
- The top-K contributing rules (rule text + α_r).
- The supporting predicate truth values used in the body.
- The selected intermediate entities for multi-hop chains.