Supplementary: Scene-Specific Anomalous Relationship Detection Using Scene Graph Summarization

A. Hyperparameter analysis

This section analyzes the impact of the hyperparameters in Sec. 6.2. The experiments were primarily conducted using our proposed method, Enhanced Counting-based Scene-specific Anomalous Relationship Detection (Enhanced C-SARD) on an image collection consisting of 31 images ($N = 31$).

Table S1 shows the results with different maximum allowable values n_{max}, which determine the maximum number of high-confidence triplets retained from each scene graph based on their confidence scores. A smaller n_{max} reduces the size of the triplet set $\mathcal{T}$, making anomalous relationships more likely to be ranked higher, thereby improving Recall@1. However, it may also filter out some anomalous relationships, resulting in a lower AUC of Recall@k. Conversely, an excessively large n_{max} may include low-confidence relationships in $\mathcal{T}$, introducing more erroneous predicates. This degrades the performance of soft counting, which, in some cases, even falls short of the performance of hard counting, as shown in Figure S1 and Figure S2. Based on these observations, n_{max} is set to 30 ($n_{\text{max}} = 30$).

Table S2 presents the results with different maximum numbers M of concepts searched for each object from the ConceptNet, as discussed in Sec. 5.2. For each object, up to M concepts are retrieved using "PartOf" and "RelatedTo" relationships. A sufficiently large value provides the best performance, while an excessively large value provides only marginal benefits and increases search time. We selected $M = 30$ as the optimal parameter.

In Table S3, we analyze the impact of various K_{soft} values on the soft counting in Sec. 5.3, where K_{soft} represents the top-K highest-confidence predicate used for counting. A smaller K_{soft} effectively improved performance, while an excessively large K_{soft} led to the inclusion of low-confidence predicates, which negatively affected the results. We selected $K_{\text{soft}} = 10$ as the optimal parameter.

B. Algorithm details

This section provides the implementation details through algorithms, focusing on the difference between hard counting and soft counting, as well as the process of noisy normal relationship reduction.

B.1. The difference between hard counting and soft counting

In the scene graph summarization stage (Sec. 4.2), we count the occurrences of each relationship type from $\mathcal{T}$. The C-SARD uses hard counting to consider the highest-confidence predicate to represent an object relationship while soft counting (Sec. 5.3) takes account of the top K_{soft} predicates. We present their algorithms in Algorithm S1 and Algorithm S2 to highlight the procedural differences: hard counting considers only the predicate with the highest confidence, while soft counting takes several high-confidence predicates into account.

B.2. Noisy normal relationship reduction

Noisy normal relationships refer to the relationships that occur infrequently but are actually normal, as described in Sec. 5.2. One source of the noisy normal relationship arises from minor objects, which can be identified by searching for "PartOf" relationships between objects from ConceptNet. Algorithm S3 shows the process to remove the minor object relationships. In the algorithm, the ObjName includes the names of all objects detected by the SGG model. The second source of noisy normal relationships arises from objects that occur infrequently but are semantically or visually similar to other frequent objects. We identify object similarities by finding shared concepts between objects using the "RelatedTo" relationship from ConceptNet. The hub-aware concept generalization algorithm (Algorithm S4) groups similar objects based on highly relevant concepts and avoids grouping several "hub objects." In the algorithm, HubObjs represents the high-frequency objects within the triplet set $\mathcal{T}$, as defined in Equation (6).

C. Robustness and Applicability

To demonstrate the robustness of our enhancements, we present results using RelTR instead of EGTR for scene graph generation in Table S4. Enhanced C-SARD still outperforms C-SARD.

Although SARD was originally designed for multiple image input, we believe it can also be extended to video input. This can be achieved by extracting frames using either uniform sampling or key frame selection from videos.

Table S1. Performance with different maximum numbers n_{max} for retaining the high-confidence triplets from each scene graph. Bold indicates the best.

n_{max}	AUC-Recall@k	Recall@1
20	55.18	**0.23**
30	**59.45**	0.17
40	37.27	0.05

Table S2. Performance with different maximum numbers M for searching ConceptNet. Bold indicates the best.

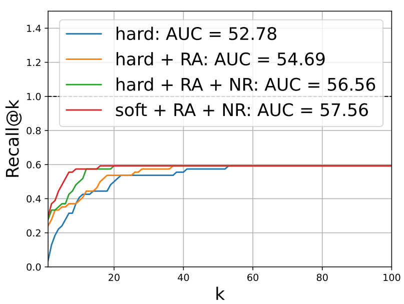
M	AUC-Recall@k	Recall@1
20	51.79	**0.17**
30	59.45	**0.17**
40	**59.98**	0.16

Table S3. Performance with different K_{soft} values for soft counting. Bold indicates the best.

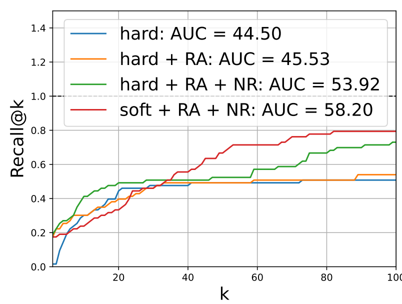
K_{soft}	AUC-Recall@k	Recall@1
5	58.93	**0.17**
10	**59.45**	**0.17**
15	51.44	**0.17**

Table S4. Performance comparison between C-SARD and Enhanced C-SARD. In this case, the scene graph detector is RelTr, pre-trained on the Visual Genome dataset.

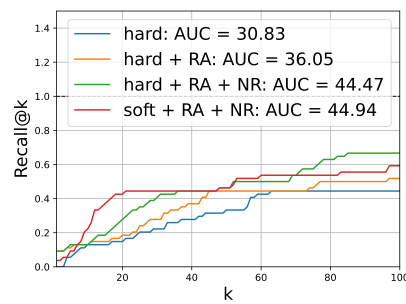
Method	AUC-Recall@k			Recall@1		
	$N = 11$	$N = 21$	$N = 31$	$N = 11$	$N = 21$	$N = 31$
C-SARD	31.58	25.95	19.63	**0.04**	0.02	0.02
Enhanced C-SARD	**38.51**	**32.57**	**28.00**	**0.04**	**0.05**	**0.05**



(a) AUC of Recall@k with $n_{\text{max}} = 20$.



(b) AUC of Recall@k with $n_{\text{max}} = 30$.

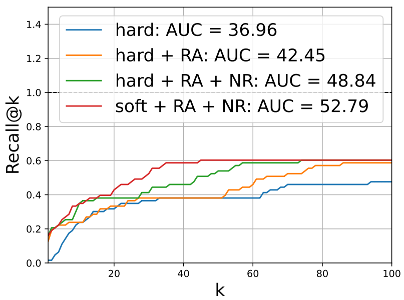


(c) AUC of Recall@k with $n_{\text{max}} = 40$.

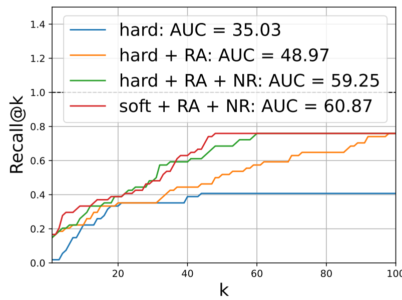
Figure S1. Results for the office scene. The y-axis represents Recall@k values (from 0 to 1), while the x-axis represents k values (from 1 to 100). “RA” and “NR” denote rarity adjustment and noisy normal relationship reduction, respectively.

D. Additional output examples

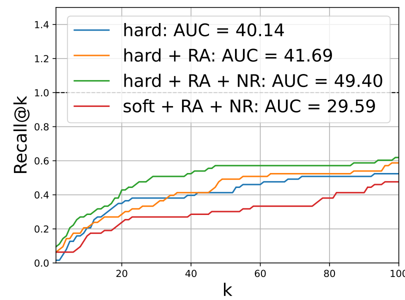
In Sec. 6.3, we provide an output example of the office scene (Figure 6), consisting of 11 images ($N = 11$). In this section, we provide additional examples to demonstrate the effectiveness of Enhanced S-CARD. Figure S3 shows an output example of the dining room scene, containing 11 images ($N = 11$), demonstrating that Enhanced C-SAR outperforms C-SARD in ranking the anomalous relationships at the top. Figure S4 and Figure S5 present output examples of the office scene and the dining room scene, each consisting of 31 images ($N = 31$). In Figure S4, the man within the 5th image is misclassified as a woman by the SGG model, which interferes with the anomalous relationship detection results. Enhanced C-SARD resolves this issue by grouping “man” and “woman,” mitigating the impact of the misclassification. In Figure S5, the anomalous relationship is ranked 118th by C-SARD, but with Enhanced C-SARD, it improves significantly to 13th.



(a) AUC of Recall@k with $n_{\max} = 20$.



(b) AUC of Recall@k with $n_{\max} = 30$.



(c) AUC of Recall@k with $n_{\max} = 40$.

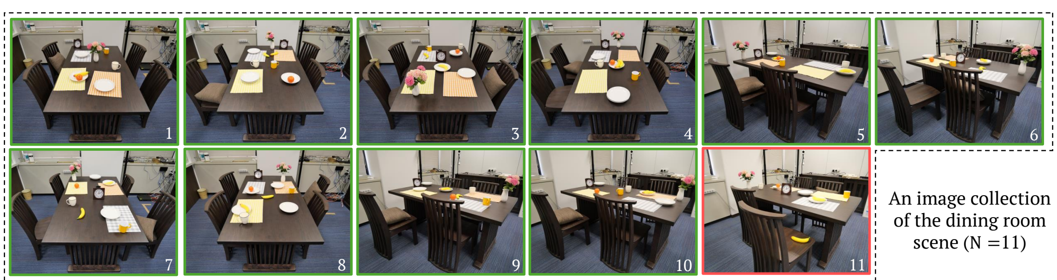
Figure S2. Results for the dining room scene. The y-axis represents Recall@k values (from 0 to 1), while the x-axis represents k values (from 1 to 100). "RA" and "NR" denote rarity adjustment and noisy normal relationship reduction, respectively. When $n_k = 40$, the performance of soft counting decreases because it adopts multiple predicates from low-confidence triplets, leading to incorrect counting.

Algorithm S1 Hard counting

Input: triplet set $\mathcal{T}$.
Output: counting tensor C.
Initialize: C with zeros.
for each $t_k^i = \langle \boldsymbol{s}_k^i, \boldsymbol{p}_k^i, \boldsymbol{o}_k^i \rangle \in \mathcal{T}$ **do**
    $(s_{\text{cls}}, s_{\text{score}}, s_{\text{bbox}}) \leftarrow \boldsymbol{s}_k^i$;
    $(p_{\text{cls}}, p_{\text{score}}) \leftarrow \boldsymbol{p}_k^i$;
    $(o_{\text{cls}}, o_{\text{score}}, o_{\text{bbox}}) \leftarrow \boldsymbol{o}_k^i$;
    $C[s_{\text{cls}}, p_{\text{cls}}, o_{\text{cls}}] \leftarrow C[s_{\text{cls}}, p_{\text{cls}}, o_{\text{cls}}] + 1$;
end for
Return: C.

Algorithm S2 Soft counting

Input: triplet set $\mathcal{T}$, K_{soft}.
Output: counting tensor C.
Initialize: C with zeros.
for each $t_k^i = \langle \boldsymbol{s}_k^i, \boldsymbol{p}_k^i, \boldsymbol{o}_k^i \rangle \in \mathcal{T}$ **do**
    $(s_{\text{cls}}, s_{\text{score}}, s_{\text{bbox}}) \leftarrow \boldsymbol{s}_k^i$;
    $(p_{\text{cls}}, p_{\text{score}}) \leftarrow \boldsymbol{p}_k^i$;
    $(o_{\text{cls}}, o_{\text{score}}, o_{\text{bbox}}) \leftarrow \boldsymbol{o}_k^i$;
    $P_{\text{top}} \leftarrow$ Indices of top K_{soft} scores in p_{score};
    for each p_{top} in P_{top} **do**
        $C[s_{\text{cls}}, p_{\text{top}}, o_{\text{cls}}] \leftarrow C[s_{\text{cls}}, p_{\text{top}}, o_{\text{cls}}] + 1$;
    end for
end for
Return: C.



Scene-specific anomalous relationship detection using C-SARD

1. fruit-on-table (in image 9)
2. banana-on-plate (in image 4)
3. plate-with-food (in image 10)
4. vase-has-flower (in image 8)
5. plate-has-food (in image 2)
6. paper-on-table (in image 5)
7. table-under-plate (in image 10)
8. leg-of-chair (in image 9)
9. **banana-on-chair (in image 11)**
10. lamp-on-table (in image 6)
11. plate-with-banana (in image 4)
12. chair-with-seat (in image 6)

...

Scene-specific anomalous relationship detection using Enhanced C-SARD

1. **banana-on-chair (in image 11)**
2. banana-on-plate (in image 4)
3. plate-with-banana (in image 4)
4. pillow-on-table (in image 3)
5. table-with-banana (in image 11)
6. table-has-cup (in image 1)
7. table-has-flower (in image 1)
8. food-on-table (in image 10)
9. plate-with-food (in image 10)
10. plate-has-food (in image 2)
11. cup-on-plate (in image 4)
12. food-on-plate (in image 10)

...

Figure S3. An example of the scene-specific anomalous relationship detection results on the dining room scene, consisting of 11 images ($N = 11$) including both 10 normal images (green border) and 1 anomalous image (red border). Enhanced C-SARD successfully ranked the anomalous relationship (red text) at the top. The relationships incorrectly ranked above the anomalous one by C-SARD (underline) are moved down.

Algorithm S3 Remove minor object relationships

Input: ObjName, triplet set $\mathcal{T}$, ConceptNet, maximum number of searches M
Output: filtered triplet set $\mathcal{T}_{\text{filtered}}$

// Step 1. Get the names of minor objects
MinorObjs $\leftarrow \emptyset$
for each obj $\in$ ObjName **do**
  Result, Weights $\leftarrow$ ConceptNet.search(src=obj, rel='PartOf', max=M)
  for each mainObj $\in$ Result **do**
    if mainObj $\in$ ObjName **then**
      MinorObjs.insert(obj);
    end if
  end for
end for

// Step 2. Convert object names in MinorObjs to class IDs
for each obj $\in$ MinorObjs **do**
  obj $\leftarrow$ obj corresponding class ID;
end for

// Step 3. Remove triplets with minor object from $\mathcal{T}$
$\mathcal{T}_{\text{filtered}} = \emptyset$;
for each $t_k^i = \langle \boldsymbol{s}, \boldsymbol{p}, \boldsymbol{o} \rangle \in \mathcal{T}$ **do**
  $(s_{\text{cls}}, s_{\text{score}}, s_{\text{bbox}}) \leftarrow \boldsymbol{s}$;
  $(o_{\text{cls}}, o_{\text{score}}, o_{\text{bbox}}) \leftarrow \boldsymbol{o}$;
  if $\{s_{\text{cls}}, o_{\text{cls}}\} \cap$ MinorObjs $= \emptyset$ **then**
    $\mathcal{T}_{\text{filtered}}$.insert($t_k^i$);
  end if
end for
Return $\mathcal{T}_{\text{filtered}}$;

Algorithm S4 Hub-aware concept generalization

Input: counting tensor C, ObjName, HubObjs, ConceptNet, maximum number of searches M
Output: updated counting tensor C

// Step 1: Search for related concepts
Graph $\leftarrow$ Graph();
for each obj $\in$ ObjName **do**
  Result, Weights $\leftarrow$ ConceptNet.search(src=obj, rel='RelatedTo', max=M)
  for each concept, weight $\in$ Result ,Weights **do**
    Graph.add_node(obj, type='object');
    Graph.add_node(concept, type='concept');
    Graph.add_edge(obj, concept, w=weight);
  end for
end for

// Step 2: Concept selection
ConceptBank $\leftarrow \{\}$;
while Graph.has_concept() **do**
  Remove concepts with degree one or zero;
  Select concept with the highest average weight;
  CoveredObjs $\leftarrow$ Graph.neighbors(concept);
  if $|$CoveredObjs $\cap$ HubObjs$| \geq 2$ **then**
    Graph.remove_node(concept);
  else
    ConceptBank[concept] $\leftarrow$ CoveredObjs;
    Graph.remove_node(concept);
    Graph.remove_node_from(CoveredObjs);
  end if
end while

// Step 3: Concept generalization
for each concept $\in$ ConceptBank **do**
  relatedObjs $\leftarrow$ ConceptBank[concept];
  relatedObjs share the counting number in C;
end for
Return C;

Scene-specific anomalous relationship detection using C-SARD

1. man-holding-laptop (in image 3)
2. glass-on-woman (in image 5)
3. towel-on-table (in image 18)
4. woman-has-head (in image 5)
5. pillow-on-table (in image 29)
6. table-near-woman (in image 5)
7. hand-holding-book (in image 27)
8. woman-near-table (in image 5)
9. woman-holding-book (in image 5)
10. woman-holding-cup (in image)
11. hair-of-woman (in image 5)
12. cup-in-hand (in image 19)
13. man-near-table (in image 16)
14. head-of-woman (in image 5)
15. towel-on-chair (in image 18)
16. **cup-on-chair (in image 31)**
17. shoe-under-table (in image 13)
18. pillow-on-chair (in image 29)

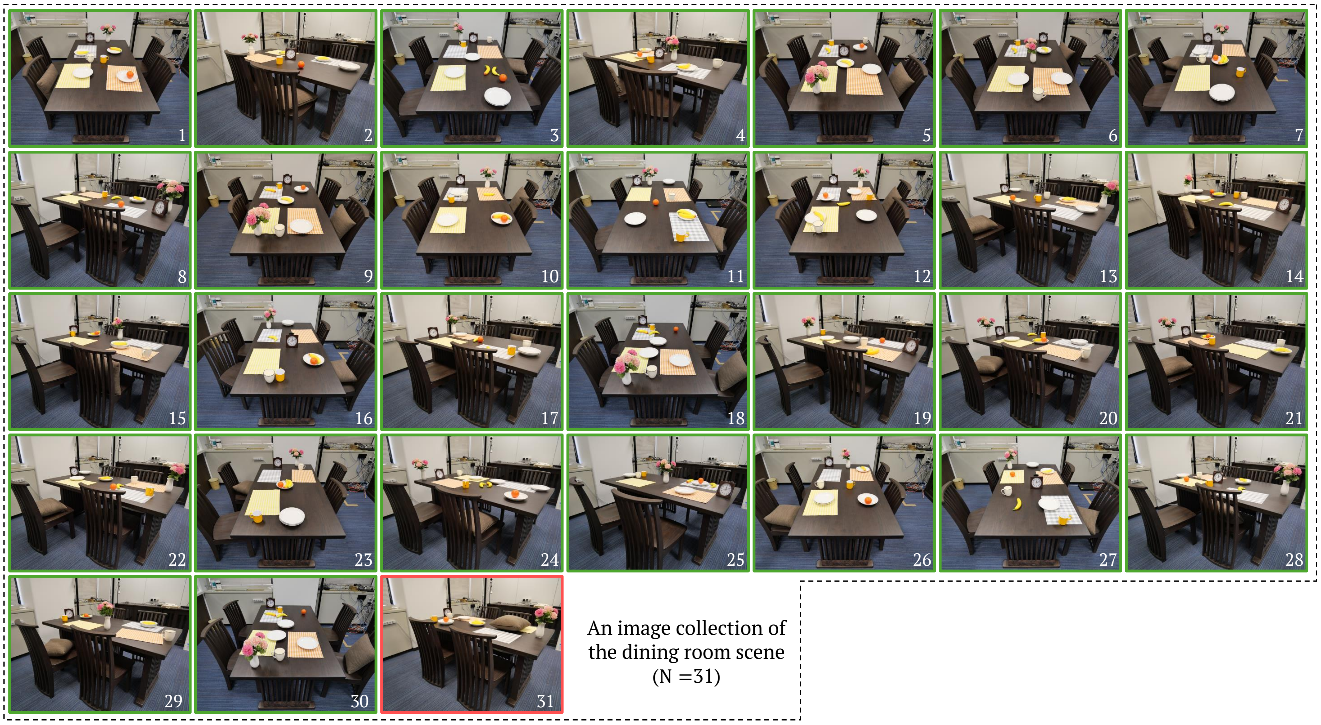
...

Scene-specific anomalous relationship detection using Enhanced C-SARD

1. **cup-on-chair (in image 31)**
2. laptop-near-cup (in image 11)
3. pillow-on-table (in image 29)
4. shoe-under-table (in image 13)
5. chair-near-laptop (in image 29)
6. table-with-book (in image 6)
7. towel-on-table (in image 18)
8. table-has-cup (in image 16)
9. door-behind-chair (in image 4)
10. towel-on-chair (in image 18)
11. pillow-on-chair (in image 29)
12. laptop-in-room (in image 21)
13. jacket-on-chair (in image 24)
14. shirt-on-chair (in image 3)
15. cup-near-laptop (in image 4)
16. chair-near-man (in image 8)
17. shirt-on-chair (in image 2)
18. jacket-on-chair (in image 20)

...

Figure S4. An example of the scene-specific anomalous relationship detection results on the office scene, consisting of 31 images ($N = 31$) including both 30 normal images (green border) and 1 anomalous image (red border). Enhanced C-SARD successfully ranked the anomalous relationship (red text) at the top. The relationships incorrectly ranked above the anomalous one by C-SARD (underline) are moved down.



Scene-specific anomalous relationship detection using C-SARD

1. banana-on-plate (in image 7)
2. banana-in-bowl (in image 11)
3. plate-of-food (in image 31)
4. bowl-on-plate (in image 4)
5. plate-with-banana (in image 7)
6. bowl-with-food (in image 10)
7. plate-under-bowl (in image 4)
8. table-has-flower (in image 18)
9. bowl-has-banana (in image 11)
10. table-has-cup (in image 1)
11. wire-on-table (in image 23)
12. vase-has-flower (in image 10)

⋮

118. pillow-on-table (in image 31)

⋮

Scene-specific anomalous relationship detection using Enhanced C-SARD

1. plate-with-banana (in image 7)
2. table-has-flower (in image 18)
3. table-has-cup (in image 1)
4. wire-on-table (in image 23)
5. banaba-on-plate (in image 7)
6. table-has-clock (in image 16)
7. table-has-clock (in image 30)
8. banana-in-bowl (in image 11)
9. bowl-has-banana (in image 11)
10. chair-with-pillow (in image 21)
11. chair-with-pillow (in image 13)
12. chair-with-pillow (in image 22)
13. **pillow-on-table (in image 31)**
14. pillow-on-table (in image 13)
15. pillow-on-table (in image 5)

...

Figure S5. An example of the scene-specific anomalous relationship detection results on the dining room scene, consisting of 31 images ($N = 31$) including both 30 normal images (green border) and 1 anomalous image (red border). Enhanced C-SARD improves the ranking of the anomalous relationship (red text) significantly. Many of the relationships incorrectly ranked above the anomalous one by C-SARD (underlined) are moved down.