Unexplored Directions for Scene Graph Generation: Ordered by Implementation Feasibility

1 Immediately Implementable Directions

1.1 1. Multi-Scale Union Features (Easiest)

Current Implementation: Single-scale RoI pooling

$$\mathbf{u}_{ij} = \mathrm{RoIAlign}(\mathcal{F}, b_{ij}) \tag{1}$$

Proposed: Feature Pyramid Network (FPN) fusion across scales

$$\mathbf{u}_{ij}^{l} = \mathrm{RoIAlign}(\mathcal{F}_l, b_{ij}), \quad l \in \{2, 3, 4, 5\} \tag{2}$$

$$\mathbf{u}_{ij} = \mathrm{Conv}_{1\times 1}\left(\sum_{l=2}^{5} w_l \cdot \mathrm{Upsample}(\mathbf{u}_{ij}^{l})\right) \tag{3}$$

where w_l are learnable scale weights. Captures fine-grained (level 2) and contextual (level 5) information.

Why Easy: Standard FPN implementation, no architectural changes needed. Add 4 lines of code in feature extraction.

1.2 2. Learnable Spatial Edge Formation

Current Implementation: Hard binary thresholds

$$\nVdash[e_{ij} \in \mathcal{E}] = \nVdash[\mathrm{dis}(b_i, b_j) < 0.5 \vee \mathrm{IoU}(b_i, b_j) > 0.1] \tag{4}$$

Proposed: Attention-based continuous edge scoring

$$s_{ij} = \sigma\left(\mathbf{w}^{\top}\mathrm{ReLU}(\mathbf{W}[\mathbf{v}_i \oplus \mathbf{v}_j \oplus \mathbf{l}_{ij}])\right) \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{d_h \times (2d_v + d_l)}$, $\mathbf{w} \in \mathbb{R}^{d_h}$, σ is sigmoid. Replace binary indicator:

$$\nVdash[e_{ij} \in \mathcal{E}] = \nVdash[s_{ij} > \tau], \quad \tau = 0.5 \text{ initially} \tag{6}$$

Why Easy: Single 2-layer MLP. Replace graph construction logic with learned scoring.

1.3 3. End-to-End Confidence Gating

Current Implementation: Piecewise function with manual thresholds

$$\gamma_{ij} = T(s_{ij}^{b}) = \begin{cases} 0 & s_{ij}^{b} \leq \beta \\ \alpha s_{ij}^{b} - \alpha\beta & \beta < s_{ij}^{b} < \frac{1}{\alpha} + \beta \\ 1 & s_{ij}^{b} \geq \frac{1}{\alpha} + \beta \end{cases} \tag{7}$$

Proposed: Fully differentiable gating via learned network

$$\gamma_{ij} = \mathrm{SoftPlus}\left(\mathrm{MLP}([\mathbf{r}_{i\to j} \oplus \mathbf{n}_i \oplus \mathbf{n}_j])\right) \tag{8}$$

where $\mathrm{SoftPlus}(x) = \frac{1}{\beta}\log(1 + e^{\beta x})$ for smooth gradients.

Why Easy: Replace conditional logic with MLP + activation. No tuning α, β.

1.4 4. Cross-Modal Attention Fusion

Current Implementation: Simple concatenation

$$\mathbf{n}_i^{(0)} = [\tilde{\mathbf{v}}_i \oplus \mathbf{s}_i \oplus \mathbf{l}_i] \tag{9}$$

Proposed: Query-Key-Value attention between modalities

$$\mathbf{Q}_{\text{vis}}^i = \mathbf{W}_Q \tilde{\mathbf{v}}_i, \quad \mathbf{K}_{\text{sem}}^i = \mathbf{W}_K \mathbf{s}_i, \quad \mathbf{V}_{\text{sem}}^i = \mathbf{W}_V \mathbf{s}_i \tag{10}$$

$$\boldsymbol{\alpha}_i = \text{softmax}\left(\frac{\mathbf{Q}_{\text{vis}}^i (\mathbf{K}_{\text{sem}}^i)^\top}{\sqrt{d_k}}\right) \in \mathbb{R}^{1\times 1} \tag{11}$$

$$\tilde{\mathbf{s}}_i = \boldsymbol{\alpha}_i \mathbf{V}_{\text{sem}}^i \tag{12}$$

$$\mathbf{n}_i^{(0)} = \mathbf{W}_{\text{proj}} [\tilde{\mathbf{v}}_i \oplus \tilde{\mathbf{s}}_i \oplus \mathbf{l}_i] \tag{13}$$

Visual features query semantic embeddings before fusion, learning modality relevance.

Why Moderate: Standard scaled dot-product attention. using PyTorch attention.

1.5 5. Semantic Misalignment Analysis (Diagnostic)

Hypothesis: Predicate semantics underperform because word embeddings do not align with visual union space.

Test 1 - Linear Probe:

$$\mathcal{L}_{\text{probe}} = \frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{D}} \|\mathbf{W}_{\text{probe}} \mathbf{s}_{\text{pred}}(r_{ij}) - \mathbf{u}_{ij}\|_2^2 \tag{14}$$

Compute R^2 coefficient. If $R^2 < 0.3$, confirms poor alignment.

Test 2 - Canonical Correlation Analysis (CCA):

$$\max_{\mathbf{w}_s, \mathbf{w}_u} \quad \rho = \frac{\mathbf{w}_s^\top \mathbf{S}^\top \mathbf{U} \mathbf{w}_u}{\sqrt{(\mathbf{w}_s^\top \mathbf{S}^\top \mathbf{S} \mathbf{w}_s)(\mathbf{w}_u^\top \mathbf{U}^\top \mathbf{U} \mathbf{w}_u)}} \tag{15}$$

where $\mathbf{S} \in \mathbb{R}^{N\times d_s}$ (semantic embeddings), $\mathbf{U} \in \mathbb{R}^{N\times d_u}$ (union features). Measure top-1 canonical correlation.

Why Easy: Post-hoc analysis only. Use sklearn.cross_decomposition.CCA. No retraining.

2 Moderately Complex Directions

2.1 6. Contrastive Predicate Learning

Current Implementation: Cross-entropy loss only

$$\mathcal{L}_{\text{CE}} = -\sum_{(i,j)} \sum_{c=1}^{C} y_{ij}^c \log p_{ij}^c \tag{16}$$

Proposed: Triplet loss for rare predicates

$$\mathcal{L}_{\text{triplet}} = \sum_{r\in\mathcal{R}_{\text{rare}}} \max\left(0, \|\mathbf{z}_r - \mathbf{z}_r^+\|_2^2 - \|\mathbf{z}_r - \mathbf{z}_r^-\|_2^2 + m\right) \tag{17}$$

where:

$$\mathbf{z}_r = \mathbf{W}_{\text{proj}} \mathbf{r}_{i\to j} \quad \text{(anchor: rare predicate)} \tag{18}$$

$$\mathbf{z}_r^+ = \mathbf{W}_{\text{proj}} \mathbf{r}_{k\to l} \quad \text{(positive: same class, different instance)} \tag{19}$$

$$\mathbf{z}_r^- = \mathbf{W}_{\text{proj}} \mathbf{r}_{m\to n} \quad \text{(hard negative: nearest neighbor in embedding space)} \tag{20}$$

Combined loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{triplet}}, \quad \lambda = 0.1 \tag{21}$$

Implementation: Mine hard negatives per batch via cosine similarity. Define $\mathcal{R}_{\text{rare}}$ as predicates with < 100 training samples.

Why Moderate: Requires batch mining logic. for mining + loss.

2.2 7. Dynamic Graph Topology Adaptation

Current Implementation: Fixed edges determined by spatial heuristics.

Proposed: Gumbel-Softmax differentiable edge sampling

$$g_{ij} \sim -\log(-\log(\text{Uniform}(0,1))) \quad \text{(Gumbel noise)} \tag{22}$$

$$\tilde{s}_{ij} = \log(s_{ij}) + g_{ij} \tag{23}$$

$$\pi_{ij} = \frac{\exp(\tilde{s}_{ij}/\tau)}{\sum_{k \neq i} \exp(\tilde{s}_{ik}/\tau)} \quad \text{(soft edge probability)} \tag{24}$$

During training, use soft weights. At inference, hard threshold:

$$\mathcal{E}^{(l+1)} = \{e_{ij} : \pi_{ij} > 0.5\} \tag{25}$$

Anneal temperature: $\tau^{(t)} = \max(0.5, \exp(-0.001 \cdot t))$ where t is training step.

Why Moderate: Gumbel sampling adds complexity. Need temperature scheduling.

2.3 8. Ensemble of Bipartite and Homogeneous Graphs

Motivation: Bipartite graphs excel at explicit relational modeling (better R@100 on VG: 29.09%), while homogeneous graphs offer computational efficiency (faster inference, attention-weighted edges). Combine strengths via multi-model ensemble.

Architecture: Train two independent models sharing Faster R-CNN backbone:

$$\text{Model}_{\text{homo}}: \quad \mathbf{p}_{ij}^{\text{homo}} = \text{softmax}(\mathbf{W}_{\text{rel}}^{\text{homo}} \mathbf{e}_{ij}^{(L)}) \tag{26}$$

$$\text{Model}_{\text{bip}}: \quad \mathbf{p}_{ij}^{\text{bip}} = \text{softmax}(\mathbf{W}_{\text{rel}}^{\text{bip}} \mathbf{r}_{i \to j}^{(L)} + \log(\hat{\mathbf{p}}_{\text{freq}})) \tag{27}$$

Joint Optimization - Multi-Task Loss:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{CE}}^{\text{homo}}(\mathbf{p}^{\text{homo}}, \mathbf{y}) + \mathcal{L}_{\text{CE}}^{\text{bip}}(\mathbf{p}^{\text{bip}}, \mathbf{y}) + \lambda_{\text{div}} \mathcal{L}_{\text{div}} \tag{28}$$

where diversity loss encourages complementary predictions:

$$\mathcal{L}_{\text{div}} = -\frac{1}{|\mathcal{E}|} \sum_{(i,j)} \text{JSD}(\mathbf{p}_{ij}^{\text{homo}}, \mathbf{p}_{ij}^{\text{bip}}) \tag{29}$$

Jensen-Shannon Divergence (JSD):

$$\text{JSD}(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\text{KL}(\mathbf{p} \| \mathbf{m}) + \frac{1}{2}\text{KL}(\mathbf{q} \| \mathbf{m}), \quad \mathbf{m} = \frac{\mathbf{p} + \mathbf{q}}{2} \tag{30}$$

Inference - Learned Ensemble:

$$\mathbf{p}_{ij}^{\text{ensemble}} = w_{\text{homo}} \mathbf{p}_{ij}^{\text{homo}} + w_{\text{bip}} \mathbf{p}_{ij}^{\text{bip}} \tag{31}$$

$$w_{\text{homo}}, w_{\text{bip}} = \text{softmax}(\mathbf{W}_{\text{gate}}[\mathbf{e}_{ij}^{(L)} \oplus \mathbf{r}_{i \to j}^{(L)}]) \tag{32}$$

Gating network learns context-dependent weighting (e.g., homogeneous for simple scenes, bipartite for complex interactions).

Why Moderate: Requires training two models jointly with shared backbone. Diversity loss adds. Gating network.

3 Advanced Directions (Future Work)

3.1 9. [Transformer-Based Message Passing]

Current Implementation: GRU cells with attention weights

$$\mathbf{e}_{ij}^{(l+1)} = \mathrm{GRU}_e\left(\mathbf{e}_{ij}^{(l)}, \alpha_{ij}^s \mathbf{W}_e \mathbf{n}_i^{(l)} + \alpha_{ij}^o \mathbf{W}_e \mathbf{n}_j^{(l)}\right) \tag{33}$$

Proposed: Multi-head self-attention over graph structure

$$\mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_H)\mathbf{W}^O \tag{34}$$

$$\mathrm{head}_h = \mathrm{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_k}}\right)\mathbf{V}_h \tag{35}$$

Node-to-Edge cross-attention:

$$\mathbf{Q}_e^{ij} = \mathbf{W}_Q^e \mathbf{e}_{ij}^{(l)}, \quad \mathbf{K}_n = \mathbf{W}_K^n[\mathbf{n}_i^{(l)}; \mathbf{n}_j^{(l)}], \quad \mathbf{V}_n = \mathbf{W}_V^n[\mathbf{n}_i^{(l)}; \mathbf{n}_j^{(l)}] \tag{36}$$

$$\mathbf{e}_{ij}^{(l+1)} = \mathrm{LayerNorm}\left(\mathbf{e}_{ij}^{(l)} + \mathrm{MLP}(\mathrm{MHA}(\mathbf{Q}_e^{ij}, \mathbf{K}_n, \mathbf{V}_n))\right) \tag{37}$$

Edge-to-Node cross-attention:

$$\mathbf{Q}_n^i = \mathbf{W}_Q^n \mathbf{n}_i^{(l)}, \quad \mathbf{K}_e = \mathbf{W}_K^e[\mathbf{e}_{ij}^{(l+1)} : j \in \mathcal{N}(i)] \tag{38}$$

$$\mathbf{n}_i^{(l+1)} = \mathrm{LayerNorm}\left(\mathbf{n}_i^{(l)} + \mathrm{MLP}(\mathrm{MHA}(\mathbf{Q}_n^i, \mathbf{K}_e, \mathbf{V}_e))\right) \tag{39}$$

Why Hard: Major architectural overhaul. Requires positional encodings for graph structure. Memory overhead ($\mathcal{O}(N^2H)$ for H heads).

3.2 10. [CLIP-Based Zero-Shot Transfer]

Current Implementation: Word2Vec/BERT trained on text-only corpora.

Proposed: Vision-language pretraining integration

$$\mathbf{s}_i = \mathrm{CLIP}_{\text{text}}(\text{“a photo of a } [c_i]''), \quad \mathbf{s}_i \in \mathbb{R}^{512} \tag{40}$$

For unseen predicates r_{new}, construct text prompt:

$$\mathbf{z}_{r_{\text{new}}} = \mathrm{CLIP}_{\text{text}}(\text{“}[o_i]\ r_{\text{new}}\ [o_j]'') \tag{41}$$

Similarity-based ranking over all candidate predicates:

$$p(r \mid \mathbf{r}_{i\to j}) = \frac{\exp(\mathrm{sim}(\mathbf{r}_{i\to j}, \mathbf{z}_r)/\tau)}{\sum_{r' \in \mathcal{R}_{\text{all}}} \exp(\mathrm{sim}(\mathbf{r}_{i\to j}, \mathbf{z}_{r'})/\tau)} \tag{42}$$

where $\mathrm{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}$ is cosine similarity, $\tau = 0.07$ (CLIP default).

Why Hard: Requires CLIP model integration. Zero-shot evaluation needs new data splits. Prompt engineering sensitive.

3.3 11. [Causal Intervention for Bias Mitigation]

Problem: Spurious correlations from dataset bias (e.g., “person-riding” biased toward “horse” due to visual co-occurrence).

Proposed: Backdoor adjustment via causal intervention. The observational distribution $P(r \mid o_i, o_j)$ is confounded by scene context c. Approximate interventional distribution:

$$P(r \mid \mathrm{do}(o_i, o_j)) = \sum_{c \in \mathcal{C}} P(r \mid o_i, o_j, c)P(c) \tag{43}$$

where $\mathrm{do}(\cdot)$ denotes cutting incoming edges to (o_i, o_j) in the causal graph.

Implementation - Stratified Training:

1. Cluster images by context c using K-means on global image features: $\mathcal{C} = \{c_1, \ldots, c_K\}, K = 10$
2. Train context-conditioned predictor:

$$\mathbf{p}_{ij}^{c} = \text{softmax}(\mathbf{W}_{\text{rel}}^{c}[\mathbf{r}_{i \to j} \oplus \mathbf{c}]) \tag{44}$$

3. Marginalize during inference:

$$\mathbf{p}_{ij} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{p}_{ij}^{c_k} \tag{45}$$

Loss Function:

$$\mathcal{L}_{\text{causal}} = \mathbb{E}_{c \sim \text{Uniform}(\mathcal{C})} \left[\mathcal{L}_{\text{CE}}(\mathbf{p}_{ij}^{c}, y_{ij}) \right] \tag{46}$$

Ensures uniform sampling across contexts during training, breaking spurious correlations.

Why Hard: Requires causal graph assumptions. Context clustering adds preprocessing. Need to validate deconfounding via backdoor criterion. K-fold context training increases compute $K\times$.

4 Implementation Priority Summary

Order	Direction	Impact
1	Multi-Scale Union Features	High
2	Learnable Edge Formation	High
3	End-to-End Confidence Gating	Medium
4	Cross-Modal Attention	Medium
5	Semantic Misalignment (Diagnostic)	Low (insight)
6	Contrastive Learning	High
7	Dynamic Graph Topology	Medium
8	Ensemble Bipartite+Homogeneous	Very High
	Future Work (Bracket Ideas)	
9	[Transformer Message Passing]	Very High
10	[CLIP Zero-Shot]	High
11	[Causal Intervention]	Medium

Table 1: Ordered by implementation feasibility. Impact assessed by expected improvement on R@100 and mR@100 metrics.

Recommended Implementation Path:

1. **Immediate** Directions 1-3. Quick wins with minimal code changes.
2. **Short-term** Directions 4-5. Requires careful debugging of attention mechanisms.
3. **Mid-term** Directions 6-7. More complex but high-impact on long-tail performance.
4. **Mid-term** Direction 8 (ensemble). Strong candidate for publication-worthy contribution.
5. **Future Research:** Directions 9-11. Suitable for follow-up papers or Ph.D. chapters.