Open Source LLM Speculation (InternVL2-8B) BBS Project

Kazi Islam

Version 1.0

Prepared for Prof David Smith

Date [10-23-2025]

Page of Content

1. Abstract

This document presents the exploratory phase of developing an open-source image-and-text recognition framework for the Balanced Blended Space (BBS) puppet show, focusing on evaluating InternVL2-8B as a candidate vision-language model. InternVL2-8B is an instruction-tuned multimodal model (~8.1B parameters) built from an InternViT-300M-448px visual encoder, an MLP projector, and the internlm2_5-7b-chat language backbone, released under the MIT license for flexible local deployment and fine-tuning. The model is trained for an 8K-token context with data that includes long text, multi-image inputs, and videos—improving handling of documents, charts, infographics, scene text/OCR, and video frames relative to prior InternVL releases. Our study investigates whether InternVL2-8B can learn from paired character/prop images with concise captions to recognize and label new photos in the BBS dataset. We examine architecture, licensing, fine-tuning feasibility (including 8-bit/4-bit quantization paths) and instruction-following behavior, and we assess suitability by probing responsiveness to visual prompts, text–image alignment accuracy, temporal consistency across multi-image/video inputs, and generalization to unseen stage scenes. We also consider indicative public benchmarks (e.g., strong DocVQA and OCRBench results and competitive performance versus proprietary systems) as signals of document understanding and perception quality relevant to BBS. Findings will inform a practical foundation for integrating open-weight vision-language models into artistic and educational media contexts, guiding a lightweight, reproducible recognition workflow tailored to BBS's storytelling environment.

2. Objective

2.1 Primary objective

Determine whether InternVL2-8B is a suitable, open-weight model for BBS image-and-text recognition, and select the deployment configuration (precision, context usage, batching, image resolution/frame sampling) that best balances accuracy, latency, and hardware cost.

2.1.2. Specific objectives

2.1.2.1. Model Selection

Tasks to test: Captioning, VQA (answer questions about an image), puppet/prop labels, multi-image prompts (and optional short frame stacks).

Try simple knobs:

- **Precision:** FP16/BF16 vs 8-/4-bit quantization
- **Image size:** ~448 px baseline vs higher if helpful
- **Context/batch:** short vs longer context; single vs small batches

Choose the setup that meets our quality bar while being the cheapest and fastest to run.

2.1.2.2. Cost & Resource Envelope

- **Record for each setup:** GPU/CPU model, VRAM/RAM, wall-clock per request, latency, and throughput (items/sec).
- **Check practicality:** Runs well on commodity GPUs/standard PCs a small team can afford (and note what happens on CPU-only if relevant).
- **Note easy wins:** Any built-in optimizations (e.g., paged attention, quantized loaders) that improve speed/cost without hurting quality.

2.1.2.3. Openness & Compliance

- **License & reuse:** Confirm the open-weight license and any usage/attribution rules on the model card for education/creative work.
- **Assets are available:** Weights, code, configs, and basic training/eval details are publicly accessible.
- **Fine-tune ready:** Provide a short step-by-step path for adapting InternVL2-8B to BBS data (LoRA/QLoRA or full FT) while staying license-compliant.

3. Background

InternVL2-8B is an open-weight vision-language model that reads images (including multiple images or pre-extracted frame sequences) together with text and replies in natural language. It pairs a high-capacity vision encoder with a chat-tuned language backbone, supports an 8K-token context, and is trained to handle long text, multi-image inputs, and video-like sequences. In offline workflows—where you upload a dataset of images instead of using a live camera—the model fits neatly into a batched, file-driven pipeline: you load images from disk, attach a task-specific prompt (captioning, VQA, puppet/prop classification, OCR), and record outputs per file. The release includes permissive open weights and straightforward inference/finetuning paths (FP16/BF16 with optional 8-/4-bit quantization), which helps run evaluations on commodity hardware.

For BBS, the offline setup is a strength: InternVL2-8B can label rehearsal or show photos, answer scene questions, and read signs or labels without any real-time constraints. Multi-image input lets you compare angles or consecutive frames for occluded props; you can also add simple multi-crop passes to capture tiny costume details or text. Because everything is disk-based, you can reproduce runs easily—fix image resolution, batch size, and prompts; log latency, throughput, and VRAM; and iterate on prompts or class lists without changing capture conditions. This dataset-first approach makes it practical to benchmark accuracy, tune costs (via quantization and batching), and prepare a clean path to lightweight fine-tuning on curated BBS images.

4. Evaluation of Model Prototypes

InternVL2-8B is distributed as an instruction-tuned multimodal checkpoint. For BBS, we evaluate three practical profiles of the same model that serve different needs.

4.1. internvl2-8b — “instruction-tuned default”

What it is: The standard Transformers/PyTorch checkpoint with ~8K context; accepts multi-image inputs and long text, trained on data that includes long text, multiple images, and videos. License is permissive (MIT for InternVL; InternLM component under Apache-2.0).
Best for: Out-of-the-box captioning, VQA/OCR, puppet/prop classification, and multi-image prompts.
Pros: Strong multimodal capabilities; clear quick-start docs; native BF16/FP16 paths.
Trade-offs: Needs a decent GPU for best latency; multi-image dialogue can be less stable—prompt carefully and consider retries.
Typical outputs: Direct, instruction-following responses suitable for pipelines.
Prompting tip: Specify schema/length (e.g., “Return JSON {puppets, props, scene}; ≤2 sentences”).
Choose this if: You want strong results now with minimal engineering.

4.2. internvl2-8b (quantized 8-/4-bit) — “memory-saver”

What it is: The same checkpoint runs with quantization to reduce VRAM/cost. Supports BNB 8-bit/4-bit in Transformers; also available as **AWQ INT4** builds via LMDeploy.
Best for: Edge/budget GPUs or higher throughput under tight memory.
Pros: Much lower memory; AWQ INT4 can deliver up to ~2.4× speedup vs FP16 on supported NVIDIA GPUs.
Trade-offs: Slight quality/latency variance vs full precision; require careful benchmarking and conservative decoding.
Typical outputs: Similar to default; occasionally shorter/safer generations.
Prompting tip: Keep temperature/top-p modest to stabilize outputs when quantized.
Choose this if: You need to fit within ~8–12 GB VRAM or share a single GPU among jobs.

4.3. internvl2-8b (deployment-optimized: LMDeploy/OpenVINO) — “portable serving”

What it is: Production-oriented runtimes for faster, portable inference:
LMDeploy: server + INT4 AWQ path (W4A16) and multi-GPU options.
OpenVINO: CPU-first optimization/compression for Intel hardware.
Best for: CPU-friendly services, containerized APIs, or mixed fleets (some GPU, many CPU).
Pros: Good CPU throughput; simple API server/Docker recipes.
Trade-offs: Conversion/ops compatibility steps; fewer bleeding-edge features than raw PyTorch.

Typical outputs: Same as default; tuned for stable serving.
Prompting tip: Use explicit schemas and pre-tokenize prompts for batching.
Choose this if: You want reliable, low-cost serving across heterogeneous hardware.

4.4. Shared traits (all three)

InternVL2-8B provides open weights with permissive licensing, supports interleaved images+text with text outputs, and is trained for an ~8K token window that improves handling of long text and multi-image/video inputs versus prior versions. BF16/FP16 and quantized paths are documented in official quick-starts.

4.4.1. Quick pick guide

Need plug-and-play quality now? → Instruction-tuned default.

Tight VRAM/budget or CPU-only? → Quantized memory-saver (BNB 8/4-bit or LMDeploy AWQ INT4).

Production serving on mixed hardware? → LMDeploy/OpenVINO deployment-optimized.

Recommendation for BBS: Start with InternVL2-8B in 8- or 4-bit for tagging, OCR, and short captions; if you need CPU-first or containerized rollout, convert to OpenVINO or serve via LMDeploy for a low-cost, reproducible pipeline.

5. References

1. OpenGVLab. “InternVL2-8B — Model Card.” Hugging Face, 2025. (Hugging Face)
2. Chen, Z., et al. “InternVL 2.5: Expanding Performance Boundaries of Open-Source Multimodal LLMs.” arXiv, 2024. (arXiv)
3. OpenGVLab. “InternVL — Official Repository.” GitHub, 2024–2025. (GitHub)
4. Shanghai AI Lab. “internlm2_5-7b-chat — Model Card (Backbone for InternVL2-8B).” Hugging Face, 2025. (Hugging Face)
5. InternVL Team. “InternVL 2.0 — Project Blog.” 2024. (internvl.github.io)
6. Chen, Z., et al. “InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks.” arXiv, 2023. (arXiv)
7. InternLM. “LMDeploy: Compressing, Deploying, and Serving LLMs/VLMs.” GitHub, 2024–2025. (GitHub)
8. OpenGVLab. “InternVL2-8B-AWQ — INT4 (W4A16) Quantization for LMDeploy.” Hugging Face, 2025. (Hugging Face)
9. Hugging Face Transformers. “Bitsandbytes 8-/4-bit Quantization Guide.” Docs, 2025. (Hugging Face)
10. Hugging Face Accelerate. “Model Quantization (bitsandbytes).” Docs, 2025. (Hugging Face)

11. OpenVINO™ Toolkit. “4-bit Weight Quantization (AWQ/GPTQ).” Docs, 2024–2025. (OpenVINO Documentation)
12. LMDeploy Docs. “Deploying InternVL (example: InternVL2-8B).” Read the Docs, 2025. (LMDeploy)
13. Liu, Y., et al. “OCRBench (and OCRBench-v2) — Benchmark for OCR-centric VLM Evaluation.” GitHub / arXiv, 2024–2025. (GitHub)

6. Version History

Version	Created on	Created by
1.0	10/23/2025	Kazi Islam
2.0	11/03/2025	Kazi Islam