# Frame2KG: A Benchmark and Evaluation Toolkit for Interpretable Frame-to-Graph Generation

**Lewis Watson, Carl Strathearn, Kenny Mitchell, and Yanchao Yu**
Edinburgh Napier University
{L.Watson, C.Strathearn, K.Mitchell2, Y.Yu}@napier.ac.uk

**Abstract**

Interpretable frame-to-knowledge-graph (Frame2KG) generation enables structured visual scene representation while supporting on-device inference to enhance privacy, improve interpretability, and minimise compute. We introduce **Frame2KG-YC2**, a synthetic, reproducible dataset derived from YouCook2 that pairs keyframes with schema-valid JSON knowledge graphs containing typed, spatially grounded entities and semantic predicates, alongside faithful textual paraphrases. Using this corpus, we fine-tune Qwen2.5-VL models (3B and 7B) with parameter-efficient LoRA adapters on attention layers (QKVO), with and without GateProj/Up/Down MLP projections. For evaluation and benchmarking, we propose a deterministic toolkit featuring two-stage node matching, an IoU gate followed by Hungarian assignment on blended spatial-semantic similarity, and comprehensive metrics spanning node/edge precision-recall-F1, matched-pair IoU, and structural validity. On a held-out test set, our models achieve Node $F1_{\mu}$ up to 0.621 and Edge $F1_{\mu}$ up to 0.208, with mean matched IoU of $\approx 0.61$ and $> 98\%$ schema conformity. We show that MLP gating consistently improves predicate accuracy and spatial grounding, while post-training quantisation maintains accuracy and improves deployability on edge hardware. We release the dataset, code, adapters, and evaluation toolkit to establish an open, interpretable baseline for future temporal and multi-view extensions.

**Keywords:** Frame-to-Knowledge-Graph (Frame2KG), conversational visual understanding, on-device real-time inference, quantisation, LoRA fine-tuning, visual knowledge graphs

## 1. Introduction

Embodied robots must perceive, plan, and communicate under tight constraints on compute, latency, and energy. However, traditional methods used in robotics, such as simultaneous localisation and mapping (SLAM), focus on perceptual planning without querying or contextualising objects in scenes by their attributes, i.e. positional data, utility/affordance knowledge and related actions. Thus, structured, queryable scene representations - typed entities grounded to image regions and linked by semantic relations - enable interpretable reasoning, verifiable actions, and privacy-preserving on-device operation. While vision–language models (VLMs) can describe scenes in free-form text and detection pipelines can localise objects, to our knowledge there is no widely adopted, open, and reproducible benchmark for mapping a single frame to a schema-valid, localised knowledge graph that is evaluated for both accuracy and efficiency.

In this paper, we scope the task and benchmark to cooking scenes via a synthetic dataset derived from YouCook2 - a large-scale collection of third-person instructional cooking videos (Zhou et al., 2018) - as a realistic deployment setting for grounded, on-device perception, and treat broader domain coverage as a primary extension.

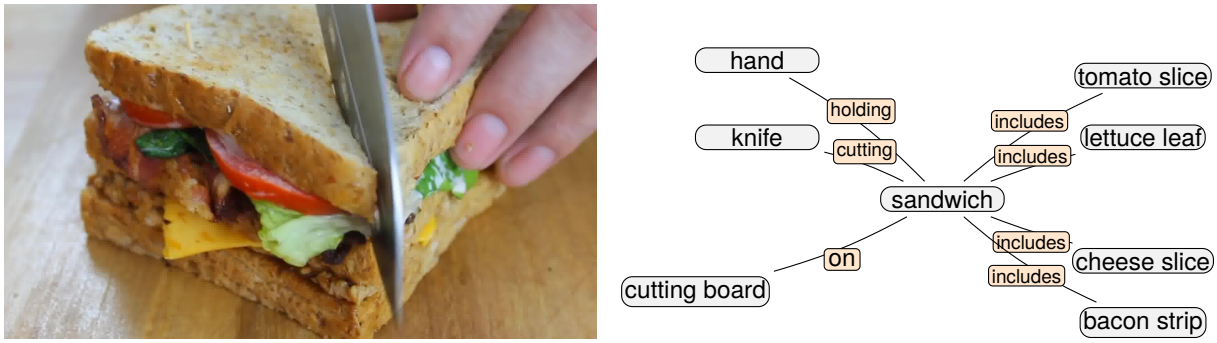
### 1.1. Frame2KG and Frame2KG-YC2

We address this gap with Frame2KG: a task, dataset, adapters, and evaluation toolkit for frame-level knowledge graph generation. Specifically, we release Frame2KG-YC2, a synthetic and reproducible dataset derived from YouCook2 (Zhou et al., 2018) that pairs frames with structured graphs whose nodes are typed entities with normalised bounding boxes with confidences, and whose edges are typed predicates. The end-to-end image-to-graph transformation is programmatic and auditable, and we additionally provide compact textual paraphrases (*graph_sentence*) that are faithful to the graphs to support instruction-style prompting and graph–text alignment.

### 1.2. Baselines

To establish open baselines, we fine-tune Qwen2.5-VL (Bai et al., 2025) in 3B and 7B variants using parameter-efficient LoRA adapters (Hu et al., 2022) on attention projections (Q,K,V,O), with and without light MLP gating (gate/up/down). Our aim is not to introduce a new decoder, but to assess how far small and mid-sized VLMs can be adapted to emit grounded, schema-valid graphs with favourable efficiency–accuracy trade-offs for edge devices; we release four adapters with complete training and inference recipes.

A central contribution is a deterministic, localisation-first evaluation toolkit. Nodes are aligned in two stages: an IoU gate (Lin et al., 2015) filters candidates, then a text-similarity score from a lightweight sentence encoder blends with IoU and supports one-to-one Hungarian assignment (Kuhn,




(a) Frame (b) Knowledge graph

**Graph sentence:** *A hand holding a silver knife is cutting a multi-layered sandwich filled with lettuce, tomato, cheese, and bacon on a wooden cutting board.*

Figure 1: Example item from Frame2KG-YC2. (a) Source frame; (b) visualisation of the generated knowledge graph (entities and predicates). Training set - video_id: nbiOaHaKuKs, frame_number: 0146.

1955). We report node/edge precision–recall–F1 (micro/macro), matched-pair IoU statistics, and explicit JSON validity and schema conformity, with optional throughput reporting. Predicate matching uses normalised string equality by default; a semantic variant is available for lexically diverse relation sets. This design avoids brittle exact matches while preserving localisation as the primary correspondence signal.

We also study deployment efficiency through evaluation of post-training quantisation at 4bit and 8bit precision (Dettmers et al., 2022) measuring their effects on structured accuracy and runtime. In testing, quantisation preserves accuracy within small deltas and reduces memory footprint, supporting on-device, private scene understanding.

On the held-out test set, our best model achieves Node $F1_{\mu}$ up to 0.621 and Edge $F1_{\mu}$ up to 0.208, with mean matched-pair IoU around 0.61 and >98% schema conformity. We observe that adding MLP gating improves predicates and localisation, while the 7B backbone aids downstream node matching. By releasing the dataset, code, adapters, and a deterministic evaluation toolkit, we provide, to our knowledge, the first open end-to-end baseline for Frame→KG with explicit emphasis on reproducibility, efficiency, and structured interpretability, laying the groundwork for temporal knowledge graphs and multi-view extensions.

### 1.3. Contributions

We release: (i) **Frame2KG-YC2**, a synthetic, reproducible dataset pairing frames with strict-JSON graphs and *graph_sentence*; (ii) **Open baselines**, four LoRA adapters for Qwen2.5-VL (3B/7B; QKVO with/without gated projections), with full training and inference recipes; (iii) a **deterministic evaluation toolkit** with localisation-first matching (IoU → semantic + Hungarian) and metrics for node/edge PRF1, box IoU, validity, conformity, and throughput; (iv) a **quantisation study** showing near-parity structured accuracy with improved deployability on edge hardware.

By releasing data, code, adapters, and a transparent evaluation suite, this work provides, to our knowledge, the first open, end-to-end baseline for frame-to-knowledge-graph generation with an explicit emphasis on reproducibility, efficiency, and structured interpretability, laying the groundwork for temporal graphs, multi-view fusion, and interactive correction in real-world embodied systems[1].

## 2. Related Work

Scene Graph Generation (SGG) represents images as object–relation graphs, popularised by Visual Genome (Krishna et al., 2017). Recent transformer approaches such as EGTR (Im et al., 2024) and DSGG (Hayder and He, 2024) improve relation extraction and graph prediction. Open-vocabulary SGG increasingly leverages VLMs to move beyond fixed label sets, including role-playing or text-intermediate pipelines (He et al., 2022; Chen et al., 2024; Li et al., 2024). Evaluation has also been scrutinised, with SGBench advocating standardised metrics and implementations (Lorenz et al., 2024). In contrast, our setting emits schema-valid JSON graphs directly from images, enforces normalised bounding boxes and confidences, and evaluates with a deterministic, localisation-first matcher while reporting validity/conformity alongside node/edge PRF1 and matched-pair IoU, with throughput to reflect deployability.

[1] https://github.com/lewiswatson55/frame2kg

Embodied AI increasingly adopts relational world models for planning and interaction, typically in the form of SLAM-related frameworks. 3D Dynamic Scene Graphs fuse metric-semantic SLAM with layered graphs over places, objects, and agents to support spatial queries and planning (Rosinol et al., 2020), while ConceptGraphs attach open-vocabulary semantics to 3D instances for natural-language-guided manipulation (Huang et al., 2023). These works demonstrate the utility of graph-structured memory for control but typically require heavy mapping stacks. Our Frame→KG target is complementary: lightweight, per-frame graphs suitable for on-device inference, providing a foundation for later temporal fusion rather than a full SLAM pipeline.

Large VLMs advance grounded language understanding via instruction tuning and alignment modules (e.g., LLaVA, InstructBLIP) (Liu et al., 2023; Dai et al., 2023), with models such as BLIP-2 showing that frozen vision backbones can be bridged to LLMs for strong zero/few-shot transfer (Li et al., 2023). Qwen-VL further improves fine-grained grounding (Bai et al., 2025). Most systems, however, prioritise free-form text over schema-constrained outputs. We instead adapt Qwen2.5-VL to emit structured graphs directly, using parameter-efficient fine-tuning: LoRA trains small low-rank adapters while keeping backbones frozen (Hu et al., 2022), and QLoRA combines LoRA with 4-bit quantisation to reduce memory and compute (Dettmers et al., 2023). Our contributions complement these efforts by releasing an auditable dataset, adapters, and a deterministic evaluation toolkit for structured graph generation.

Complementary work on video relation detection and temporal scene graphs (Shang et al., 2017; Ji et al., 2020) and VideoQA (Jang et al., 2017; Lei et al., 2018) underscore the value of temporal grounding. Additionally, Vision-Language-Action (VLA) models learn policies directly from paired vision, language, and action, e.g., RT-1/RT-2 (Brohan et al., 2023a,b), PaLM-E (Driess et al., 2023), SayCan (Ahn et al., 2022), and the Open-X Embodiment collaboration (Collaboration, 2023). These systems distil web-scale world knowledge into control and demonstrate broad generalisation; however, these models typically emit action sequences rather than explicit, schema-valid scene representations. In contrast, our Frame2KG setting is supplementary: we target compact, grounded, queryable graphs suitable for *efficient, on-device representation and reasoning*, which can serve as interpretable inputs or constraints for VLA planners and future temporal graph agents.

## 3. Frame2KG-YC2 Dataset and Synthetic Pipeline

We introduce *Frame2KG-YC2*, a synthetic, auditable, and fully reproducible dataset that annotates a derived sample of video frames from YouCook2 (Zhou et al., 2018). The frames are annotated as structured, schema-valid knowledge graphs. Each sample pairs an RGB frame with a graph that localises entities to normalised bounding boxes and connects them through semantic predicates. To support multimodal learning, we also provide *graph_sentence* - short, faithful textual paraphrases of each graph for instruction-style fine-tuning and graph–text alignment.

### 3.1. Task Definition

Given a single video frame, the model must generate a structured graph consisting of *nodes* (unique ID, label, attributes, and a normalised bounding-box location (x1,y1,x2,y2) with confidence) and *edges* (directed predicates linking source and target nodes). Evaluation emphasises schema validity and grounded localisation, with node alignment performed via IoU-gated semantic matching and edge alignment computed over the induced node mapping (see Section 5).

In an embodied setting, the generated graph is intended to be consumed as robot state for queryable perception and explainable decision-making, and as a unit that can be aggregated over time into temporal graphs.

### 3.2. Source Corpus and Frame Selection

We sample frames from YouCook2 (Zhou et al., 2018) - a dataset containing 2,000 untrimmed YouTube videos across 89 cooking recipes - at 1 fps using `ffmpeg`[2] and apply keyframe filtering with OpenCLIP ViT-B/32 embeddings (Radford et al., 2021)[3] to remove near-duplicates. For each video, we retain the first frame and include subsequent frames only if their cosine similarity to the previous kept frame falls below a threshold $\tau = 0.90$. A higher $\tau$ preserves more frames (looser filtering), whereas a lower value yields sparser, more diverse samples.

### 3.3. Synthetic Annotation Pipeline

Annotations are automatically generated using OpenAI's GPT-o4 (OpenAI, 2025) as a vision language teacher. Batched prompts specify a strict JSON schema requiring that each node contain

[2] https://ffmpeg.org

[3] https://hf.co/laion/CLIP-ViT-B-32-laion2B-s34B-b79K

*id*, *label*, *attributes*, and *location*, and that each edge contain *source*, *target*, and *predicate*. The locations are encoded as "`x1,y1,x2,y2,conf`" (all in `[0,1]`). Invalid or incomplete outputs are re-validated and re-generated automatically. To ensure cost efficiency and reproducibility, annotations are produced in JSONL batches (one frame per line), reducing API costs by approximately 50%.

### 3.4. Graph Schema and File Format

Each dataset row includes the following: *video_id*, *frame_number*, and *category* (YouCook2 identifiers), *image* (keyframe, PIL), *graph* (dictionary with *graph.nodes* and *graph.edges*), and *graph_sentence* (text caption). Nodes contain *id* (short unique string, e.g., *pan1*), *label* (e.g., "cutting board"), *attributes* (array with objects such as *appearance* [string|null] and *size* [small|medium|large|null]; always present, though fields may be null), and and *location* (stored as the comma-separated string "`x1,y1,x2,y2,conf`", where $(x_1, y_1)$ is top-left and $(x_2, y_2)$ is bottom-right; all values are in $[0, 1]$ and $conf \in [0, 1]$). Edges include *source* and *target* (node IDs) and a *predicate* (natural language relation in snake_case).

### 3.5. Graph-Sentences

For each frame, we generate a textual paraphrase summarising the corresponding graph. Using the same teacher LLM, we prompt: *"Based on the following graph representation of a frame, generate a caption describing the scene. Output only the caption."* This produces concise, semantically faithful captions without using the image itself, enabling cross-modal alignment and instruction tuning.

### 3.6. Splits

Data are split by *video_id* to prevent leakage between frames of the same video. We provide standard training, validation, and test partitions. Each split is defined by non-overlapping video IDs to avoid temporal or content leakage across training and evaluation. The *validation_dev* split is a 100-frame subset of the validation split used for debugging/selection.

| Split | Videos | Frames |
|---|---|---|
| Training | 117 | 12,512 |
| Validation | 25 | 3,357 |
| Validation_dev | 25 | 100 |
| Test | 25 | 2,849 |

Table 1: Frame2KG-YC2 dataset splits.

### 3.7. Dataset Statistics

The training split contains 12,512 frames from 117 videos (mean 106.9 frames/video; median 92; max 296), totalling 92,672 nodes and 82,027 edges. On average, each frame includes 7.4 nodes (median 7; max 31) and 6.6 edges (median 6; max 30), reflecting moderate relational density with occasional empty scenes. Per video, counts average 792 nodes and 701 edges. Table 2 lists the five most frequent object and predicate categories, highlighting the cooking-centric bias inherited from YouCook2. An extended breakdown of the top-10 labels and predicates is provided in Appendix C.

| Top-5 node labels (*training*) | | Top-5 predicates (*training*) | |
|---|---|---|---|
| Label | Count | Predicate | Count |
| countertop | 2753 | on | 34046 |
| cutting board | 1919 | in | 7724 |
| hand | 1898 | next_to | 7665 |
| person | 1654 | holding | 7211 |
| man | 1566 | in_front_of | 5571 |

Table 2: Most frequent node labels and predicates.

### 3.8. Quality Control

Before release, each sample must parse and conform to a strict JSON schema. In practice, the only systematic failure observed was output truncation due to an insufficient maximum token budget; affected items were re-generated with a higher budget and re-validated.

### 3.9. Release and reproducibility

We release the Frame2KG-YC2 dataset on Hugging Face[4], including per-frame graphs, *graph_sentence*, split files, and metadata. A central documentation hub[1] links to the dataset, the Frame2KG-Trainer, the Frame2KG Evaluation Toolkit (Sec. 5), and the released LoRA adapters (Qwen2.5-VL backbones).

To support traceable reruns, each released adapter is accompanied by its exact `frame2kg-trainer` YAML configuration (stored alongside the weights), and the training configuration includes a `training_repo` field that points to the code repository used for the run (or fork when any breaking changes are required). This mirrors prior evidence that reproduction attempts often fail due to missing or unobtainable experimental artefacts and configuration details (Belz et al., 2023). We encourage future work building on Frame2KG to follow the same practice by publishing (i) the complete YAML used for training, and (ii) a `training_repo` reference to the exact codebase used, to keep results auditable and comparable.

[4] https://huggingface.co/datasets/lewiswatson/Frame2KG-YC2

## 4. Models & Training

We fine-tune Qwen2.5-VL (3B and 7B "Instruct") for Frame→KG using parameter-efficient adapters in a lightweight, reproducible setup with Frame2KG-Trainer[5]. The aim is schema-valid, grounded graphs with strong efficiency–accuracy trade-offs.

### 4.1. Backbone and Adapter Variants

We adopt **Qwen2.5-VL-3B-Instruct** and **Qwen2.5-VL-7B-Instruct** as the base multimodal backbones, both used with their official visual processor and tokenizer. To examine the effects of adapter placement and model capacity, we train four LoRA-based variants: (1) **3B–QKVO**, which applies adapters to $\{q_{\text{proj}}, k_{\text{proj}}, v_{\text{proj}}, o_{\text{proj}}\}$; (2) **3B–QKVO–Gate**, which extends this configuration with additional adapters on MLP layers (`Gate_Proj`, `Up_Proj`, and `Down_Proj`); (3) **7B–QKVO**, a larger-capacity version of the base configuration; and (4) **7B–QKVO–Gate**, combining both increased backbone scale and extended adapter coverage. All adapters use rank $r$=8, scaling factor $\alpha$=16, and dropout $p$=0.05. Backbone weights remain frozen during fine-tuning, with only adapter parameters optimised. Gradient checkpointing is enabled to reduce activation memory and ensure efficient training under constrained GPU resources.

### 4.2. Data formatting and supervision

We train all models on the **Frame2KG-YC2** corpus introduced in Section 3, using the pre-defined training split as provided on the Hugging Face dataset hub. The data are shuffled at the beginning of each epoch to prevent ordering bias. Each example consists of a single video frame paired with its corresponding schema-valid graph representation, ensuring consistent structure across training samples.

Each training instance is formatted as a **single-turn chat** comprising three roles: (1) a **System** message providing a concise instruction that enforces strict JSON compliance and describes the expected schema; (2) a **User** message containing the RGB frame; and (3) an **Assistant** message representing the gold knowledge graph, serialised as a single JSON object. Prompts are rendered using Qwen2.5-VL's *apply_chat_template* collator, which inserts the appropriate multimodal tokens. During optimisation, only the assistant segment contributes to the loss, while system/user context and padding tokens are masked out.

[5] https://github.com/lewiswatson55/frame2kg-trainer

Training optimises the standard next-token cross-entropy loss, computed exclusively over the assistant segment to focus learning on structured graph generation. During evaluation, the model produces the assistant response within the same chat template, from which we extract the first syntactically valid JSON object containing both *nodes* and *edges*. This approach ensures robustness to any spurious or trailing text occasionally emitted by the model. All generated outputs are subsequently validated against the strict Frame2KG schema prior to metric computation, guaranteeing that evaluation reflects only structurally and semantically valid graphs.

### 4.3. Optimisation and hyperparameters

All models are fine-tuned using the AdamW optimiser (Transformers default) with a base learning rate of $5 \times 10^{-5}$, weight decay of $0.01$, and a linear warmup schedule (`warmup_ratio`=$0.03$). Training is performed for three epochs using full-sequence teacher forcing. To ensure reproducibility, we fix `seed`=$42$ across the model, data loader, and NumPy random number generators. Each run uses a per-device batch size of 8 for training and 2 for evaluation, with `gradient_accumulation_steps`=$1$. The tokenizer employs left padding to maximise prefill efficiency. During evaluation, the generation length is capped at `generation_max_new_tokens`=$4098$ to prevent truncation.

Adapters modify only a small proportion of the overall parameters relative to the frozen backbone. The trainable-to-total parameter ratios are automatically logged at run initialisation, providing a transparent measure of computational efficiency. Detailed trainable parameter counts for each adapter configuration are summarised in Table 3.

| Model | Trainable | Total | Trainable (%) |
|---|---|---|---|
| 3B-QKVO | 3,686,400 | 3,758,309,376 | 0.098 |
| 3B-QKVO_Gate | 18,576,384 | 3,773,199,360 | 0.492 |
| 7B-QKVO | 5,046,272 | 8,297,212,928 | 0.061 |
| 7B-QKVO_Gate | 23,794,688 | 8,315,961,344 | 0.286 |

Table 3: Trainable parameter counts and percentages for adapter variants.

### 4.4. Checkpointing, validation, and model selection

**Online evaluation.** Full-generation evaluation is too slow to run continuously during training, so we do not perform online evaluation every few steps. Instead, selection is performed by scoring the four persisted checkpoints (see below) on the *validation_dev* slice. At offline scoring time we generate predictions, parse the first valid JSON, and compute the deterministic metrics using the toolkit. See section 5.

### 4.5. Checkpoint persistence

We persist adapter checkpoints every 100 global steps and retain these intermediate snapshots for post-hoc analysis. For downstream selection, once trained, we carry forward four representative checkpoints per variant and score them offline:

1. the first plateau point ("stabilised loss"), typically global step $\approx 1{,}000$;
2. a mid-run snapshot, $\approx 2{,}000$ steps;
3. the checkpoint with the lowest training loss;
4. the final checkpoint at the end of epoch 3.

These four candidates are generated and evaluated on the *validation_dev* slice, the offline scoring procedure determines which checkpoint is selected for full test evaluation and release.

**Checkpoint selection:** Cross-entropy is a weak proxy for structured graph quality. Instead we rank checkpoints by Frame2KG metrics (See section 5) on *validation_dev* (n=100) in the order: Node $F1_{\mu} \rightarrow$ Edge $F1_{\mu} \rightarrow$ Box IoU $\rightarrow$ validity/conformity. Per-variant winners were: `3B/QKVO` - *final*; `3B/QKVO_Gate` - *final*; `7B/QKVO` - *best*; `7B/QKVO_Gate` - *step1k*. The full raw outputs are available in the Frame2KG repo[1]. Full tables are in Appendix A.

*Note that all experiments were conducted on NVIDIA A100 GPUs (40 GB/80 GB) hosted on Google Cloud Platform, using Hugging Face Transformers and PEFT.

### 4.6. Reproducibility and release

We release the artefacts required to reproduce and deploy our results as part of the Frame2KG-Trainer and Frame2KG Evaluation Toolkit (Sec. 5): (a) four LoRA adapter checkpoints (3B/7B $\times$ QKVO/QKVO+MLP), with the best-per-variant adapter selected from *validation_dev* provided as the canonical release; (b) the exact training configurations and scripts (single-entry CLI) used to run and reproduce experiments; and (c) utilities to merge adapters into base weights and to quantise merged models for edge deployment. No data augmentation or additional instruction tuning beyond the Frame2KG chat template was applied. The Frame2KG-Trainer pipeline is backend-pluggable and currently implements a Qwen2.5-VL backend. We invite future work to reproduce results or train domain-specific adapters with minimal changes (e.g. *model_id*, LoRA targets, schedule). The complete frozen evaluation configuration used for our comparisons is detailed in Appendix D. To our knowledge, these are the first open Frame$\rightarrow$KG LoRA baselines on Qwen2.5-VL.

## 5. Frame2KG Evaluation Toolkit

### 5.1. Design goals

The *frame2kg-eval* toolkit is designed to: (i) isolate correctness from formatting issues; (ii) prioritise localisation when aligning entities; and (iii) support deterministic, reproducible comparisons across variants, sizes, and deployment settings. The implementation is modular (I/O adapters, matching, metrics, CLI) and covered by unit and integration tests for matching, IoU, metrics, schema normalisation, and CLI paths.

### 5.2. Deterministic pipeline and I/O.

We fix the evaluation seed across Python, NumPy, and Torch, and use deterministic tie-handling where applicable. The evaluation assumes that the predictions have been precomputed and stored in a local directory using the stable filename convention:

`<video_id>.<frame_no>.(json|raw.txt)`

Files with `.raw.txt` are treated as invalid predictions. Ground truth may be provided via a Hugging Face adapter (e.g., `hf:dataset:split`) or a local JSON directory. All graphs are passed through a common normalisation layer:

- **Nodes:** required fields are *id*, *label*, *attributes*, and *location* with *confidence*. The *location* field must be a normalised 5-tuple $[x_1, y_1, x_2, y_2, \text{conf}]$, and all numeric values in $[0, 1]$.
- **Edges:** required fields are *source*, *target*, and *predicate* (strings).

### 5.3. Validity and conformity checks

Before scoring, each prediction is subjected to two checks: (i) *JSON validity* - the file parses as JSON and contains a minimal graph structure; and (ii) *schema conformity* - required fields are present and correctly typed (for example, node *location* must be a normalised 5-tuple and numeric values must lie in $[0, 1]$ with $x_1 < x_2,\ y_1 < y_2$). We report both the overall conformity rate and the conformity rate restricted to valid JSON files.

### 5.4. Two-stage node matching

Node alignment is one-to-one and proceeds in two stages: a spatial gate followed by spatial–semantic scoring and Hungarian assignment.

**Stage 1 - IoU gate.** Compute the IoU matrix between predicted and ground-truth boxes, and discard pairs with $\text{IoU} < \tau$. In all experiments we fix $\tau = 0.3$.

**Stage 2 - spatial–semantic scoring ($\alpha$-blend).** For surviving pairs, compute a text-similarity matrix using a sentence-transformer encoder (default: *all-MiniLM-L6-v2*). Node text is formed from *label*, and includes attribute values (ie. appearance, size, colour) where available. Labels are normalised by dropping trailing digits; strings are lowercased, punctuation is removed, and whitespace collapsed. The blended score is:

$$s(i,j) = \alpha \cdot \mathrm{IoU}(i,j) + (1-\alpha) \cdot \mathrm{sim}(i,j),$$

with $\alpha = 0.7$. A semantic floor (default 0.25) suppresses pairs with no textual evidence. One-to-one assignment is solved with the Hungarian algorithm on the negated score matrix; masked pairs are set to $-\infty$ so they cannot be selected. The text backend may be *semantic*, *tfidf*, or *hybrid*; for our experimentation we use *semantic*, with caching of GT embeddings per frame to improve efficiency.

The matcher provides: a mapping of predicted-node to GT-node indices; unmatched prediction and GT node sets; and matrices for IoU, text similarity, and blended scores for diagnostics.

### 5.5. Predicate (edge) matching

After node alignment, each predicted edge is compared to the ground truth under a simple normalised string equality. A predicted edge, defined by its *source*, *predicate*, and *target* fields, counts as a true positive if and only if (i) both of its endpoint nodes map to the corresponding ground-truth nodes under the established node assignment, and (ii) its predicate label matches after normalisation. The normalisation step lowercases all text, replaces hyphens and underscores with spaces, trims whitespace, and removes non-alphanumeric punctuation. We deliberately avoid any embedding-based or semantic matching. Edges are treated as directed *(source, predicate, target)* is not equivalent to *(target, predicate, source)*. All remaining predicted edges are counted as false positives, and all unmatched ground-truth edges as false negatives. For now we do not perform edge "lifting" (no label-only credit or multi-edge remapping).

### 5.6. Outputs and diagnostics

The toolkit produces deterministic scalar metrics (node/edge precision, recall, F1; box IoU statistics; validity and conformity rates) and CSV outputs. All evaluation steps are reproducible and auditable from the saved CSVs and prediction directories, enabling straightforward reruns and alternative ranking heuristics.

| Metric | Description |
|---|---|
| Node PRF1 (micro / macro) | TP from the one-to-one node mapping; FP/FN from unmatched predictions / GT. Macro = mean of per-frame PRF1; micro = aggregate TP/FP/FN across frames then compute PRF1. |
| Edge PRF1 (micro / macro) | Analogous to Node PRF1 but computed only for edges whose endpoints are matched to the correct GT nodes. |
| Matched-pair box IoU | Per-frame statistics (mean, median, std, min, max), matched-pair count, and coverage at $\mathrm{IoU} \geq 0.50$ and $\mathrm{IoU} \geq 0.75$. Micro averages weight by matched-pair counts; macro averages are the mean of per-frame means. |
| Validity & conformity rates | Overall JSON validity and schema-conformity rates; also report conformity restricted to valid JSON files. |

Table 4: Metrics produced by *frame2kg-eval*.

## 6. Experiment

### 6.1. Experimental setup

We evaluate the four LoRA adapter variants introduced in Section 4 - **3B/7B–QKVO** and **3B/7B–QKVO–Gate** - to examine both accuracy and deployability trade-offs. Each model is fine-tuned on the Frame2KG-YC2 training split and assessed on the held-out test set using the deterministic *Frame2KG Evaluation Toolkit* (Section 5). Evaluation follows the protocol defined in Section 6.2, which provides consistent handling of structural validity, grounding quality, and performance metrics.

To assess deployment feasibility, we further perform a post-training quantisation study on the top-performing model, applying both INT8 and INT4 quantisation and re-evaluating absolute and relative performance changes. This setup allows us to determine whether structured accuracy is maintained under hardware-efficient inference. Throughput is measured when a `manifest.csv` is available, reporting mean and median generation time as well as processed graphs per second.

All training configurations, selection scripts, evaluation utilities, and per-run logs are versioned and publicly released[1], ensuring full reproducibility and enabling external verification or alternative ranking heuristics under identical experimental conditions.

### 6.2. Metrics & diagnostics

All evaluations are deterministic and reproducible. The toolkit reports quantitative metrics for accuracy and efficiency, together with optional diagnostics for error analysis.

**Core metrics** For each evaluated model, we compute *node-* and *edge-level precision*, *recall*, and *F1 scores* (*micro* and *macro*), matched-pair intersection-over-union (IoU) statistics (*mean*, *median*, and *IoU*@$0.50/0.75$), JSON validity, and schema conformity. Frames with missing or invalid predictions are treated as empty outputs and included in both micro and macro aggregations (`include_invalid=True`) unless strict mode is

enabled. Under `strict_mode=True`, empty predictions are penalised by assigning false positives equal to the ground-truth support. The main benchmark uses `strict_mode=False`.

**Throughput and efficiency** Throughput is measured only when a `manifest.csv` is available. The evaluator reads `gen_wall_time_s` and `num_new_tokens` (or `decode_tps`) from batch-inference logs (e.g., `batch_infer.py`) and reports mean and median generation time, together with processed graphs per second. These measurements are used solely for profiling and have no effect on accuracy or model ranking.

**Compositional error diagnostics** An optional spatial–semantic grouping diagnostic provides insight into false negatives and false positives arising from compositional or compound labels (e.g., *fruit_basket* vs. individual fruits). This diagnostic clusters predictions by spatial proximity and text similarity, estimates how many errors are "explained" by composition, and reports these proportions separately. It serves as an interpretive aid and does not modify the reported PRF1 values.

**CLI** The evaluation toolkit provides four primary entry points: *frame2kg-eval* for single-run scoring and per-frame summaries; *frame2kg-sweep* for $\tau/\alpha$ sensitivity sweeps; *frame2kg-aggregate* for batch aggregation across runs; and *frame2kg-doctor* for integrity and sanity checks. All default settings and frozen configurations used in our comparisons are documented in Appendix D.

## 6.3. Results

Table 5 outlines the full test-set results of the four LoRA adapters, revealing three consistent trends across entity, relation, and localisation metrics.

#### 6.3.1. Entity and relation performance.

Model capacity and adapter design affect entities and relations in complementary ways. The **7B–QKVO–Gate** model achieves the highest Node F1 (micro $0.621$; macro $0.666$), reflecting stronger open-vocabulary entity naming and disambiguation. In contrast, the lighter **3B–QKVO–Gate** variant performs best on relational accuracy (Edge F1 $0.208$) and spatial grounding (mean IoU $0.609$; IoU coverage $0.68/0.27$ at thresholds $0.50/0.75$). This suggests that larger backbones primarily enhance semantic precision, while MLP-side gating (GateProj/Up/Down) benefits relational structure and geometry. Ablation results (detailed in Appendix B), available in our public repository[1], confirm that gated adapters consistently improve predicate and localisation metrics across both model scales with minimal additional parameters.

#### 6.3.2. Schema reliability and stability.

All models exhibit high structural reliability, with JSON validity and schema conformity exceeding $98\%$ (Table 5). This consistency is essential for robotic deployment, where structured outputs must be syntactically valid and executable. The narrow macro-micro gaps (e.g., Node F1 $0.597 - 0.621$ vs. $0.644 - 0.666$) indicate stable behaviour across frames rather than overfitting to specific scenes, reinforcing the general robustness of LoRA-tuned adapters.

#### 6.3.3. Model and evaluation selection.

Performance variation across checkpoints highlights the importance of structured selection criteria. The optimal **7B–QKVO–Gate** checkpoint was reached at step $1k$, preceding the final epoch, showing that evaluation guided by structural metrics (Sec. 5) can outperform loss-based selection. Similarly, evaluator configuration impacts results: IoU-gated spatial–semantic matching ($\tau = 0.3$, $\alpha = 0.7$) proved robust to $\alpha \in [0.6, 0.8]$ (see Appendix Table 10), while edge-level semantic matching increased runtime by 80% yet reduced Edge F1 ($-0.01$). Consequently, we adopt normalised string equality for efficiency and consistency, providing the semantic variant for future, linguistically diverse datasets. Composite-node diagnostics (see Appendix Table 11) confirm that split/merge effects explain only a minor fraction of errors ($\leq 5.5\%$ FPs, $\leq 2.7\%$ FNs), validating the current node granularity.

#### 6.3.4. Deployment efficiency.

Deployment-oriented measurements show that post-training quantisation preserves accuracy within normal variance (see Appendix G for full throughput and accuracy deltas). These results, together with the released scripts, demonstrate that *on-device Frame2KG inference* is both feasible and reliable, enabling efficient structured perception under real-world resource constraints.

# 7. Discussion

Our goal is to achieve efficient, accurate and interpretable scene understanding for use with embodied agents. The **Frame2KG-YC2** dataset, LoRA adapters, and evaluation toolkit were co-designed to advance this objective, emphasising both reproducibility and deployability.

## 7.1. Interpretable structured outputs:

We prioritise structured, schema-valid graphs that link grounded visual entities to their spatial coordinates. This "localisation-first" approach reflects

| Size | Variant | Node F1$_\mu$ | Edge F1$_\mu$ | IoU (mean) | Cov. (0.50 / 0.75) | Valid / Conf. (%) | Node F1$_{\text{macro}}$ |
|---|---|---|---|---|---|---|---|
| 3B | QKVO | 0.597 | 0.187 | 0.600 | 0.66 / 0.25 | 98.60 / 98.07 | 0.644 |
| 3B | QKVO-Gate | 0.616 | **0.208** | **0.609** | **0.68** / **0.27** | 99.19 / **98.84** | 0.661 |
| 7B | QKVO | 0.607 | 0.195 | 0.601 | 0.66 / 0.26 | **99.40** / 98.28 | 0.653 |
| 7B | QKVO-Gate | **0.621** | 0.204 | 0.605 | 0.67 / 0.26 | 99.16 / 98.56 | **0.666** |

Table 5: Final test results. Coverage (Cov.) is the fraction of matched pairs with IoU $\geq$ 0.50 / 0.75. "Valid / Conf." are JSON validity and schema conformity. Best values per column are **bold**.

the operational reality of robots, which must act on positions, not just descriptions. By explicitly reporting *validity* and *schema conformity*, our evaluation disentangles structural correctness from semantic accuracy. The observed high conformity rate ($> 98\%$) demonstrates that LoRA-tuned vision–language models can reliably generate executable, machine-interpretable outputs.

### 7.2. Compact efficiency and scalability:

Parameter-efficient fine-tuning (PEFT) reduces trainable weights to below $0.5\%$ of the full model (Table 3), yielding compact Frame2KG adapters that preserve accuracy while remaining lightweight. The 3B–QKVO-Gate variant significantly narrows the gap to its 7B counterpart, improving edge and localisation metrics while maintaining interpretability. Post-training INT4 quantisation further enables on-device inference with minimal degradation on *val_dev* (Node F1: $0.664 \rightarrow 0.658$). Our focus is not leader-board performance but *local feasibility*, achieving accurate, efficient, controllable scene graphs suitable for embedded or privacy-sensitive deployments.

### 7.3. The role of gating in fusion:

Adding MLP GateProj/Up/Down projections systematically strengthens cross-modal alignment. In the 3B configuration, gating raised Edge F1$_\mu$ from $0.187$ to $0.208$ and mean IoU from $0.600$ to $0.609$, confirming that gating mechanisms help integrate visual and linguistic features more effectively. By contrast, node naming, being semantically richer, benefits more from backbone scale and pre-training, explaining the consistent 7B advantage on Node F1.

### 7.4. Evaluation methodology:

Our deterministic two-stage node matching first aligns bounding boxes via IoU gating, then applies lightweight semantic matching with Hungarian assignment. For edges, we use normalised predicate equality, as embedding-based matching nearly doubled compute without improving accuracy under our predicate distribution. This keeps evaluation transparent, replicable, and adaptable to more lexically diverse domains.

### 7.5. Deployment preference and trade-offs:

Although the 7B/QKVO-Gate configuration achieves the highest Node F1$_\mu$ (0.621), the 3B/QKVO-Gate model performs nearly on par (0.616; $\Delta = 0.005$) while outperforming or matching it on edges (0.208 vs. 0.204) and localisation (IoU: 0.609 vs. 0.605) with far lower memory requirements. We therefore recommend the 3B/Gate variant as the preferred deployment option, balancing accuracy, interpretability, and efficiency.

## 8. Conclusion and Future Work

We introduced **Frame2KG**, an open synthetic dataset and pipeline for frame-to-knowledge-graph generation, with four reproducible LoRA adapters based on Qwen2.5-VL (3B/7B; QKVO with and without GateProj/Up/Down). A deterministic evaluation toolkit disentangles validity, localisation, and semantic correctness. Results show gating improves predicate accuracy and spatial grounding, while larger models enhance entity naming. With $> 98\%$ schema conformity and robust entity–relation matching, the framework demonstrates practical structured scene graph generation, with post-training quantisation enabling efficient on-device deployment.

A primary next step is extending Frame2KG beyond the cooking domain by expanding dataset and schema coverage to new environments and evaluating cross-domain robustness under the same deterministic protocol. In parallel, we will extend per-frame graphs into temporal knowledge graphs with persistent identities, event-level edges, and calibrated uncertainty. We also plan to integrate multi-camera views and 3D reconstruction for metrically consistent, spatially grounded graphs.

On the generation and deployment side, we explore speculative decoding (e.g., draft-verify (Leviathan et al., 2023)) to accelerate inference, quantisation-aware tuning for low bit-width efficiency, and schema-constrained decoding to maintain structural validity, alongside backbone comparisons under a shared training and evaluation protocol.

## Ethics Statement

The Frame2KG-YC2 dataset is derived from the YouCook2 corpus (Zhou et al., 2018) in accordance with its non-commercial research licence. Our release includes extracted frames and newly generated structured annotations. All graphs and accompanying *graph_sentence* are produced synthetically by our pipeline and distributed under the licence provided in our repository. To ensure compliance, we release only adapter weights, not the proprietary Qwen2.5-VL base model.

Some frames depict individuals in domestic settings, but the dataset contains no identity or biometric information. Outputs are schema-constrained graphs or brief textual paraphrases, avoiding free-form content that could compromise privacy. Scripts are provided for users to regenerate annotations locally.

The dataset inherits the cooking-centric bias of YouCook2 and may generalise less effectively to other domains. As annotations are generated by a vision–language model, residual societal biases may persist. We document our pipeline, release validators, and encourage independent audits and counter-datasets. Finally, we discourage surveillance-related use and prioritise low-carbon, energy-efficient training through parameter-efficient fine-tuning (PEFT) and quantisation to promote responsible and sustainable AI research.

## Limitations

While the Frame2KG-YC2 benchmark establishes a reproducible foundation for structured vision–language research, several limitations remain that inform the scope and interpretation of our findings. These limitations reflect both the design choices of the dataset and the current state of multimodal modelling.

- **Language scope:** *Both the dataset and the sentence-level semantic embedding model used for node-matching are limited to English.* This language restriction may under-represent multilingual and culturally diverse visual contexts, potentially limiting generalisation to non-English instructions, captions, or object naming conventions.
- **Synthetic supervision:** All graph and caption annotations are generated by a teacher vision–language model (VLM). This synthetic supervision can propagate biases, omissions, or factual inaccuracies inherited from the teacher model.
- **Domain scope:** Frame2KG-YC2 is cooking-centric (YouCook2-derived), so cross-domain generalisation is not evaluated in this work.
- **Backbone coverage:** Results are reported only for Qwen2.5-VL backbones (3B and 7B); we do not claim that the same trade-offs necessarily hold across other VLM families.
- **Deployment variance:** Quantisation and runtime performance vary across hardware and software stacks. Throughput figures reported here were measured on NVIDIA A100 GPUs and are provided as indicative rather than universally transferable benchmarks; real-world performance may differ on smaller or embedded systems.

Overall, Frame2KG-YC2 represents an initial step toward structured, interpretable scene understanding for embodied AI. Its design prioritises transparency, schema validity, and reproducibility over coverage and complexity. Future extensions will aim to incorporate multilingual supervision, temporal grounding, and broader domain diversity to improve fairness, ecological validity, and general applicability.

## Acknowledgements

We thank the anonymous reviewers and the area chair for their constructive comments. This research is supported by the EPSRC, under grant number UKRI210 (LoCS project).

## References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can, not as i say: Grounding language in robotic affordances.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-vl technical report.

Anya Belz, Craig Thomson, Ehud Reiter, Gavin Abercrombie, Jose M. Alonso-Moral, Mohammad Arvan, Anouck Braggaar, Mark Cieliebak, Elizabeth Clark, Kees van Deemter, Tanvi Dinkar, Ondřej Dušek, Steffen Eger, Qixiang Fang, Mingqi Gao, Albert Gatt, Dimitra Gkatzia, Javier González-Corbelle, Dirk Hovy, Manuela Hürlimann, Takumi Ito, John D. Kelleher, Filip Klubicka, Emiel Krahmer, Huiyuan Lai, Chris van der Lee, Yiru Li, Saad Mahamood, Margot Mieskes, Emiel van Miltenburg, Pablo Mosteiro, Malvina Nissim, Natalie Parde, Ondřej Plátek, Verena Rieser, Jie Ruan, Joel Tetreault, Antonio Toral, Xiaojun Wan, Leo Wanner, Lewis Watson, and Diyi Yang. 2023. Missing information, unresponsive authors, experimental flaws: The impossibility of assessing the reproducibility of previous human evaluations in NLP. In *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023a. Rt-1: Robotics transformer for real-world control at scale. In *Proceedings of the Robotics: Science and Systems (RSS) workshop / Robotics Conference 2023*. Also preprint arXiv:2212.06817.

Anthony Brohan et al. 2023b. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research (PMLR)*, pages 2165–2183. PMLR.

Guikun Chen, Jin Li, and Wenguan Wang. 2024. Scene graph generation with role-playing large language models.

Open X-Embodiment Collaboration. 2023. Open x-embodiment: Robotic learning datasets and rt-x models. *CoRR*, abs/2310.08864.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*.

Zeeshan Hayder and Xuming He. 2024. Dsgg: Dense relation transformer for an end-to-end scene graph generation.

Tao He, Lianli Gao, Jingkuan Song, and Yuan-Fang Li. 2022. Towards open-vocabulary scene graph generation with prompt-based finetuning.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Siyuan Huang, Zhi-Qi Cheng, Karl Schmeckpeper, Shuran Zhang, Anima Anandkumar, Yuke Zhu, and Zoran Popović. 2023. Conceptgraphs: Open-vocabulary 3d scene graphs for language-grounded manipulation. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pages 1–16. PMLR.

Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. 2024. Egtr: Extracting graph from transformer for scene graph generation.

Yunseok Jang, Yale Song, Chris Dongjoo Kim, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2758–2766.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. 2020. Action genome: Actions as compositions of spatio-temporal scene graphs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10233–10244.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2018. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1369–1379.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*.

Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. 2024. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Julian Lorenz, Robin Schön, Katja Ludwig, and Rainer Lienhart. 2024. A review and efficient implementation of scene graph generation metrics. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2567–2575.

OpenAI. 2025. Openai o3 and o4-mini system card. Technical report, OpenAI. Accessed: Oct. 16, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 2020. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. In *Robotics: Science and Systems (RSS)*.

Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. 2017. Video visual relation detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, page 1300–1308, New York, NY, USA. Association for Computing Machinery.

Luowei Zhou, Nathan Louis, and Jason J. Corso. 2018. Weakly-supervised video object grounding from text by loss weighting and object interaction.

# Appendix

## A. Validation_dev winners and selection details

Table 6 provides the detailed performance metrics used to select the best checkpoint for each model variant. Because cross-entropy loss is an unreliable proxy for structured graph quality, models were evaluated on the 100-frame *validation_dev* split using our deterministic evaluation toolkit. The winning checkpoints detailed here were subsequently carried forward for the final test-set evaluation and public release.

| Model | Checkpoint | Node $F1_{\mu}$ | Edge $F1_{\mu}$ | Box IoU | IoU@0.5 | IoU@0.75 | Validity (%) | Conform. (%) | Node $F1_{M}$ |
|---|---|---|---|---|---|---|---|---|---|
| 3B/QKVO | final | 0.636 | 0.204 | 0.603 | 0.650 | 0.270 | 99 | 99 | 0.653 |
| 3B/QKVO_Gate | final | 0.658 | 0.236 | 0.613 | 0.690 | 0.250 | 100 | 100 | 0.667 |
| 7B/QKVO | best | 0.669 | 0.257 | 0.619 | 0.680 | 0.310 | 100 | 99 | 0.664 |
| 7B/QKVO_Gate | step1k | 0.664 | **0.290** | **0.620** | **0.720** | 0.280 | 99 | 99 | **0.680** |

Table 6: Validation_dev winners per family with key metrics (rounded). The full per-checkpoint CSV and prediction directories are archived with the release for reproducibility.

## B. Ablation: effect of adding MLP (GateProj/Up/Down)

This ablation isolates the performance impact of adding MLP gating projections (`Gate_Proj`, `Up_Proj`, `Down_Proj`) to the standard QKVO LoRA configuration. As shown in Table 7, extending the adapter coverage to these layers consistently improves relational accuracy (Edge F1) and spatial grounding (Box IoU) across both the 3B and 7B model scales with only a marginal increase in trainable parameters.

| Model | Node $F1_{\mu}$ | Edge $F1_{\mu}$ | IoU mean |
|---|---|---|---|
| 3B–QKVO | 0.597 | 0.187 | 0.600 |
| 3B–QKVO-Gate | 0.616 | **0.208** | **0.609** |
| 7B–QKVO | 0.607 | 0.195 | 0.601 |
| 7B–QKVO-Gate | **0.621** | 0.204 | 0.605 |

Table 7: Ablation: effect of adding MLP (+GateProj/Up/Down). Gate improves predicates and localisation; 7B helps nodes.

## C. Dataset statistics (training split)

To provide deeper insight into the composition of the Frame2KG-YC2 dataset, Table 8 details the top-10 most frequent node labels and relational predicates found within the training split. These distributions highlight the domain-specific, cooking-centric nature of the source data inherited from YouCook2.

| Top-10 node labels (*training*) | | Top-10 predicates (*training*) | |
|---|---|---|---|
| Label | Count | Predicate | Count |
| countertop | 2753 | on | 34046 |
| cutting board | 1919 | in | 7724 |
| hand | 1898 | next_to | 7665 |
| person | 1654 | holding | 7211 |
| man | 1566 | in_front_of | 5571 |
| stove | 1344 | left_of | 3242 |
| plate | 1340 | behind | 3005 |
| woman | 1332 | supports | 2788 |
| bowl | 1265 | includes | 2218 |
| frying pan | 1086 | right_of | 1951 |

Table 8: Most frequent node labels and predicates in the *training* split.

## D. Frozen frame2kg-eval default config

Reproducibility is a core focus of this benchmark. To ensure that future models can be evaluated under strictly identical conditions, Table 9 explicitly defines the frozen configuration parameters and default settings used by the `frame2kg-eval` toolkit throughout our experiments.

| **Setting** | **Default** | **Notes** |
|---|---|---|
| Node IoU gate $\tau$ | 0.3 | Stage-1 box gating |
| Blend weight $\alpha$ | 0.7 | IoU vs. text in Stage-2 |
| Text mode | `semantic` | Options: `tfidf`, `semantic`, `hybrid` |
| Text fields | `labels, attributes` | Used to form node text |
| Text floor | 0.25 | Min cosine sim to count |
| Sentence encoder | `all-MiniLM-L6-v2` | `sentence-transformers/` model |
| Predicate mode | `normalised` | String eq. after normalisation |
| Semantic predicate $\theta$ | 0.6 | Not used |
| Include invalid | `true` | Invalid JSON counted (empties too) |
| Strict mode | `false` | If true, penalises invalids as FPs |
| Aggregation | `micro, macro` | Both are reported |
| Dataset | `lewiswatson/Frame2KG-YC2` | HF dataset name |
| Default split | `validation_dev` | For selection/sweeps |
| Output format | `csv` | Deterministic summaries + per-frame |
| Verbose | `true` | Detailed logs enabled |
| Sweep $\tau$ | `[0.3, 0.5, 0.7]` | Matcher sensitivity |
| Sweep $\alpha$ | `[0.5, 0.7, 0.85]` | Matcher sensitivity |

Table 9: Frozen `frame2kg-eval` config used for the experimentation in this paper.

## E. Matcher sensitivity to IoU gate and blend weight

We present the results of a hyperparameter sweep over the IoU gate threshold ($\tau$) and the spatial-semantic blend weight ($\alpha$) used during the two-stage node matching process. The results in Table 10 illustrate the robustness of our chosen default parameters ($\tau = 0.3$, $\alpha = 0.7$), showing stable performance across minor variations in the matching criteria.

| Model | $\tau/\alpha$ | Node F1$_\mu$ | Edge F1$_\mu$ | Comb. F1 |
|---|---|---|---|---|
| **3B–QKVO-Gate (final)** | **0.3/0.5** | **0.668** | **0.236** | **0.452** |
| | **0.3/0.7** | **0.668** | 0.234 | 0.451 |
| | 0.3/0.85 | 0.668 | 0.226 | 0.447 |
| | 0.5/0.5 | 0.479 | 0.129 | 0.304 |
| | 0.5/0.7 | 0.479 | 0.127 | 0.303 |
| **7B–QKVO-Gate (step1k)** | **0.3/0.5** | **0.674** | **0.282** | **0.478** |
| | **0.3/0.7** | **0.674** | **0.282** | **0.478** |
| | 0.3/0.85 | 0.674 | 0.280 | 0.477 |
| | 0.5/0.5 | 0.489 | 0.189 | 0.339 |
| | 0.5/0.7 | 0.489 | 0.189 | 0.339 |

Table 10: Matcher sensitivity on *val_dev*. $\tau$ is the IoU gate for stage-1 node alignment; $\alpha$ is the blend weight between localisation and text similarity in stage 2. Top-5 settings per model from the default sweep $\tau \in \{0.3, 0.5, 0.7\}$, $\alpha \in \{0.5, 0.7, 0.85\}$. Tied bests are bolded. All other evaluation settings remain frozen.

## F. Composite-node diagnostic

Table 11 details the results of our composite-node diagnostic on the *validation_dev* split. This diagnostic analyses structural errors stemming from label granularity, such as predicting a single merged entity instead of its constituent parts (splits), or vice versa (merges). The relatively low occurrence of these errors supports the node-level granularity adopted in our dataset schema.

| Model | Split (pred→GT) FN | Merge (GT→pred) FP |
|---|---|---|
| 3B–QKVO-Gate (final) | 2.7% (6/221) | 5.5% (10/181) |
| 7B–QKVO-Gate (step1k) | 1.9% (4/212) | 2.1% (4/188) |

Table 11: Composite-node diagnostic on *val_dev* (100 frames). Splits count cases where multiple predicted nodes align to a single GT node (as % of node FNs). Merges count cases where multiple GT nodes align to a single predicted node (as % of node FPs).

## G. Quantisation Impact

To further assess the feasibility of deploying these models on constrained edge hardware, Table 12 provides a detailed breakdown of model performance and inference throughput under INT8 and INT4 post-training quantisation. The results demonstrate the trade-offs between memory footprint, generation speed, and the preservation of structured graph accuracy.

| Model | Format | Node $F1_\mu$ | Edge $F1_\mu$ | IoU mean | s/graph [× vs FP16] |
|---|---|---|---|---|---|
| 3B–QKVO-Gate | FP16 | 0.658 | 0.236 | 0.613 | 30.3 [1.00×] |
| | INT8 | 0.646 ($\Delta = -0.012$) | 0.254 ($\Delta = +0.018$) | 0.619 | 101.9 [0.30×] |
| | INT4 | 0.636 ($\Delta = -0.022$) | 0.232 ($\Delta = -0.004$) | 0.612 | 43.7 [0.69×] |
| 7B–QKVO-Gate | FP16 | 0.664 | 0.290 | 0.620 | 27.8 [1.00×] |
| | INT8 | 0.686 ($\Delta = +0.022$) | 0.287 ($\Delta = -0.003$) | 0.620 | 95.3 [0.29×] |
| | INT4 | 0.658 ($\Delta = -0.006$) | 0.278 ($\Delta = -0.012$) | 0.616 | 41.4 [0.67×] |

Table 12: Post-training quantisation on *val_dev* (same eval defaults: $\tau$=0.3, $\alpha$=0.7, semantic). Deltas ($\Delta$) are absolute changes vs the FP16 run for the same model and split. Speed is mean end-to-end wall time per frame from `manifest.csv`; bracket shows throughput relative to FP16.