Small yet Assistive: Spatially-Aware Post-Training for Low Vision

Rishabh Choudhary[1*], **Shreyansh Raj**[1*], **Umesh Goyal**[1]
Shubh Kashyap[1], Shrestha Kumar[1], Sushovan Jena[1*†]
Komal Kumar[2*], Hisham Cholakkal[2], Aditya Nigam[1]
[1]Indian Institute of Technology Mandi
[2] Mohamed bin Zayed University of Artificial Intelligence

GitHub: github.com/Shreyansh262/Small-Yet-Assistive

Website: smol-vl-blv.github.io/Smol-VL-BLV-website/

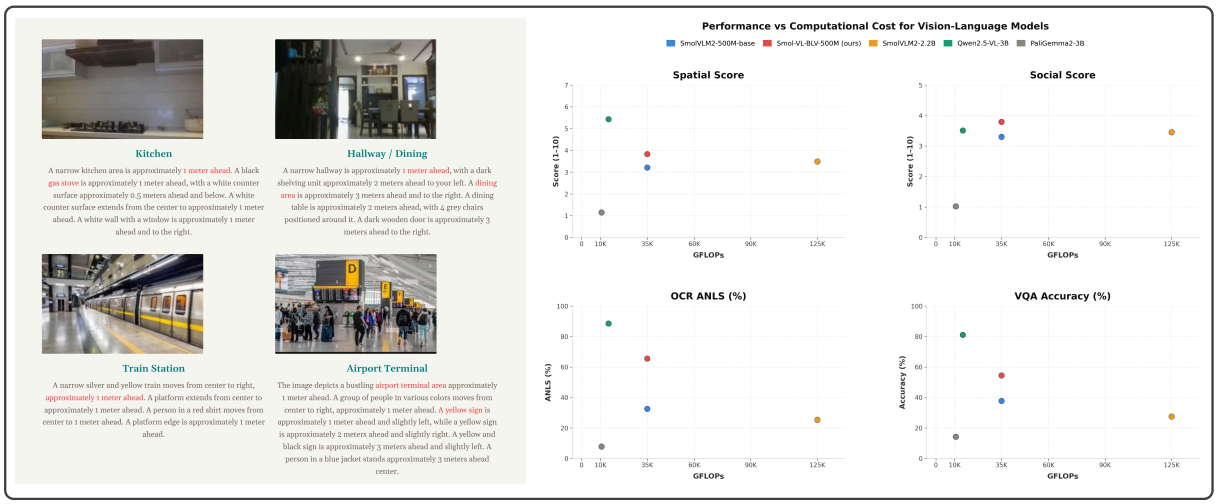

Figure 1: Overview of our on-device BLV-oriented vision-language model. Left: qualitative examples of generated scene descriptions across indoor, transit, and public-space settings. The highlighted phrases show BLV-critical information captured by the model, including distance estimates, directional cues, obstacles, people, signs, and spatial layout. Right: performance–efficiency comparison across VLMs. After spatial-grounding supervised fine-tuning and GRPO-based preference optimization, our SmolVLM-500M model improves spatial reasoning, social understanding, OCR, and VQA accuracy while remaining substantially cheaper than larger VLMs in computational cost. These results suggest that high-quality BLV-focused synthetic supervision from a stronger teacher model can induce emergent spatial grounding and more precise accessibility-oriented descriptions in compact VLMs.

Abstract

An estimated 1 billion people worldwide live with vision impairment, yet current vision-language models (VLMs) produce descriptions too vague for safe navigation by blind and low-vision (BLV) users. Large VLMs can generate high-quality audio-description-compliant narrations but cannot run on mobile devices; small VLMs offer competitive latency but lack spatial detail, directional cues, and hazard awareness for navigational assistance. We present Smol-VL-BLV, a compact VLM for blind and low-vision users that closes this gap using a 500M decoder transformer model and two post-training mechanisms: (1) teacher-student distillation and (2) Group Relative Policy Optimization (GRPO) with a composite BLV reward targeting directional language, metric distances, and hazard detection. Because multi-stage post-training can induce catastrophic forgetting, we add a lightweight finetuning stage after the last stage GRPO finetuning to recover general descriptive quality while preserving BLV-specific spatial grounding. Our best model substantially outperforms the baseline across various benchmarks, including tasks: VQA, BLV captioning, OCR, and latency. Compared with the baseline for relative improvement, it improves the Spatial score gain of 19.3%, and the Social score gain of 14.8%. It also increases OCR-Bench by 101.5%, and raises TextVQA accuracy by 44.2%. These results show that BLV-focused post-training improves both accessibility-specific spatial grounding and general visual-text reasoning. Deployed on a mid-range Android smartphone via Mixed-Precision Quantization, the model remains approx. 450 MB and runs entirely on-device, offline and without network de-

* Equal contribution. † Corresponding author: sushovanjena@gmail.com

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 51690–51703
October 24-29, 2026 ©2026 Association for Computational Linguistics

pendency, generating descriptions with latency dependent on host hardware capabilities. Our model, dataset, and code is publicly released at https://smol-vl-blv.github.io/Smol-VL-BLV-website/.

1 Introduction

The affected population worldwide with vision impairment is estimated around 1 billion which includes 94 million with cataracts and 88.4 million with uncorrected refractive errors (Steinmetz et al., 2021). While smartphone-based tools such as Apple VoiceOver and Google TalkBack effectively describe user interfaces, they fall short at describing real-world scenes with the spatial detail, directional cues, and hazard awareness that blind and low-vision (BLV) (Corn and Erin, 2010; Leat et al., 1999) users require for safe, independent navigation.

Professional audio description (AD) standards, established by the ITC and adopted by Netflix (López, 2023) and the ACB, prescribe a specific style of visual narration: present-tense, spatially grounded, hazard-first, and objective (Conway et al., 2020). A description following these guidelines might read: *"A corridor extends approximately eight meters ahead. A partially open door is on your left, three meters away. A person walks toward you from the far end."* Current vision-language models (VLMs), trained predominantly on internet-scale data written for sighted audiences (Wang et al., 2025), instead produce vague outputs such as *"A hallway with some doors and people"*—adequate for image captioning benchmarks but inadequate for physical navigation (Baghel et al., 2025).

Earlier BLV assistive technology research shows a shift from basic OCR and object recognition toward richer visual access, including video description, visual question answering, spatial exploration, customizable assistance, and conversational tools (Li et al., 2025; Cheema et al., 2025a,b, 2026; Xu et al., 2025, 2026; Li et al., 2026). In parallel, efficient vision-language model research has advanced compact architectures and mobile deployment through smaller VLMs, faster vision encoding, token reduction, and low-latency inference (Wang et al., 2020; Marafioti et al., 2025a; Chu et al., 2023; Vasu et al., 2025; Huang et al., 2025). However, these directions rarely combine real-time mobile capability with BLV-specific spatial grounding, hazard awareness, and concise navigation-oriented descriptions.

Prior work (Li et al., 2025; Baghel et al., 2025) benchmarked SmolVLM2-500M on a smartphone and established two domain-specific evaluation frameworks—the Multi-Context Framework (MCF) and the Navigational Assistance Framework (NAF): showing that the base model produces descriptions that are too vague and spatially unaware for BLV use. The present work closes this quality gap by developing a Smol-VL-BLV model, which consists mainly of two mechanisms: (1) **Teacher-student distillation.** We use Gemma-4-31B-IT as a teacher to generate AD-guideline-compliant captions for 9,646 videos from the Charades and AVCaps datasets. (2) **Reinforcement learning via GRPO.** We apply GRPO with a domain-specific reward targeting directional content, spatial language, and hazard mentions.

Beyond post-training, we perform a systematic deployment study targeting real-world mobile inference on a mid-range Android smartphone. Our final trained model is exported to GGUF format (Tripathi, 2024), producing a 783 MB float16 checkpoint. To compress this for on-device use, we apply IQ4_NL quantization guided by an importance matrix computed over our Gemma-generated BLV navigation captions: the same captions used during training. Within this scheme, the attention layers that govern where the model focuses spatially are kept at a higher precision (Q5_K) than the rest of the network, and the vision projector is held at float16 throughout. The result is a 442MB two-file deployment package: a 251MB quantized language model backbone and a 191MB vision projector: that runs via CPU-only inference on a Samsung A55, achieving 39.3 tok/s generation throughput and 27.1s end-to-end latency per frame.

We conduct comprehensive experiments using the Charades and AVCaps datasets, evaluating across keyword-based BLV and navigational scoring, standard NLP metrics, and LLM-as-judge assessment. Our model achieves strong results on spatial orientation, social interaction, navigational coverage, and general vision-language benchmarks, including OCR and VQA. Figure 1 illustrates qualitative examples and a cross-model performance comparison. Our main contributions are:

- A complete teacher-student pipeline for distilling BLV-compliant video description capability into a $< 1B$ VLM deployable on com-

blindness-and-visual-impairment

modity smartphones.

- A multi-stage post-training strategy combining SFT, GRPO, and patch-based refinement, improving BLV coverage, spatial grounding, navigational cues, and OCR/VQA transfer while yielding the best overall model across evaluation metrics.
- Comprehensive evaluation across BLV-specific metrics, OCR, VQA, and on-device benchmarking demonstrating both model quality and practical viability for real-world BLV assistance.

2 Related Work

BLV-Centered Visual Assistance. Assistive technologies for BLV users help them access visual information through OCR readers, object recognition tools, navigation aids, wearable cameras, screen readers, smartphone apps, and remote human assistance. However, BLV users do not all need the same kind of help. Their needs depend on the task, environment, level of vision, and type of information required (Corn and Erin, 2010; Leat et al., 1999; Steinmetz et al., 2021). Recent systems move beyond only detecting objects or reading text by supporting video description, visual question answering, spatial exploration, customizable descriptions, and conversational assistance (Li et al., 2025; Cheema et al., 2025a,b, 2026). Other works such as Branch Explorer, Sonic Stage, and AD-Canvas show the importance of spatial understanding, interaction, and user control (Xu et al., 2025, 2026; Li et al., 2026). Still, many systems focus on video/media access or authoring workflows rather than real-time scene understanding for navigation and hazard awareness. This motivates our focus on short, spatially grounded, and safety-aware BLV descriptions instead of generic captions.

Real-Time & On-Device Multimodal Systems. In the development of real-time and on-device multimodal systems, vision-language models are designed to work efficiently on mobile devices, edge devices, and other resource-constrained environments. This is especially important for BLV assistance, because delayed responses may be unsafe or unhelpful during navigation, obstacle detection, or hazard avoidance. Prior work on efficient VLMs has focused on reducing model size, improving vision encoders, compressing visual tokens, using speculative decoding, and applying quantization to reduce memory and computation cost (Chu et al., 2023; Vasu et al., 2025; Huang et al., 2025; Marafioti et al., 2025a). Recent work on lightweight VLMs for BLV accessibility also shows the need to evaluate compact models for accessibility-specific use cases rather than only on generic benchmarks (Baghel et al., 2025). However, most prior work mainly optimizes general-purpose VLM efficiency and does not directly address BLV-specific requirements such as when to warn users, when to express uncertainty, how to describe hazards, or how to provide concise spatial guidance. Our work builds on efficient lightweight VLM research, but specifically targets low-latency, on-device BLV visual assistance by adapting a compact model for spatially grounded and hazard-aware descriptions.

3 Methodology

Our pipeline comprises four stages: (1) data preparation—keyframe extraction and teacher caption generation; (2) supervised fine-tuning with phase-switching (SFTv2); (3) reinforcement learning via GRPO with a BLV-specific reward; and (4) SFT-patch recovery and mobile deployment. Figure2 provides an overview. The student model throughout is SmolVLM2-500M-Video-Instruct (Marafioti et al., 2025b): a SigLIP-based vision tower (86.4M), a multi-modal connector (11.8M), and a LLaMA-architecture (SmolLM2) language decoder with LM head (409.3M), totalling 507.5M parameters. Following the SmolVLM naming convention, "500M" is used throughout the paper as a colloquial class label rather than a literal parameter sum.

3.1 Data Preparation

Keyframe Extraction We extract $n{=}4$ keyframes per video using perceptual color-space differencing in LUV space. After subsampling to at most 64 evenly-spaced frames, we compute the mean absolute inter-frame difference:

$$d_i = \frac{1}{HW} \sum_{h,w} |\mathrm{LUV}(f_i)_{h,w} - \mathrm{LUV}(f_{i-1})_{h,w}| \tag{1}$$

The $n{-}1$ highest-d_i frames plus f_1 are selected and sorted chronologically, capturing scene transitions and significant motion events. LUV differencing requires no additional model inference—unlike DINOv2(Oquab et al., 2023) semantic embeddings

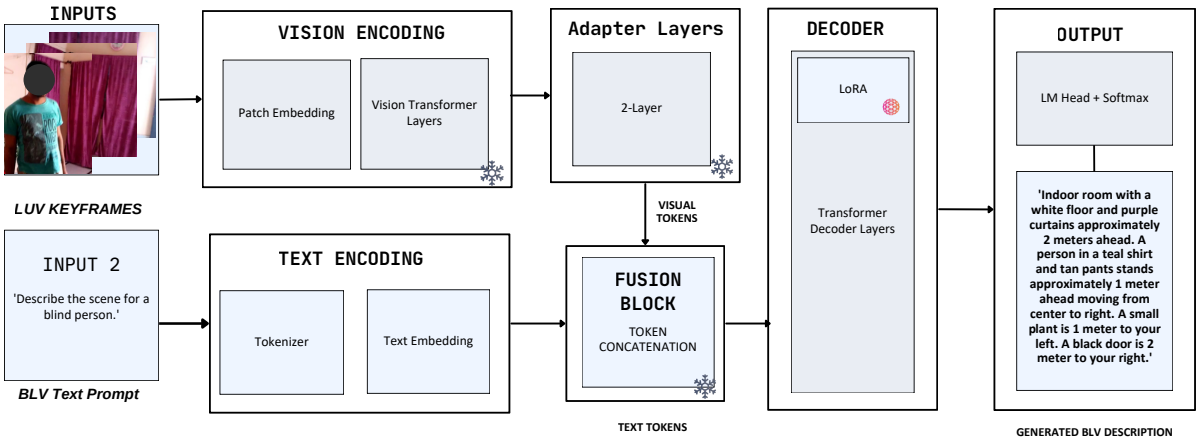

Figure 2: Overview of the post-training pipeline. LUV keyframes and a BLV-oriented text prompt are encoded separately through the vision and text encoders, aligned through lightweight adapter layers, and merged in a fusion block via token concatenation. The decoder, fine-tuned with LoRA while keeping selected backbone components frozen, generates spatially grounded BLV descriptions that include distances, directions, objects, obstacles, etc.

(86M parameters, 50–100 ms overhead)—making it suitable for mobile deployment.

Teacher Caption Generation We use Gemma-4-31B-IT as the teacher, as it is a very powerful image-to-text model and shows excellent detailing and grounding in video. The system prompt encodes professional AD guidelines (Bittner, 2012): present tense, safety-critical content first, explicit spatial positions with distances, observable features only, environment type in the first sentence, hazard/obstacle/level-change flags, active voice, maximum 4 sentences. We generate captions for 9,646 videos: 7,985 from Charades (Sigurdsson et al., 2016) and 1,661 from AVCaps (Sudarsanam et al., 2025). You can find the coverage of the dataset in the pie chart shown in the Figure 3.

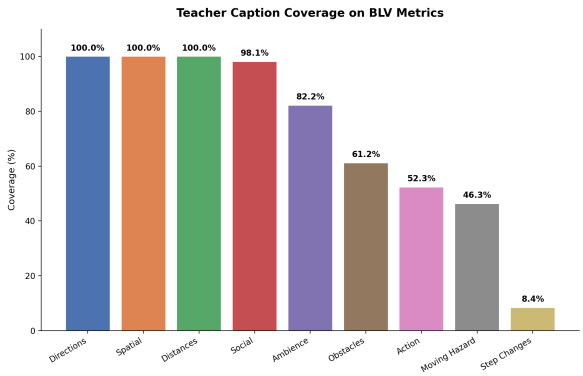

Figure 3: Coverage of BLV-relevant attributes in teacher-generated captions. The captions consistently include core navigation cues such as directions, distances, and spatial layout, while also covering social context, ambience, obstacles, actions, and moving hazards. Lower coverage for step changes indicates an important remaining challenge for BLV-oriented scene description.

3.2 SFT with Phase-Switching

We fine-tune SmolVLM2-500M via LoRA(Hu et al., 2022a) in a single training run that internally switches between two phases, combining projector-only and full-adapter training without checkpoint restarts. Unless otherwise specified, we adopt the training configuration used in SmolVLM2(Marafioti et al., 2025b).

Phase 1: Visual Alignment (steps 0–T_s). During the first 25% of training ($T_s = 0.25 \times T_{\text{total}}$), only the SigLIP encoder and multimodal connector are trainable; LoRA adapters remain frozen. This aligns visual representations with the AD distribution before language model adaptation, preventing early overfitting to misaligned features.

Phase 2: Full Adaptation (steps T_s–T_{total}). A callback unfreezes all LoRA adapters and halves the vision encoder LR to 5×10^{-6}. We implement this via a custom `DifferentialLRTrainer` with three independent AdamW parameter groups.

Label Masking. Causal LM loss is computed over assistant response tokens only. Prompt tokens (system + user + image tokens) are masked to -100. Prompt length is determined by tokenising the prompt-only portion separately—avoiding error-prone token-ID-based detection.

Gemma-4-31B-IT

3.2.1 GRPO

Group Relative Policy Optimization(Shao et al., 2024) is our primary RL method. For each prompt x, the policy samples G=4 completions $\{y_i\}$, scored by reward $R(x, y_i)$ normalised within the group:

$$\hat{r}_i = \frac{R(x, y_i) - \mu_R}{\sigma_R}; \tag{2}$$

$$\mu_R, \sigma_R = \text{mean/std}_j\big(R(x, y_j)\big) \tag{3}$$

The policy is updated to increase the probability of high-$\hat{r}$ completions.

Composite BLV Reward. $R(x, y)$ comprises six deterministic, rule-based components computable in <1 ms per sample (Table1).

3.3 SFT-Patch Recovery

GRPO improves NAF but the RL optimization makes the policy focus on reward maximization, which degrades the Ambience sub-dimension of MCF (3.44→2.94). To recover general descriptive quality, we apply a lightweight SFT-patch: 3 epochs on 50 curated Gemma captions, using the GRPO checkpoint as base with a fresh LoRA adapter (Hu et al., 2022b) (r=64, α=128, LR 5×10^{-5}, batch size 2, gradient clip 1.0). At inference, all three LoRA adapters (SFTv2, GRPO, SFT-patch) are merged sequentially: $\mathbf{W}' = \mathbf{W} + \mathbf{AB}$, yielding a single model with no additional inference overhead.

Component	Criterion	Score
CAUTION	Both ref. & gen. contain it	+0.40
	Ref. has, gen. missing	−0.30
	Gen. has, ref. missing	−0.10
Directional	{left, right, center, ahead, behind}	+0.08/word (max +0.24)
Distances	≥1 metric distance match	+0.20
Structure	2–4 sentences	+0.16
	>6 sentences	−0.10
Hallucination	Hedging phrases†	−0.20
Env. ID	First sentence names scene‡	+0.10

Table 1: GRPO reward components. Range: [−0.50, +1.06]. † *"appears to be," "I cannot,"* etc. ‡14 scene-type terms.

3.4 Mobile Deployment

Model Preparation & Conversion: All three LoRA adapters (SFT v2, GRPO, SFT-patch) are folded directly into the base model weights, producing a single unified HuggingFace checkpoint (969 MB, FP16). This merged model is then converted to the GGUF format via `llama.cpp`, a process that inherently separates the architecture into two distinct components for deployment: the language model backbone (783 MB, F16) and the vision projector (`mmproj.gguf`, 191 MB, F16).

Importance Matrix Calibration: To optimize the weights for our specific task, `llama-imatrix` is run on a calibration dataset of 500 BLV captions. This records which weight channels are most frequently activated, generating an importance matrix that guides non-uniform bit allocation during quantization.

Mixed-Precision Quantization Strategy: To compress the model while preserving its ability to generate accurate descriptions, we employed a mixed-precision quantization pipeline. The majority of the network was compressed using importance-weighted, non-linear 4-bit quantization (IQ4_NL) (Malakhov, 2025). Guided by an importance matrix derived from our calibration dataset, this method maps weights down to an effective 4.5 bits by concentrating precision around the most frequently used values. However, applying uniform 4-bit compression degrades the model's positional reasoning. To mitigate this, we explicitly overrode the self-attention projections (q_proj, k_proj, v_proj) to a higher 5-bit precision (Q5_K). Protecting these specific attention mechanisms is critical for maintaining directional coherence when mapping visual tokens to text. This targeted override costs only a marginal increase in memory but substantially preserves output accuracy, yielding a highly optimized language backbone of 251 MB.

On-Device Inference: The two GGUF files (442 MB total) are deployed on a Samsung Galaxy A55 (Exynos1480, 8 GB RAM). `llama-mtmd-cli` is compiled natively in Termux. The vision projector encodes the input keyframe into token embeddings fed into the language model alongside the BLV navigation prompt.

4 Experiments

Experimental details. All GRPO experiments use the best SFT checkpoint as initialization and are trained with 4-bit NF4 quantization, bfloat16 compute, and a single 48 GB RTX A6000 GPU. We use group size G=4, a maximum generation length of

Model	Params	OCR ANLS	VQA Accuracy	VQA ANLS
SmolVLM2-256M base	256M	16.49%	1.08%	3.09%
SmolVLM2-500M base	500M	32.54%	37.75%	47.52%
SmolVLM2-2.2B base	2.2B	25.44%	27.51%	29.64%
LLaVA-1.5-7B	7B	28.78%	47.79%	62.96%
PaliGemma-3B	3B	69.76%	76.09%	87.36%
Qwen2-VL-2B	2B	86.10%	80.73%	89.24%
Qwen2.5-VL-3B	3B	88.56%	81.12%	90.39%
Qwen2-VL-7B	7B	87.23%	85.09%	92.89%
Smol-VL-BLV (Ours)	**500M**	**65.54%**	**54.47%**	**65.97%**

Table 2: Comparison of OCR and VQA performance across different vision-language models. Our 500M model substantially improves over the SmolVLM2-500M baseline and remains competitive with larger models.

256 tokens, learning rate 5×10^{-6}, batch size 1 with gradient accumulation of 8, KL coefficient β=0.1, weight decay 0.01, and train for 2 epochs, corresponding to 9,076 optimization steps. All LLM-judge scoring (MCF and NAF) uses Qwen2.5-32B run locally. Prior to adoption, we manually reviewed a sample of judge outputs against human ratings to validate scoring reliability. In the experimental results, we use four training configurations: Base (A), Base+SFT (B), Base+SFT+GRPO (C), and Base+SFT+GRPO+Patch (D).

4.1 Main Results

BLV improves OCR and TextVQA. Table 2 shows that BLV-oriented fine-tuning improves not only accessibility-focused description, but also general text-centric visual reasoning. These results indicate that training on spatially grounded BLV descriptions strengthens the model's ability to parse visual text, spatial layout, and object-text relations.

Cross-Model Comparison on BLV Quality under Efficiency. Table 3 compares Smol-VL-BLV with representative VLM baselines under BLV-oriented quality and computational cost. At the same 0.5B parameter scale and GFLOP budget as the SmolVLM2-500M baseline, our model improves Spatial score from 3.21 to 3.83 and Social score from 3.30 to 3.79. It also substantially improves general visual-text reasoning, increasing OCRBench ANLS from 32.5% to 65.5% and TextVQA accuracy from 37.8% to 54.5%. These results suggest that BLV-focused spatial post-training transfers beyond accessibility-specific captioning and strengthens OCR and VQA capabilities. Compared with larger VLMs, our model provides a favorable quality: compute trade-off. While Qwen2.5-VL-3B achieves higher OCR and VQA scores, it uses 6× more parameters. Smol-VL-BLV

Model	Params	Spatial	Social	OCR	VQA	GFLOPs
		(MCF sub-dims)				
PaliGemma2-3B	3B	1.15	1.02	7.8	14.3	10,543
moondream2	1.8B	1.00	1.00	–	–	–
SmolVLM2-2.2B	2.2B	3.49	3.45	25.4	27.5	125,406
SmolVLM2-500M(base)	0.5B	3.21	3.30	32.5	37.8	34,855
Smol-VL-BLV (Ours)	**0.5B**	**3.83**	**3.79**	**65.5**	**54.5**	**34,855**
Qwen2.5-VL-3B	3B	5.43	3.51	88.6	81.1	14,280

Table 3: Overall comparative analysis (n=47, GPU server). Spatial and Social are MCF sub-dimension scores (1–10 scale). OCR = OCR-Bench ANLS (%). VQA = TextVQA Accuracy (%). ms = mean latency.

Metric Quantization	Paper Baseline Vivo Y27, INT8	SFT A55, Q4_K_M	GRPO A55, Q4_K_M	GRPO + sft patch A55, IQ4_NL
Model (LM)	–	290 MB	290 MB	251 MB
Total on-device	–	481 MB	481 MB	442 MB
RAM load time	–	504 ms	488 ms	291 ms
TTFT (prompt eval)	–	∼35 s	35.3 s avg	25.1 s avg
Generation speed	13.55 tok/s	17.1 tok/s	19.1 tok/s	39.3 tok/s
Total per frame	29.9 s	44.2 s	38.7 s	27.1 s
Peak RAM	761 MB	619 MB	985 MB	780 MB
Model quality (MCF)	–	4.38	4.71	same weights

Table 4: On-device performance comparison across model versions and hardware configurations

therefore offers strong BLV-oriented spatial and social understanding in a compact model, making it more suitable for resource-constrained BLV assistance.

Mobile Deployment Results. Table 4 reports on-device benchmarks across deployment iterations on the Samsung Galaxy A55 (n=21 keyframes for the final configuration).

The IQ4_NL quantization with importance-matrix calibration yields substantial gains over both the paper baseline and the initial Q4_K_M deployment. Generation speed more than doubles relative to Q4_K_M (+106%), and at 39.3 tok/s also surpasses the paper's reported 13.55 tok/s by nearly 3x . TTFT and total per-frame latency fall by 29% and 30% respectively, while the model file shrinks 13% and peak RAM drops 21% — all relative to Q4_K_M. No crashes or out-of-memory errors were observed across all 21 evaluation frames.

For cross-device context, we evaluated the same 251 MB quantized model on a cloud T4 GPU, the A55 CPU, and an Apple Silicon Mac using Metal (Table 5). Full generation completes in under 2.5 s on both the T4 and Mac, compared with 27.1 s on the A55. The A55 uses CPU-only inference due to unavailable hardware acceleration support (Appendix A.3). Thus, its 27.1 s latency reflects the deployment limitation rather than the model's potential with GPU/NPU acceleration.

Human Evaluation Twenty sighted crowd work raters evaluated five real time demo videos across

Metric	Cloud (T4)	A55 (CPU)	Mac (Metal)
TTFT / Prefill	1.39 s	25.1 s	1.95 s
Generation Time	0.31 s	1.4 s	0.46 s
Total Latency	1.70 s	27.1 s	2.41 s
New Tokens	58	60	61
Gen. Speed (tok/s)	187.42	42.9	132.6
Latency/Token (ms)	5.34	23.3	7.54

Table 5: Cross-hardware latency for the same quantized model. The A55's disproportionate latency reflects CPU-only fallback from missing NPU/GPU driver support (Appendix A.3), not a property of the model itself.

Metric	**A**(Base)	**B**(SFT)	**C**(+GRPO)	**D**(+Patch)	Δ
BLEU-1	19.02	18.08	22.55	**45.57**	+26.55
BLEU-4	0.74	1.66	1.97	**13.53**	+12.79
ROUGE-L	10.74	17.66	17.93	**30.49**	+19.75
METEOR	13.76	15.65	17.68	**35.99**	+22.23
CIDEr	0.0002	0.0021	0.0018	**0.0112**	+0.011

Table 6: NLP metrics on the 458-sample final evaluation set. Condition D achieves large gains across all metrics.

different environments (obstacles, floor objects, doors). Each video was rated from 0–5 on five criteria: Navigation Relevance, CAUTION/Alert Accuracy, Spatial and Directional Precision, Response Conciseness, and Overall Trustworthiness. Navigation Relevance measures whether the description contains directly actionable spatial content: obstacles, directions, and distances. CAUTION/Alert Accuracy evaluates whether hazards are correctly identified. Spatial and Directional Precision captures whether grounded spatial cues, such as left/right orientation and estimated distances, are present. Response Concise reflects whether the description is delivered within two sentences, satisfying the low-latency constraint of mobile deployment.

Mean scores across evaluators were: Navigation Relevance (4.0), CAUTION/Alert Accuracy (3.5), Spatial and Directional Precision (4.5), Response Conciseness (4.5), and Overall Trustworthiness (4.0), yielding an overall average of 4.1 out of 5. The high scores on spatial precision and conciseness confirm the model's suitability as a navigation aid, while the lower hazard accuracy score reflects the inherent difficulty of conveying hazard severity through language alone.

Qualitative Results In the Figure 4 illustrates the progressive improvement from base to fine-tuned outputs on a representative indoor scene. Our model's results show that compact and BLV are relevant in complex scenes.

Metric	**A** (Base)	**B** (SFT)	**C** (+GRPO)	**D** (+Patch)
BLV Coverage (%)				
Spatial	79	81	97	**100**
Social	94	94	98	**100**
Action	68	79	**88**	62
Ambience	76	91	91	**98**
BLV Mean	79	86	**93**	90
Navigational Coverage (%)				
Obstacles	77	85	92	**96**
Step changes	4	7	7	**5**
Directions	29	51	88	**100**
Moving hazards	16	21	17	**66**
Distances	45	20	22	**100**
Nav. Mean	34	37	45	**73**

Table 7: BLV keyword coverage (458 samples). Condition D achieves 100% coverage on spatial orientation, directions, and distances.

Dimension	**A** (Base)	**B** (SFT)	**C** (+GRPO)	**D** (+Patch)
LLM Judge (1–10 scale)				
MCF Score	1.88	2.26	2.39	**2.94**
NAF Score	1.84	2.44	2.35	**3.57**
MCF Sub-Dimensions				
Spatial orient.	1.21	1.71	1.85	**3.23**
Social interact.	1.47	1.62	1.82	**2.60**
Action events	2.16	2.36	2.47	**3.00**
Ambience	2.70	3.37	**3.44**	2.94
NAF Sub-Dimensions				
Descriptiveness	1.46	1.85	1.92	**2.87**
Objectivity	2.71	3.77	3.38	**5.38**
Accuracy	1.09	1.59	1.55	**2.42**
Clarity	2.09	2.55	2.54	**3.63**

Table 8: LLM judge scores and sub-dimension breakdown (1–10, 458 samples). Condition D achieves the highest scores on 7 of 8 dimensions, with GRPO (C) leading only on ambience.

Method	**MCF**	**NAF**	**Overall**
Base (zero-shot)	3.90	3.98	3.92
SFT v2	**4.38**	3.98	3.97
DPO	3.89	3.98	3.92
RLAIF-V DPO	3.88	3.96	3.91
GRPO	4.18	**4.20**	**4.19**

Table 9: Preference alignment comparison on the balanced evaluation set. Base, SFT v2, DPO, and RLAIF-V DPO are evaluated on 469 samples using a 1–5 judge scale. The DPO and RLAIF-V DPO fall below SFT v2 on MCF.

4.2 Discussion

Some discussions on emergent capabilities after spatial grounding SFT and GRPO finetuning of SmolVLM-500M : **(1)** The spatially grounded question-answer pairs generated by Gemma 4, included the state of the environment, nearby hazards, directional cues, person's position, color of cloths,

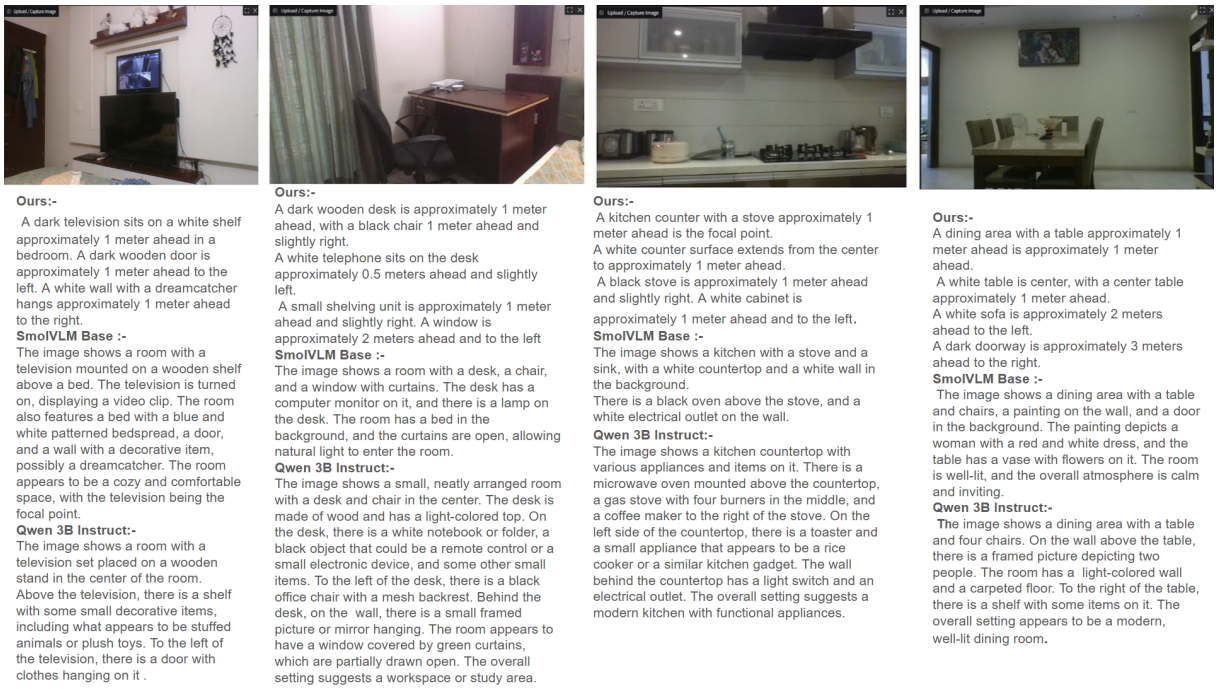

Figure 4: Qualitative comparison of scene descriptions generated by our model, SmolVLM-500M (Base), and Qwen-3B across four indoor scenes. Our model produces concise, spatially grounded, navigation-focused descriptions, while base models generate generic, observation-style captions without actionable spatial information.

and their movements. Spatial awareness needs the model to specialise in coordinate mapping, relative positioning and bounding logic. **(2)** Using a very good teacher (Gemma 4) to generate the BLV focused description having spatial awareness, forces the SmolVLM to build spatial maps for BLV navigation. **(3)** This structural alignment makes it much sharper at parsing the geometric boundaries of characters and text strings, so there is a significant improvement in the OCR performance of the finetuned model over the baseline SmolVLM. **(4)** Small models have a tendency to copy the style and rigor of their training data. The injection of high-quality synthetic data from Gemma 4 makes the SmolVLM adopt a better reasoning strategy, which can be clearly inferred from Table 8, resulting in 44 % improvement from baseline. **(5)** Early GRPO experiments used a few-shot BLV examples in the prompt, but this led the model to imitate their structure instead of grounding outputs in the scene. We therefore switched to zero-shot prompting during GRPO and used the later SFT-patch stage to improve descriptive quality. We also note that CIDEr scores remain low in absolute terms across all conditions (Table 6) this is expected under single reference evaluation, since CIDEr's n-gram/TF-IDF overlap is typically computed against multiple human references, making exact overlap structurally rare here with only one teacher generated reference per video. The relative improvement across conditions remains the meaningful signal.

4.3 Ablation Studies

This design allows us to isolate the contribution of each stage: supervised distillation for output structure, GRPO for BLV-specific preference alignment, and the patch-based SFT stage for recovering descriptive quality and improving spatial grounding.

Stages of post-training. The cumulative ablation shows that each post-training stage contributes a distinct improvement. SFT first establishes the basic audio-description structure, improving general description quality and BLV coverage, with ROUGE-L increasing from 10.74 to 17.66, METEOR from 13.76 to 15.65, and BLV Mean from 79% to 86% (Tables 6 and 7). GRPO then strengthens BLV-specific alignment by increasing BLV Mean to 93%, improving directional coverage from 51% to 88%, and raising obstacle coverage from 85% to 92%, showing that the online reward encourages navigationally useful descriptions. Finally, the SFT-patch stage produces the strongest overall gains: condition D achieves the best NLP scores and improves spatial coverage to 100%, direction coverage to 100%, distance coverage to

100%, and moving-hazard coverage to 66%. These results indicate that SFT transfers the teacher's descriptive structure, GRPO reinforces BLV-oriented spatial and hazard cues, and the patch stage provides localized visual grounding for precise metric and directional descriptions.

LLM Judge: Sub-Dimension Analysis Table 8 breaks down the LLM judge scores across all eight dimensions for the four primary conditions. The most striking improvement is in **objectivity**: from 2.71 (Base) to 5.38 (D), a +2.67 gain. This indicates that the pipeline successfully eliminates subjective hedging (e.g., *"appears to be," "seems like"*) and produces factual, AD-compliant descriptions. **Spatial orientation** shows the second-largest gain (+2.02), directly reflecting the GRPO reward function's emphasis on directional language and metric distances.

Comparison with other preference-alignment methods. Table 9 compares GRPO with offline preference-alignment methods. DPO and RLAIF-V do not improve over SFT, with MCF scores of 3.89 and 3.88 compared with 4.38 for SFT v2. In contrast, GRPO achieves stronger NAF performance, reaching 4.20 after rescaling. This suggests that offline preference pairs may capture general caption quality but are less effective at enforcing BLV-specific navigational behavior. The online GRPO reward is more suitable because it directly optimizes directional language, distance references, and hazard awareness.

5 Conclusion

This work demonstrates that a <1B VLM which is targeted to run on-device with resource constraints, can be effectively specialized for low vision-compliant scene narration for navigational assistance. Starting from baseline, our distillation pipeline transfers spatial awareness and safety from a larger teacher, resulting in a model that outperforms even larger models on custom and benchmark evaluation criteria. We observed SFT alone establishes a strong general baseline, but it is GRPO that drives the gains in navigational content that define BLV utility. GRPO is not just a contributor to the RL stage, it is the underlying driver of the best overall model's superiority. The on-device benchmarking results on a mid-range Android handset makes it practically deployable and meaningfully superior to larger baselines.

6 Future Work

There are several directions that we would like to explore in future work. We plan to collect more training data covering stairs, curbs, elevation changes, and moving hazards to address the gaps observed in Table 6. We also plan to introduce a dedicated step-change reward during the GRPO stage to encourage the model to learn this behavior directly. Further, we will evaluate the system in challenging environments such as crowded crosswalks and busy intersections to test its robustness beyond the current dataset. We also plan to improve our evaluation by developing a geometry based spatial correctness metric, reducing our reliance on keyword matching and LLM judge scores. Finally, we will add a separate factual accuracy check for incorrect directions, false hazard claims, and fabricated distances to better assess the system's reliability and safety.

Limitations

This work primarily targets low-vision users; while we attempted to extend its utility to fully blind users, the system does not yet meet the reliability required for safe independent blind navigation. The dataset used is limited to some indoor and outdoor scenes, which can be improved to more difficult navigation related domains such as road and crowded scenes. The evaluation can be made by BLV users. Coverage gaps remain for step-change (5%) and moving-hazard (66%) detection Table 7. We attribute this primarily to under representation of these scenarios in our training data Figure 3 rather than a limitation of the GRPO reward function itself. Stairs, curbs, elevation changes, and moving hazards are comparatively rare events in these source datasets, giving the model few positive examples to learn from. Distances generated in the teacher descriptions are approximate visual judgments derived from the teacher model rather than sensor measured values.

Ethics Statement

Assistive AI systems for low-vision people raise issues of privacy, consent for bystanders, overreliance, safety, accessibility equity, and representation of disability communities. User data should be collected with informed consent and protected carefully.

References

Shruti Singh Baghel, Yash Pratap Singh Rathore, Anurag Pradhan, Sushovan Jena, Arnav Bhavsar, Amit Shukla, and Pawan Goyal. 2025. Towards blind and low-vision accessibility of lightweight vlms and custom llm-evals. In *Proceedings of the 1st Workshop on Multimodal Models for Low-Resource Contexts and Social Impact (MMLoSo 2025)*, pages 86–94.

Hansjörg Bittner. 2012. Audio description guidelines: A comparison. *New perspectives in translation*, 20(2012):41–61.

Maryam Cheema, Hasti Seifi, and Pooyan Fazli. 2025a. Describe now: User-driven audio description for blind and low vision individuals. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*, pages 458–474.

Maryam S Cheema, Sina Elahimanesh, Pooyan Fazli, and Hasti Seifi. 2026. Vidscribe: Multimodal ai for customizing audio description and question answering in online videos. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–6.

Maryam S Cheema, Sina Elahimanesh, Samuel Martin, Pooyan Fazli, and Hasti Seifi. 2025b. Describepro: Collaborative audio description with human-ai interaction. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–19.

Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, and 1 others. 2023. Mobilevlm: A fast, strong and open vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*.

Megan Conway, Brett Oppegaard, and Tuyet Hayes. 2020. Audio description: Making useful maps for blind and visually impaired people. *Technical Communication*, 67(2):68–86.

Anne Lesley Corn and Jane N Erin. 2010. *Foundations of low vision: Clinical and functional perspectives*. American Foundation for the Blind.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022a. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022b. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Jin Huang, Yuchao Jin, Le An, and Josh Park. 2025. Litevlm: A low-latency vision-language model inference pipeline for resource-constrained environments. *arXiv preprint arXiv:2506.07416*.

Susan J Leat, Gordon E Legge, and Mark A Bullimore. 1999. What is low vision? a re-evaluation of definitions. *Optometry and Vision Science*, 76(4):198–211.

Chaoyu Li, Sid Padmanabhuni, Maryam S Cheema, Hasti Seifi, and Pooyan Fazli. 2025. Videoa11y-40k: A large-scale dataset for accessible video understanding. In *VizWiz Grand Challenge Workshop, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Franklin Mingzhe Li, Michael Xieyang Liu, Cynthia L Bennett, and Shaun K Kane. 2026. Adcanvas: Accessible and conversational audio description authoring for blind and low vision creators. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–21.

Aida Torre López. 2023. Audio description guidelines: Une 153020 standard and netflix style guide. *Professional Communication and Translation Studies*, (16):89–100.

Kyrylo S Malakhov. 2025. Deploying llms on cpu-only environments with llama. cpp library set: Medlocalgpt project case.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, and 1 others. 2025a. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, and 1 others. 2025b. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and 1 others. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European conference on computer vision*, pages 510–526. Springer.

Jaimie D Steinmetz, Rupert RA Bourne, Paul Svitil Briant, Seth R Flaxman, Hugh RB Taylor, Jost B Jonas, Amir Aberhe Abdoli, Woldu Aberhe Abrha, Ahmed Abualhasan, Eman Girum Abu-Gharbieh, and 1 others. 2021. Causes of blindness and vision impairment

in 2020 and trends over 30 years, and prevalence of avoidable blindness in relation to vision 2020: the right to sight: an analysis for the global burden of disease study. *The Lancet global health*, 9(2):e144–e160.

Parthasaarathy Sudarsanam, Irene Martín-Morató, Aapo Hakala, and Tuomas Virtanen. 2025. Avcaps: An audio-visual dataset with modality-specific captions. *IEEE Open Journal of Signal Processing*, 6:691–704.

OM Tripathi. 2024. " gguf models and quantization. *SSRN*.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and 1 others. 2025. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19769–19780.

Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xiujun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. 2020. Minivlm: A smaller and faster vision-language model. *arXiv preprint arXiv:2012.06946*.

Xiao Wang, Ibrahim Alabdulmohsin, Daniel Salz, Zhe Li, Keran Rong, and Xiaohua Zhai. 2025. Scaling pre-training to one hundred billion data for vision language models. *arXiv preprint arXiv:2502.07617*.

Shuchang Xu, Xiaofu Jin, Gaurav Jain, Wenshuo Zhang, Huamin Qu, Brian A Smith, and Yukang Yan. 2026. Sonic stage: Automatically generating an interactive spatial soundscape to facilitate dialogue video comprehension for blind and low vision viewers. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Shuchang Xu, Xiaofu Jin, Wenshuo Zhang, Huamin Qu, and Yukang Yan. 2025. Branch explorer: Leveraging branching narratives to support interactive 360 video viewing for blind and low vision users. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18.

A Deployment Architecture & Optimization

A.1 Model Architecture

Our deployed model is SmolVLM2-500M-Video-Instruct with a three-stage LoRA training lineage. Each stage added a LoRA adapter on top of the previous, and all three adapters were merged into the base model weights before export, producing a single HuggingFace model folder with no separate adapter files.

A.1.1 Neural Network Breakdown

Component	Architecture Detail	Params	Role
Vision Encoder	SigLIP, ViT-style, Patch/14, 384×384 input	∼86.4M	Pixels → patch embeddings
Vision Projector	Linear projection + LayerNorm	∼11.8M	Vision space → LM tokens
Language Model	LLaMA arch, 32 blocks, ctx 8192, vocab 49280	∼409.3M	Text generation
Full model	`blv_final`/ on server	∼507.5M	Single artifact

Table 10: Neural network component breakdown of the deployed model.

A.1.2 Why the Model Splits into Two GGUF Files

The model produces two GGUF files totaling 442 MB, rather than the single 103 MB file used by the unmodified paper baseline. When converting any VLM via `convert_hf_to_gguf.py`, the `llama.cpp` script always separates the vision projector (`mmproj`) from the language model backbone by design, since the two components run sequentially and can be quantized independently. The paper baseline used the original unmodified SmolVLM2-500M (no LoRA, no fine-tuning), whose architecture the conversion script packed into one file. Once LoRA adapters are merged and the modified weights are exported, the script treats them differently. At runtime, `llama-mtmd-cli` loads both files into RAM, runs `mmproj` to encode the image into vision token embeddings, feeds those into the LM's token stream, and the LM generates text. From the user's perspective, the system behaves as a single model—two files is purely a storage artifact.

A.1.3 Full Deployment Pipeline

```
blv_final/  (merged HF model, float16, ~969 MB)
  SFT v2 + GRPO + SFT Patch v2 LoRAs - all folded in
          ↓   Step 1: convert_hf_to_gguf.py
blv_final_f16.gguf   (GGUF, F16, 783 MB)
sft_patch_mmproj.gguf   (vision projector, extracted separately)
          ↓   Step 2: llama-imatrix  (BLV calibration, ~30 min)
          ↓   Step 3: llama-quantize  IQ4_NL + imatrix + Q5_K attn
blv_final_iq4nl_best.gguf   251 MB   (LM backbone, IQ4_NL)
sft_patch_mmproj.gguf   191 MB   (vision projector, F16)
  Total on-device: 442 MB
          ↓   ADB push / USB transfer
Samsung Galaxy A55   (Exynos 1480, 8 GB RAM, Android 16)
  Termux + llama-mtmd-cli   (CPU-only, 4 threads)
   · mmproj encodes frame.jpg → vision token embeddings
   · LM receives [vision tokens] + [BLV navigation prompt]
   · LM generates BLV description (≤60 tokens, greedy)
   · termux-tts-speak reads output aloud
```

Figure 5: End-to-end deployment pipeline from merged HuggingFace model to on-device inference on Samsung Galaxy A55.

A.2 Quantization

The goal of quantization is to shrink the model from its training-time float16 representation (783 MB) into something a phone CPU can load and run at acceptable speed. We explored several approaches before settling on IQ4_NL.

A.2.1 Step-by-Step Procedure

Step 1 — Merge LoRA into base weights. After training, three LoRA adapters are stored as small delta matrices (each 32–77 MB). These are added into the base model weights via $W_{\text{merged}} = W_{\text{base}} + (A \times B)$, where A and B are the low-rank matrices, producing `blv_final/`. The merge takes ∼3 minutes on CPU with no GPU required. `llama.cpp`'s converter requires complete weight matrices per layer and does not understand HuggingFace adapter files.

Step 2 — Convert to GGUF (F16). `convert_hf_to_gguf.py` reads the merged HuggingFace model and writes it to GGUF format—a single binary packaging all weights plus model metadata (architecture, vocabulary, context length). Weights remain float16 at this stage. Output: `blv_final_f16.gguf` (783 MB). The mmproj is extracted as a separate file automatically. A known issue: the script triggers a `torchvision` circular import through `transformers 5.5.1`; fixed with a `ModuleSpec` mock that registers a dummy `torchvision` module before conversion runs.

Step 3 — Generate importance matrix (imatrix). `llama-imatrix` is run against the F16 GGUF using a calibration dataset of ∼500 BLV captions from training data, recording which weight channels were most frequently activated. Standard quantization compresses all weights equally; the imatrix directs the quantizer to allocate more bits to high-salience channels. Runtime: ∼30 minutes (one-time cost). Output: `blv_imatrix.dat`.

Step 4 — Quantize to IQ4_NL with mixed precision. `llama-quantize` takes the F16 GGUF and imatrix file and produces the final model using IQ4_NL (importance-weighted, non-linear 4-bit), with a precision override: attention layers (`q_proj`, `k_proj`, `v_proj`) are quantized at Q5_K because they directly control spatial and directional focus. IQ4_NL uses a non-linear mapping (float16 → 4.5 bits effective), concentrating precision around frequently-used values. The Q5_K override costs ∼15 MB but improves BLV description coherence. Output: `blv_final_iq4nl_best.gguf` (251 MB), 13% smaller than the Q4_K_M baseline (290 MB).

Step 5 — Transfer and run on Android. The two GGUF files were transferred via USB using ADB. `llama-mtmd-cli` was compiled directly on-device (∼30–40 min, one-time). Inference command: `llama-mtmd-cli -m blv_final_iq4nl_best.gguf --mmproj sft_patch_mmproj.gguf --image frame.jpg -p [BLV_PROMPT] -n 60 --temp 0.0 -t 4`. Output is read aloud via `termux-tts-speak`.

A.2.2 Quantization Methods Explored

Method	What We Did	Outcome
Q4_K_M	Standard 4-bit K-quant; Apr 25 & May 9 benchmarks.	290 MB; TTFT ∼35 s; 17–19 tok/s. Baseline for comparisons.
Q5_K_M	5-bit K-quant; tested spatial-language coherence gain.	∼340 MB. RSS >1050 MB with mmproj; quality gain marginal.
Q8_0	Closest-to-F16 quality; evaluated theoretically.	783 MB; rejected—total RAM exceeds 1 GB plus mmproj plus OS.
IQ4_XS	Importance-weighted 4-bit extra-small.	∼230 MB; noticeably worse coherence on BLV spatial descriptions.
IQ4_NL (selected)	imatrix from BLV captions; Q5_K on attn layers.	**251 MB; TTFT 25.1 s; 39.3 tok/s.** Final deployed model.
Q3 & below	IQ3, Q3, Q2 variants; reviewed llama.cpp docs.	Rejected. Sub-4-bit renders small VLMs (<1B) incoherent (SPEED-Q).
AWQ	Activation-aware quant via `autoawq`.	No support for SmolVLM2's SigLIP+LLaMA hybrid; `autoawq` targets LLaVA/Qwen-VL only.
GPTQ	Hessian-based PTQ; calibration on BLV data.	Same blocker as AWQ. imatrix in IQ4_NL is conceptually equivalent and works natively in `llama.cpp`.
QAT	Re-train with fake quant ops in forward pass.	Requires full SFT+GRPO+SFT_Patch rerun; `bitsandbytes` QAT experimental for VLMs. Future work.
BitNet 1.58b	Ternary weights; integer-only arithmetic.	Inapplicable post-hoc; requires training from scratch with ternary constraints.

Table 11: Quantization methods explored and outcomes.

A.3 NPU Acceleration: Approaches Explored

TTFT is CPU-bound at ∼25 s. The Exynos 1480 in the A55 includes a dedicated NPU. We investigated several acceleration routes.

Android NNAPI. NNAPI routes neural network computations to device accelerators (NPU/DSP/GPU). `llama.cpp`'s Android build does not expose an NNAPI execution provider—its GPU path targets OpenCL and Vulkan. Samsung's Exynos NNAPI driver also frequently falls back to GPU for unsupported ops. A proper NNAPI pipeline would further require a full Android app (Java/Kotlin + NDK), incompatible with our Termux CLI workflow.

ONNX Runtime Mobile + NNAPI. HuggingFace `optimum-cli export onnx` does not support SmolVLM2: its SigLIP encoder combined with the pixel_values mid-sequence injection has no registered export class in `optimum`'s `TasksManager`. The exporter raised `KeyError: 'smolvlm2'` immediately. A custom ONNX exporter would need to implement the full SigLIP + token-injection logic as ONNX-compatible ops—not feasible within the timeline.

MLC-LLM. MLC-LLM supports LLaMA, Mistral, and Qwen; SmolVLM2 is not included. Its Android hardware acceleration path targets Snapdragon via Qualcomm ADRENO/Hexagon. On Exynos 1480, the app falls back entirely to CPU—NPU acceleration is unavailable for non-Snapdragon devices.

Vulkan / OpenCL (`llama.cpp`). Building `llama.cpp` with `-DGGML_VULKAN=ON` in Termux failed: the Vulkan headers were found but the vendor-specific Mali ICD loader (needed to route calls to hardware) was not. The CLBLAST/OpenCL path had the same issue—the Exynos Mali OpenCL 2.0 driver resides in Samsung's proprietary partition, inaccessible from Termux.

Pocketpal AI. This consumer GGUF inference app did not support vision (`mmproj`) at time of testing. A later version requires LLaVA-style mmproj naming conventions, which our export does not match. The GPU toggle reports "No GPU backend available" on A55 (same Vulkan blocker as above).

Samsung ONE SDK. Direct Exynos NPU programming via Samsung's proprietary ON-device Neural Engine SDK requires NDA registration; access was not granted in time. The SDK also requires models in `.nnc` format, produced from TFLite or ONNX—both of which were blocked for SmolVLM2 (see above).

A.4 Benchmark Results

A.4.1 IQ4_NL Benchmark — Samsung A55 (n=21 frames)

Model: `blv_final_iq4nl_best.gguf` (251 MB) + `sft_patch_mmproj.gguf` (191 MB) = 442 MB. Termux, 4 threads, CPU-only. Date: May 24, 2026.

Metric	Min	Avg	Max
Model load into RAM	272 ms	291 ms	321 ms
Prompt eval (TTFT)	24.1 s	25.1 s	28.2 s
Generation	1.0 s	1.4 s	1.7 s
Total per frame	26.0 s	27.1 s	30.5 s
Prompt speed (tok/s)	32.7	37.0	38.2
Generation speed (tok/s)	34.2	39.3	43.2
Peak RSS	—	∼780 MB	1004 MB

Table 12: Per-frame benchmarks for the final IQ4_NL deployment. 2 of 21 frames showed thermal throttling; 19 frames sustained normal speed. No crashes or OOM events.

A.4.2 Deployment History — All Stages

Note on TTFT. Early benchmarks reported 504 ms as TTFT; that was the RAM load time. True TTFT—processing image and prompt tokens—is ∼35 s for Q4_K_M and ∼25 s for IQ4_NL.

Metric Quantization	Paper Baseline Vivo Y27, INT8	SFT A55, Q4_K_M	GRPO A55, Q4_K_M	GRPO + sft patch A55, IQ4_NL
Model (LM)	–	290 MB	290 MB	251 MB
Total on-device	–	481 MB	481 MB	442 MB
RAM load time	–	504 ms	488 ms	291 ms
TTFT (prompt eval)	–	∼35 s	35.3 s avg	25.1 s avg
Generation speed	13.55 tok/s	17.1 tok/s	19.1 tok/s	39.3 tok/s
Total per frame	29.9 s	44.2 s	38.7 s	27.1 s
Peak RAM	761 MB	619 MB	985 MB	780 MB
Model quality (MCF)	–	4.38	4.71	same weights

Table 13: Deployment benchmark history across all stages. *Paper's 29.9 s is for text-only inference (no vision tokens); the same paper reports 60–83 s for full multimodal inference with images. †Paper baseline TTFT likely includes image encoding overhead in their unified model and is not directly comparable to our split pipeline.

A.4.3 IQ4_NL vs. Q4_K_M — Summary

Metric	Q4_K_M (May 9)	IQ4_NL (May 24)	Δ
Total per frame	38.7 s	27.1 s	−30%
TTFT	35.3 s	25.1 s	−29%
Generation speed	19.1 tok/s	39.3 tok/s	+106%
LM file size	290 MB	251 MB	−13%
Total on-device	481 MB	442 MB	−8%
Peak RAM	985 MB	∼780 MB	−21%
RAM load time	488 ms	291 ms	−40%

Table 14: IQ4_NL vs. Q4_K_M improvement summary.