Not Truly Multilingual: Script Consistency as a Missing Dimension in VLM Evaluation*

Anonymous ACL submission

Abstract

Current multilingual evaluations for Vision-Language Models (VLMs) assume a one-to-one mapping between language and orthography, overlooking billions of users of multi-script languages. We introduce **PuMVR** (Punjabi Multimodal Visual Reasoning), a benchmark of 1,000 strictly parallel image-text instances across Punjabi's three active scripts: Gurmukhi, Shahmukhi, and Roman. Evaluating 10 state-of-the-art VLMs, we expose a substantial and systematic **Script Gap**. Models frequently solve visual tasks in one script while failing identical tasks in another, with accuracy deltas reaching 16%. Crucially, visual input boosts absolute performance uniformly yet does not close the orthographic gap. Furthermore, cross-script in-context transfer is highly brittle, exposing script-locked knowledge representation. Supported by McNemar tests across all script pairs, our findings demonstrate that current "multilingual" VLMs are not truly multi-script. We propose the **Script Consistency Rate (SCR)**, which falls as low as 24.8% on our benchmark, as a mandatory metric for script-agnostic evaluation to ensure equitable AI access.

1 Introduction

The rapid evolution of Multimodal Vision-Language Models (VLMs) has been accompanied by claims of broad linguistic competence. Frontier systems such as GPT-4o (OpenAI, 2024) and Gemini 1.5 (Team et al., 2024) are validated on their ability to reason across dozens of languages. However, these evaluations rest on a precarious assumption: the **"One Language, One Script" (OLOS) paradigm**. By treating orthography as a deterministic function of language, current benchmarks, including XM3600 (Thapliyal et al., 2022) and MaXM (Changpinyo et al., 2023), overlook the reality of hundreds of millions of users who navigate the world through multi-script systems.

This evaluation gap reveals important questions about model explainability. If a model's reasoning fluctuates when identical semantic content is presented in different scripts, its knowledge may rely more heavily on orthographic patterns than language-level semantic concepts. Script variation serves as a critical diagnostic probe: if a visual concept understood in one script is inaccessible in another, the model's "multilinguality" may rely more on pattern matching than purely semantic grounding.

This work exposes an equity gap: benchmarks assume script-language isomorphism, which misrepresents performance for script-switching users. For a Punjabi speaker using Roman and Gurmukhi, a model that succeeds in one script but fails in the other is not partially capable, it is unreliable. We introduce Script Consistency Rate (SCR) to reframe multilingual evaluation from language breadth to orthographic robustness, establishing script-agnosticism as a requirement for true multilingual AI.

We use Punjabi as a diagnostic case study because it uniquely satisfies three conditions for isolating script as an independent variable. Spoken by over 125 million people, Punjabi operates through three distinct systems: Gurmukhi (Indic script), Shahmukhi (Perso-Arabic), and Roman (Latin transliteration). We introduce **PuMVR** (Punjabi Multimodal Visual Reasoning), a parallel-script benchmark of 1000 expert-

*Data and code is available at https://anonymous.4open.science/r/Not-Truly-Multilingual-PuMVR-BFC1/.

curated tasks isolating orthography as an independent variable in multimodal reasoning.

Our contributions are:

1. **The PuMVR Benchmark**: 1000 parallel instances across three scripts, providing the first controlled, three-way orthographic evaluation for multimodal reasoning.

2. **Quantification of Script Bias**: A systematic audit of 10 state-of-the-art VLMs revealing significant performance gaps (accuracy deltas up to 16%, Script Consistency Rates as low as 24.8%) and limited cross-script transferability.

3. **Statistical Validation of Script Bias**: McNemar tests across all pairwise script comparisons confirm the Script Gap is statistically robust - Gurmukhi-Shahmukhi gaps reach significance for 8 of 10 models, with 6 models at $p < 0.001$.

2 Related Work

The rise of VLMs, from CLIP (Radford et al., 2021) to instruction-tuned systems such as LLaVA (Liu et al., 2024) and Qwen-VL (Bai et al., 2023), has been accompanied by claims of multilingual competence across multiple languages. However, existing evaluation paradigms largely follow a one-script-per-language assumption. Current benchmarks conflate language with orthography, treating script as deterministic rather than variable. For over one billion speakers of digraphic languages like Punjabi (ਗੁਰਮੁਖੀ (Gurmukhi)), شاہمکھی (Shahmukhi), Roman), Serbian (Cyrillic, Latin), and Kurdish (Arabic, Latin, Cyrillic), this assumption masks orthographic bias that fragments AI access.

2.1 Orthographic Gaps in Multilingual Multimodal Benchmarks

Recent multilingual multimodal benchmarks have expanded evaluation coverage substantially. IGLUE (Bugliarello et al., 2022), XM3600 (Thapliyal et al., 2022), PaLo (Maaz et al., 2024), and MVL-SIB (Schmidt et al., 2025) broadened cross-lingual evaluation across dozens to hundreds of languages, while MaRVL (Liu et al., 2021), BLEnD-Vis (Tan et al., 2025), IndicVisionBench (Faraz et al., 2025), and ALM-Bench (Vayani et al., 2025) introduced culturally grounded multilingual reasoning tasks.

However, these milestones share a fundamental limitation: *each language appears in exactly one script*. MaRVL represents Tamil in Tamil script and Swahili in Latin, but never tests whether Tamil speakers using Romanization experience equivalent performance. This OLOS assumption artificially inflates multilingual capabilities, a model achieving 85% on "Punjabi" in Gurmukhi may drop to 69% in Shahmukhi for identical content, a disparity invisible on current leaderboards.

2.2 The Script Gap: From Text-Only Evidence to Multimodal Urgency

Text-only NLP has documented substantial script-dependent degradation. Pfeiffer et al. (2021) showed that multilingual BERT fails on unseen scripts, while Rust et al. (2021) demonstrated inequitable tokenizer allocation across orthographies, with low-resource scripts often receiving significantly fewer subword units. These disparities have practical consequences: Khullar et al. (2025) reported 5–12 point F1 degradation for Romanized Hindi-Urdu healthcare queries. Although prior work has explored romanization effects (Amrhein and Sennrich, 2020) and cross-script training strategies (Nguyen et al., 2024), no existing work systematically evaluates whether multimodal grounding mitigates or amplifies script-dependent failures.

2.3 Cultural Grounding and Script-Locked Knowledge

Effective multimodal reasoning requires cultural grounding beyond object recognition (Yin et al., 2021). Prior work has shown that VLMs struggle with culturally specific concepts and reasoning (Liu et al., 2021; Tan et al., 2025). However, existing benchmarks do not examine whether cultural knowledge itself becomes *script-dependent*. This question is particularly important for multiscript languages such as Punjabi, where Gurmukhi, Shahmukhi, and Roman scripts are associated with distinct historical, religious, and sociocultural contexts. PuMVR enables systematic

evaluation of whether VLMs activate different knowledge representations based solely on script variation.

PuMVR addresses this gap by isolating script as an independent variable, introducing SCR and Transfer Efficiency (TE) as metrics for orthographic robustness, and providing a replicable methodology for multi-script evaluation.

3 The PuMVR Benchmark

We introduce **PuMVR** (Punjabi Multimodal Visual Reasoning), the first benchmark designed to isolate script as an independent variable in multimodal reasoning. Unlike traditional multilingual benchmarks that conflate language with orthography, PuMVR provides 1000 culturally grounded image-reasoning tasks, each existing in perfect semantic equivalence across ਗੁਰਮੁਖੀ (Gurmukhi), شاہمکھی (Shahmukhi), and Roman scripts.

3.1 Design Rationale: Why Punjabi?

Punjabi serves as an ideal diagnostic for script bias due to three critical properties.

- Its three active scripts, Gurmukhi (Indian Punjab, 50M speakers), Shahmukhi (Pakistani Punjab, 60M speakers), and Roman (diaspora/digital, 15M users), are geographically and culturally isolated, minimizing training data overlap.
- The scripts represent fundamentally different visual systems: Gurmukhi is an Indic abugida, Shahmukhi uses Perso-Arabic cursive, and Roman relies on Latin linearity.
- Punjabi remains low-resource compared to Hindi or Urdu, amplifying reliance on memorized patterns over semantic grounding.

Crucially, certain cultural concepts carry script-specific associations, Gurmukhi predominates in Sikh contexts (Golden Temple), while Shahmukhi aligns with Islamic heritage (Badshahi Mosque), enabling tests of script-locked knowledge retrieval.

3.2 Dataset Composition and Structure

Example ID: I_1

Gurmukhi: ਪੱਗ : ਸਿਰ :: ਜੁੱਤੀ : ?
A ਹੱਥ B ਪੈਰ C ਅੱਖ D ਨੱਕ

Shahmukhi: پگ : سر :: جُتی : ؟
A ہتھ B پیر C اکھ D ناک

Roman: Pagg : Sir :: Jutti : ?
A hath **B** pair **C** akkh **D** nakk

Correct Answer: B

Reasoning: Since a turban (*pagg*) is worn on the head (*sir*), just as a shoe (*jutti*) is worn on the foot (*pair*).

Figure 1: Sample instance from PuMVR showing cross-script equivalent options.

PuMVR comprises 1,000 parallel instances. Each instance consists of one image paired with a question and four multiple-choice options provided in Gurmukhi, Shahmukhi, and Roman scripts, along with the corresponding correct answer (semantically equivalent across scripts); Figure 1 shows a representative instance.

Each instance contains: an image, question text in all three scripts, four multiple-choice options per script, the correct answer per script, and a human-authored reasoning explanation. By making script an explicit, controllable variable, the methodology extends naturally to other multi-script languages including Hindi-Urdu, Serbian, Kurdish, and Sindhi.

This structure preserves the same visual and semantic content across scripts, allowing observed differences in model behavior to be attributed primarily to orthographic variation rather than task difficulty or dataset composition.

3.3 Annotation Efforts and Quality

To validate the quality and reliability of PuMVR, we conducted a formal annotation quality study. The pipeline involved four annotators: two dataset curators (one of whom is an author of this paper), who were responsible for initial instance creation, and two independent, paid quality-check annotators recruited specifically for verification. All four participants are native speakers of Punjabi or Urdu and are fluent in the scripts they evaluated.

Annotation Guidelines. Comprehensive guidelines were prepared before any curation and annotation process, covering two complementary roles: (1) *curation guidelines* for instance authoring and semantic equivalence standards; and (2) *quality-checking guidelines* for evaluating instances and script correctness without reference to the original author's judgments. This dual approach ensured a shared standard and high-quality annotations.

Annotator Profiles. Both annotators were native Punjabi or Urdu speakers proficient across all three scripts, compensated at USD 15 per day, and had no prior access to the benchmark instances or hypotheses.

Annotation Protocol. All 1,000 instances were independently evaluated across five dimensions: semantic equivalence, answer correctness, and script accuracy for each of the three orthographies. Annotators applied a conservative criterion, marking *No* under genuine uncertainty rather than defaulting to *Yes*, ensuring that near-universal agreement reflects genuine expert consensus rather than passivity (see Appendix A for full guidelines).

Inter-Annotator Agreement. Inter-annotator agreement was consistently high across all five evaluation dimensions (Table 1). The label distribution is strongly skewed toward the positive class across all dimensions, reflecting the datasets curation process rather than annotator behavior. In such settings, chance-corrected agreement coefficients can become less informative due to prevalence effects. We therefore report Prevalence-Adjusted Bias-Adjusted Kappa (PABAK) (Byrt et al., 1993), along with positive-class F1 and observed agreement (P_o), as the primary agreement metrics.

Dimension	P_o	PABAK	F1
Semantic Equivalence	99.8%	0.996	0.999
Answer Correctness	98.5%	0.970	0.992
Script Acc. - Gurmukhi	100%	1.000	1.000
Script Acc. - Shahmukhi	100%	1.000	1.000
Script Acc. - Roman	100%	1.000	1.000

Table 1: Inter-annotator agreement across all 1,000 instances. We report observed agreement (P_o), Prevalence-Adjusted Bias-Adjusted Kappa (PABAK), and positive-class F1.

PABAK scores reach 0.970 or above and F1 scores reach 0.992 or above across all five dimensions. Script accuracy reached perfect agreement (100%) across all three orthographies, confirming that orthographic correctness is unambiguous among native-script-proficient annotators. That these same instances expose accuracy deltas of up to 16% and SCR values as low as 24.8% across 10 VLMs validates PuMVR as a genuinely challenging benchmark - the annotators confirmed dataset quality, not task simplicity.

4 Experimental Setup

Our experimental framework examines the **Script-Reality Gap** through three investigations in sequence: Experiment 1 establishes that script-dependent gaps exist and quantifies their magnitude; Experiment 2 tests whether visual input closes those gaps; and Experiment 3 tests whether in-context examples transfer knowledge across orthographies.

We evaluate 10 state-of-the-art VLMs spanning both proprietary and open-weight systems (Table 2), enabling assessment of frontier models alongside reproducible baselines. PuMVR contains 1,000 parallel instances; a stratified 375-instance evaluation split was fixed prior to any model evaluation to prevent data snooping, with the remaining 625 instances reserved for future fine-tuning and mitigation studies.

4.1 Experiment 1: The "Script Gap" Quantification

Objective: Establish the existence and magnitude of script-dependent performance bias under identical semantic conditions.

Model	Organization
Proprietary Frontier Models	
gpt-4o	OpenAI
gemini-2.5-flash	Google DeepMind
claude-sonnet-4	Anthropic
grok-4-1-fast-reasoning	xAI
Open-Weights Models	
Qwen2-VL-7B-Instruct	Alibaba
Qwen2-VL-72B-Instruct	Alibaba
Llama-3.2-11B-Vision	Meta
LLaVA-OneVision-1.5-8B-Instruct	LMMS-Lab
InternVL2_5-26B	OpenGVLab
Kimi-VL-A3B-Instruct	Moonshot AI

Table 2: Evaluated VLMs spanning frontier and open-weights systems.

Design : Each PuMVR instance is evaluated in three isolated passes, one per script, to prevent cross-script priming. Models receive the image, question, and four options in a single script using script-specific instruction templates (App. D), and must output the exact text of the correct option. We evaluate model performance using the following metrics.

1. **Script Accuracy** (Acc_s): Raw per-script performance

$$\text{Acc}_s = \frac{1}{|I|} \sum_{i \in I} \mathbb{1}[\text{Pred}_s(i) = \text{GT}_s(i)] \quad (1)$$

2. **Script Consistency Rate (SCR)**: Percentage of instances answered correctly across all three scripts simultaneously

$$\text{SCR} = \frac{1}{|I|} \sum_{i \in I} \prod_{s \in \mathcal{S}} \mathbb{1}[\text{Correct}_s(i)] \quad (2)$$

where $\mathcal{S} = \{\text{Gur}, \text{Shah}, \text{Rom}\}$
SCR serves as a strict script-agnostic benchmark. A model achieving 90% per-script accuracy but only 70% SCR reveals that 20% of its knowledge is orthographically fragmented.

3. **Performance Delta** (Δ): Maximum accuracy variance

$$\Delta = \max_{s_1, s_2 \in S} |\text{Acc}_{s_1} - \text{Acc}_{s_2}| \quad (3)$$

To confirm that this design measures comprehension rather than output formatting difficulty, we computationally verified all 11,250 model responses (10 models × 375 instances × 3 scripts). Across all scripts, 99.59% of errors were complete wrong-option selections indicating semantic comprehension failures, 0.41% were empty responses, and 0.00% were formatting artifacts. These results confirm that observed performance gaps reflect orthographic comprehension failures rather than formatting limitations. Unlike per-script accuracy, SCR exposes orthographic fragmentation that leaderboards mask: a model with 90% Gurmukhi and 85% Shahmukhi accuracy may still serve fewer than 78% of instances reliably across both scripts.

4.2 Experiment 2: Modality Importance Ablation

Objective: Determine whether visual grounding compensates for weak script comprehension or merely provides additive benefit.

Design : We compare Text-Only (question + options, no image) versus Multimodal (complete input) conditions across all scripts.

Metric : Visual Gain (VG) quantifies visual contribution:

$$\text{VG}_s = \text{Acc}_{\text{Multimodal}}(s) - \text{Acc}_{\text{Text-Only}}(s) \quad (4)$$

Uniform VG across scripts indicates visual information does not close the script gap, the bias is systematic, not compensatory.

4.3 Experiment 3: Cross-Script Transfer with Few-Shot Learning

Objective: Test whether in-context knowledge transfers across orthographies or remains script-locked.

Design : We employ $k = 3$ in-context exemplars under three conditions:

1. **Monoscript**: Examples and test in same script (e.g., Gurmukhi → Gurmukhi)

2. **Cross-Script**: Examples in different script (e.g., Roman → Gurmukhi)

3. **Mixed-Script**: Examples rotated through all three scripts

Metrics:

1. **Few-Shot Lift (FSL)**: Improvement from zero-shot

$$\mathrm{FSL}_s = \mathrm{Acc}_{\mathrm{fs}}(s \to s) - \mathrm{Acc}_{\mathrm{zs}}(s) \quad (5)$$

where $\mathrm{Acc}_{\mathrm{fs}}$ and $\mathrm{Acc}_{\mathrm{zs}}$ denote the accuracies for few-shot and zero-shot conditions, respectively.

2. **Transfer Efficiency (TE)**: Cross-script to in-script ratio

$$\mathrm{TE}_{T \to S} = \frac{\mathrm{Acc}_{\text{Few-Shot}}(T \to S)}{\mathrm{Acc}_{\text{Few-Shot}}(S \to S)} \times 100\% \quad (6)$$

TE quantifies how much in-context knowledge transfers across scripts. If TE $<$ 50%, the model has encoded script-specific surface patterns rather than transferable semantic representations. If TE exhibits asymmetry (e.g., $\mathrm{TE}_{\text{Roman}\to\text{Gurmukhi}} \gg \mathrm{TE}_{\text{Gurmukhi}\to\text{Roman}}$), it reveals an anchor script bias.

5 Results and Analysis

Our systematic evaluation of 10 state-of-the-art VLMs across the 375-instance evaluation split reveals a consistent and statistically significant finding: **current multilingual systems are not truly multi-script**. Orthographic variation systematically fragments model performance despite constant semantic content and visual grounding, with gaps confirmed significant for 8 of 10 models.

5.1 The Script Gap is Universal and Substantial

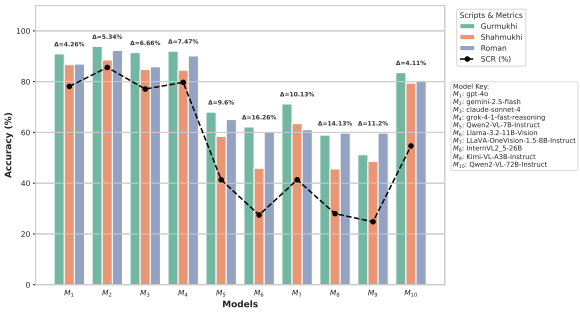

Figure 3: Script-dependent accuracy and Script Consistency Rate (SCR) across 10 VLMs.

Figure 3 reveals our primary finding: **all evaluated models exhibit script-dependent performance degradation**, detailed in the per-model breakdown (App. C). Accuracy deltas range from 4.11% (Qwen2-VL-72B-Instruct) to 16.26% (Llama-3.2-11B-Vision). Critically, this affects frontier models, gpt-4o, gemini-2.5-flash, and claude-sonnet-4 demonstrate Δ values of 4.26%, 5.34%, and 6.66% respectively, proving script bias is *not* a low-resource artifact but a systematic limitation of current VLM designs.

The **Script Consistency Rate (SCR)** exposes the severity of orthographic fragmentation. gpt-4o achieves 90.93% in Gurmukhi yet records only 78.13% SCR, **12.8% of its correctly answered instances are orthographically inconsistent**. This pattern intensifies in open-weights models: Llama-3.2-11B-Vision's SCR of 27.47% means nearly three-quarters of instances cannot be solved consistently across scripts, making reported Punjabi accuracy figures unreliable for real-world multi-script users.

Eight of ten models peak in Gurmukhi, reflecting its predominance in Indian web corpora. Shahmukhi consistently underperforms, averaging 7.7% below Gurmukhi, indicating that Perso-Arabic cursive representations remain systematically undertrained even in frontier systems.

To confirm these gaps exceed chance variation, we applied McNemar's test to all pairwise script comparisons per model (Table 8, Appendix F). The Gurmukhi-Shahmukhi gap reaches statistical significance for 8 of 10 models, with 6 models at $p < 0.001$. This holds across both frontier and open-weights systems, confirming that script bias is not a low-resource artifact. Qwen2-VL-72B-Instruct is the only model where no pairwise comparison reaches significance (minimum $p = 0.131$), consistent with its anomalous few-shot profile discussed in Section 4.3. Non-significant comparisons - for example, GPT-4o Shah vs. Rom ($p = 0.883$) and claude-sonnet-4 Shah vs. Rom ($p = 0.892$) - are not overclaimed; they reflect genuine script-pair similarity for those models. Taken together, the McNemar results establish the Script Gap as statistically robust and not an artifact of dataset size.

Model	Monoscript			Cross-Script			Mixed-Script		
	G→G	S→S	R→R	→G	→S	→R	→G	→S	→R
gpt-4o	93.9 +2.9	90.7 +4.0	92.0 +5.1	95.1 +4.1	91.2 +4.5	93.1 +6.1	94.7 +3.7	90.4 +3.7	92.5 +5.6
gemini-2.5-flash	95.5 +1.6	90.4 +1.9	93.6 +1.3	94.0 +0.1	90.0 +1.5	93.2 +0.9	94.4 +0.5	89.3 +0.8	94.7 +2.4
claude-sonnet-4	91.7 +0.3	85.6 +0.8	90.7 +4.8	90.0 −1.4	86.8 +2.0	90.8 +4.9	90.4 −1.0	86.7 +1.9	90.1 +4.3
grok-4-1-fast-reasoning	91.5 −0.5	85.9 +1.3	89.1 −1.1	92.0 0.0	85.7 +1.2	82.5 −7.6	93.1 +1.1	85.3 +0.8	82.4 −7.7
Qwen2-VL-7B-Instruct	65.6 −2.4	57.9 −0.5	64.5 −0.5	64.0 −4.0	57.7 −0.7	64.0 −1.1	65.6 −2.4	57.6 −0.8	62.4 −2.7
Llama-3.2-11B-Vision	71.7 +9.6	64.3 +18.4	65.9 +5.6	73.1 +10.9	52.9 +7.1	61.9 +1.6	70.7 +8.5	56.5 +10.7	65.1 +4.8
LLaVA-OneVision-1.5-8B-Instruct	69.6 −1.6	64.8 +1.3	62.7 +1.6	70.8 −0.4	64.5 +1.1	62.9 +1.9	69.9 −1.3	65.1 +1.6	63.5 +2.4
InternVL2_5-26B	45.9 −13.1	38.7 −6.9	53.3 −6.4	45.2 −13.7	39.2 −6.4	53.3 −6.4	45.6 −13.3	37.1 −8.5	52.3 −7.5
Kimi-VL-A3B-Instruct	47.5 −3.7	46.9 −1.6	59.5 −0.3	51.7 +0.5	48.7 +0.1	48.9 −10.8	49.9 −1.3	49.1 +0.5	47.2 −12.5
Qwen2-VL-72B-Instruct	33.6 −49.9	70.9 −8.5	80.3 0.0	51.7 −31.8	63.3 −16.1	75.9 −4.4	47.5 −36.1	70.1 −9.3	78.7 −1.6

Table 3: Few-Shot Lift (FSL): accuracy with $k = 3$ examples (subscripts show lift from zero-shot in gray). Blue indicates highest accuracy, red indicates lowest accuracy per column.

5.2 Visual Grounding Provides Parallel, Not Compensatory Gain

Model	Gurmukhi		Shahmukhi		Roman	
	T	VG	T	VG	T	VG
gpt-4o	74.1	16.8	69.9	16.8	62.4	24.5
gemini-2.5-flash	71.2	22.7	67.7	20.8	66.9	25.3
claude-sonnet-4	74.3	17.2	64.5	20.2	58.4	27.4
grok-4-1-fast-reasoning	73.6	18.4	63.2	21.3	67.2	22.9
Qwen2-VL-7B-Instruct	40.8	27.2	37.1	21.3	34.9	30.1
Llama-3.2-11B-Vision	46.9	15.2	38.4	7.5	34.7	25.6
LLaVA-OneVision-1.5-8B-Instruct	48.8	22.4	45.3	18.1	37.1	24.0
InternVL2_5-26B	36.8	22.1	32.3	13.3	32.8	26.9
Kimi-VL-A3B-Instruct	38.1	13.1	37.1	11.5	35.7	24.0
Qwen2-VL-72B-Instruct	47.5	36.1	53.1	26.4	51.7	28.5

Table 4: Modality ablation showing Text-Only accuracy (T) and Visual Gain (VG). Blue indicates highest, red indicates lowest values per column.

Table 4 reveals that visual information boosts performance uniformly (VG: 7.5–36.1%) but **does not close the script gap**. gpt-4o's VG is nearly identical across Gurmukhi (16.8%) and Shahmukhi (16.8%), yet absolute accuracies differ by 4.3%. Images provide **additive benefit**, not **compensatory repair**, the underlying orthographic fragmentation persists in multimodal settings.

Roman script shows the highest average VG (25.9%) despite often being the best-performing script in zero-shot settings. This pattern suggests models lean on memorized surface patterns in high-resource scripts while requiring visual evidence when orthographic priors are weaker. Llama-3.2-11B-Vision's substantially low Shahmukhi VG (7.5%) indicates **script-specific visual grounding failure**: even with images, it cannot integrate Perso-Arabic text effectively.

5.3 Limited Cross-Script Transfer Suggests Script-Dependent Knowledge

Table 3 and Figure 2 reveal that in-context knowledge does not transfer reliably across scripts, indicating script-locked knowledge representation.

Negative Few-Shot Lift Exposes Brittleness. Several models exhibit *performance degradation* when provided with in-context examples, revealing instability rather than adaptation under few-shot prompting. Most strikingly, Qwen2-VL-72B-Instruct shows a G→G Few-Shot Lift (FSL) of **−49.9%**: zero-shot accuracy drops from 83.5% to 33.6% when conditioned on three Gurmukhi exemplars. This catastrophic decline suggests severe in-context brittleness under low-resource script conditions. A plausible explanation is that Gurmukhi's comparatively limited representation in multimodal pretraining data prevents the model from developing stable in-context learning pathways for this orthography. Consequently, Gurmukhi exemplars may introduce tokenization-level ambiguity or activate conflicting attention patterns, causing contextual interference rather than task grounding. Importantly, the same model exhibits substantially more stable few-shot behavior in Shahmukhi and Roman Punjabi (FSL of −8.5% and 0.0%, respectively), indicating that the failure is script-specific rather than a general limitation of in-context learning.

Figure 2: Transfer Efficiency (TE) heatmap (baseline=100%).

Transfer Efficiency Reveals Anchor-Script Asymmetry. Transfer Efficiency (TE) further exposes asymmetries in cross-script in-context transfer behavior. Frontier models achieve near-perfect TE scores (98–101%), suggesting robust script-agnostic transfer under in-context prompting. In contrast, Qwen2-VL-72B-Instruct exhibits highly asymmetric transfer dynamics: $TE_{S\to G}$=154.76% and $TE_{R\to G}$=153.17% indicate that cross-script exemplars outperform same-script prompting for Gurmukhi inference, while $TE_{G\to S}$=81.20% reflects a substantial 19% degradation in transfer efficiency in the reverse direction. Similar asymmetry appears in Llama-3.2-11B-Vision, where $TE_{G\to S}$=67.22% corresponds to a **33% reduction** in cross-script transfer efficiency. Together, these results suggest that certain scripts act as stronger "anchor scripts" for in-context knowledge transfer, while multilingual multimodal knowledge appears encoded in script-specific internal representations rather than shared semantic ones.

6 Conclusion

We identify a critical blind spot in multilingual VLM evaluation: the "One Language, One Script" paradigm. Through PuMVR's 1000 parallel-script instances, we provide the first systematic evidence that state-of-the-art models exhibit substantial script-dependent bias: accuracy deltas reach 16%, SCR falls to 24.8%, and Transfer Efficiency drops below 67%. Visual grounding provides additive benefit but does not close the script gap, confirming the bias is systematic and not a modality artifact.

We propose SCR as a mandatory metric for script-agnostic evaluation and provide a methodology transferable to dozens of multi-script languages. Achieving equitable AI coverage requires moving beyond language breadth to orthographic robustness.

Limitations

Dataset Scope. Our benchmark contains 1,000 curated instances across various reasoning dimensions. Experiments were conducted on the 375-instance evaluation split described in Section 4; all reported statistics are computed exclusively on this split. While this represents the first systematic cross-script multimodal evaluation, the size constrains fine-grained statistical analysis and represents a focused subset of reasoning capabilities. We prioritized quality and semantic equivalence

over scale; future work should expand instance counts while maintaining our rigorous parallel-script methodology.

Linguistic Generalizability. We focus on Punjabi, whose three scripts provide typological diversity and minimal corpus overlap. However, script bias patterns may differ for: (1) languages with closer orthographic relationships (Serbian Cyrillic/Latin), (2) logographic systems (Chinese traditional/simplified), or (3) scripts with more balanced web representation. Our methodology is transferable, but empirical validation across diverse language families is needed before broad generalization. We provide a blueprint, not a universal law.

Evaluation Settings. We assess zero-shot and few-shot performance without fine-tuning. Script bias may behave differently under full fine-tuning with script-balanced data, instruction tuning for cross-script robustness, or retrieval-augmented architectures. Our findings reflect contemporary deployment settings but may not predict behavior under targeted mitigation strategies.

Model Coverage. We evaluate 10 contemporary VLMs representing current state-of-the-art. Findings may not generalize to future architectures with improved cross-script mechanisms, proprietary systems with undisclosed script-balancing, or domain-specific models. Our work establishes a baseline for measuring progress rather than immutable limitations.

Cultural Entanglement. Despite rigorous efforts at semantic equivalence, some instances may carry subtle script-specific cultural associations (Gurmukhi/Sikh, Shahmukhi/Islamic contexts) that are difficult to fully disentangle in a culturally grounded benchmark. This reflects the lived reality of script-culture associations but may introduce confounds beyond pure orthographic variation.

Ethics Statement

Cultural Authenticity and Representation. PuMVR was developed by native Punjabi speakers fluent in all three scripts and reviewed by community members. This ensured linguistic accuracy, cultural authenticity, and avoidance of stereotyping. Our teams lived experience with script-dependent technological barriers informed the work but may limit perspective on other multi-script contexts.

Data Sources and Privacy. All images are either AI-generated, public domain, or openly licensed. No personally identifiable information is included. Human subjects in photographs are from public historical archives or are AI-generated.

Dual-Use Considerations. This research exposes systemic bias to promote equitable AI access for multi-script communities. However, we acknowledge potential misuse: script-dependent performance insights could facilitate targeted disinformation or discrimination against low-resource script users. Our work diagnoses rather than optimizes such biases; we advocate for script-agnostic improvements that benefit all orthographies equally. We believe the benefits, improved fairness for over a billion users, substantially outweigh the risks.

Environmental Impact. Total evaluation required approximately 250 Nvidia GH200 GPU hours, consuming an estimated 193 kWh of electricity and producing about 91.4 kg CO_2 (equivalent to 368 km of passenger vehicle travel).[1] While non-trivial, this cost is reported transparently and reflects current practices for large-scale multimodal evaluation, with potential benefits for improving equity across over one billion users of multi-script languages.

Transparency. We will release all data, code, and detailed experimental protocols to enable reproducibility and community validation of our findings.

[1] Estimates assume a 700W average power draw for the GH200 Superchip and a PUE of 1.1.

References

Chantal Amrhein and Rico Sennrich. 2020. On Romanization for model transfer between scripts in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2461–2469, Online. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vuli. 2022. Iglue: A benchmark for transfer learning across modalities, tasks, and languages. *Preprint*, arXiv:2201.11732.

Ted Byrt, Janet Bishop, and John B. Carlin. 1993. Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46(5):423–429.

Soravit Changpinyo, Linting Xue, Michal Yarom, Ashish V. Thapliyal, Idan Szpektor, Julien Amelot, Xi Chen, and Radu Soricut. 2023. Maxm: Towards multilingual visual question answering. *Preprint*, arXiv:2209.05401.

Ali Faraz, Akash, Shaharukh Khan, Raja Kolla, Akshat Patidar, Suranjan Goswami, Abhinav Ravi, Chandra Khatri, and Shubham Agarwal. 2025. Indicvisionbench: Benchmarking cultural and multilingual understanding in vlms. *Preprint*, arXiv:2511.04727.

Manurag Khullar, Utkarsh Desai, Poorva Malviya, Aman Dalmia, and Zheyuan Ryan Shi. 2025. Script gap: Evaluating llm triage on indian languages in native vs roman scripts in a real world setting. *Preprint*, arXiv:2512.10780.

Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021. Visually grounded reasoning across languages and cultures. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10467–10485, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Muhammad Maaz, Hanoona Rasheed, Abdelrahman Shaker, Salman Khan, Hisham Cholakal, Rao M. Anwer, Tim Baldwin, Michael Felsberg, and Fahad S. Khan. 2024. Palo: A polyglot large multimodal model for 5b people. *Preprint*, arXiv:2402.14818.

Hoang Nguyen, Chenwei Zhang, Ye Liu, Natalie Parde, Eugene Rohrbaugh, and Philip S. Yu. 2024. CORI: CJKV benchmark with Romanization integration - a step towards cross-lingual transfer beyond textual scripts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4008–4020, Torino, Italia. ELRA and ICCL.

OpenAI. 2024. GPT-4o System Card. https://openai.com/index/gpt-4o-system-card/. Accessed: 2025-12-20.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya 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. *Preprint*, arXiv:2103.00020.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Fabian David Schmidt, Florian Schneider, Chris Biemann, and Goran Glava. 2025. Mvlsib: A massively multilingual vision-language benchmark for cross-modal topical matching. *Preprint*, arXiv:2502.12852.

Bryan Chen Zhengyu Tan, Zheng Weihua, Zhengyuan Liu, Nancy F. Chen, Hwaran Lee, Kenny Tsu Wei Choo, and Roy Ka-Wei Lee. 2025. Blend-vis: Benchmarking multimodal cultural understanding in vision language models. *Preprint*, arXiv:2510.11178.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Ashish V. Thapliyal, Jordi Pont Tuset, Xi Chen, and Radu Soricut. 2022. Crossmodal-3600: A massively multilingual multimodal evaluation dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 715–729, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ashmal Vayani, Dinura Dissanayake, Hasindri Watawana, Noor Ahsan, Nevasini Sasikumar, Omkar Thawakar, Henok Biadglign Ademtew, Yahya Hmaiti, Amandeep Kumar, Kartik Kuckreja, Mykola Maslych, Wafa Al Ghallabi, Mihail Mihaylov, Chao Qin, Abdelrahman M Shaker, Mike Zhang, Mahardika Krisna Ihsani, Amiel Esplana, Monil Gokani, and 50 others. 2025. All languages matter: Evaluating lmms on culturally diverse 100 languages. *Preprint*, arXiv:2411.16508.

Da Yin, Liunian Harold Li, Ziniu Hu, Nanyun Peng, and Kai-Wei Chang. 2021. Broaden the vision: Geo-diverse visual commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2115–2129, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Annotation Protocol Details

Each instance was assessed across five dimensions: (i) semantic equivalence across all three scripts, (ii) answer correctness, verified by independently inspecting the image and question without consulting the provided answers, and script accuracy for (iii) Gurmukhi, (iv) Shahmukhi, and (v) Roman Punjabi. Annotators were instructed to form independent judgments before recording any label and to apply a conservative criterion, marking *No* in cases of genuine uncertainty rather than defaulting to *Yes*. This conservative protocol is important for interpreting the results: the near-universal agreement observed reflects genuine expert consensus reached under an instruction regime explicitly designed to surface disagreement, rather than annotator passivity or a low-effort default. A stratified sample of 375 instances drawn from the full benchmark was selected to match the evaluation split used in all three experiments, enabling agreement figures to be interpreted directly with respect to experimental coverage.

B Model Configurations

We provide detailed specifications for all evaluated models to ensure reproducibility. Table 5 summarizes the key hyperparameters used across all experiments.

B.1 Hardware Infrastructure

API-Based Models: All proprietary models were accessed via their respective APIs with no local computational requirements. claude-sonnet-4 was accessed through Google Cloud's Vertex AI service (us-east5 region).

Local Models : Evaluated on NVIDIA GH200 Grace Hopper Superchip with:

- **GPU Memory:** 120GB HBM3
- **CUDA Version:** 12.1+
- **Framework:** PyTorch 2.1.0+
- **Memory Management:** `device_map = "cuda"` with `max_memory = {0: "110GB"}`
- **Optimization:** `low_cpu_mem_usage = True`, `trust_remote_code = True`

B.2 Model-Specific Implementation Details

Qwen2-VL Series (7B & 72B):

- **Architecture:** `Qwen2VLForConditionalGeneration`
- **Tokenizer:** `AutoProcessor` with custom Qwen2 tokenizer
- **Image Processing:** Automatic via processor's `apply_chat_template`
- **Input Format:** List-of-dicts conversation structure with image and text content blocks
- **Generation:** Output token trimming (remove input length from generated sequences)

Llama-3.2-11B-Vision:

- **Architecture:** `MllamaForConditionalGeneration`
- **Processor:** `AutoProcessor` (handles `MllamaProcessor` automatically)
- **Image Handling:** Passed to processor via `images` parameter
- **Special Tokens:** `<|image|>` tokens inserted automatically by `apply_chat_template`
- **Decoding:** Input tokens trimmed from output: `output_ids[len(input_ids):]`

LLaVA-OneVision-1.5-8B-Instruct:

- **Architecture:** `AutoModelForCausalLM` (requires `trust_remote_code=True`)
- **Vision Backbone:** Custom `rice_vit` architecture
- **Processor:** `AutoProcessor` with batch processing
- **Input Format:** Text list + image list with automatic alignment
- **Decoding:** `skip_special_tokens=True`, `clean_up_tokenization_spaces=True`

InternVL2_5-26B:

Model	Organization	Params	Precision	Temp.	Max Tok.	Seed	Inference
Proprietary Frontier Models							
gpt-4o	OpenAI	–	Native	0.1	128	42	API
gemini-2.5-flash	Google DeepMind	–	Native	0.1	128	42	Vertex AI
claude-sonnet-4	Anthropic	–	Native	0.1	128	–	Vertex AI
grok-4-1-fast-reasoning	xAI	–	Native	0.1	128	42	API
Open-Weights Models							
Qwen2-VL-7B-Instruct	Alibaba	7B	bfloat16	0.1	128	42	Local
Qwen2-VL-72B-Instruct	Alibaba	72B	bfloat16	0.1	128	42	Local
Llama-3.2-11B-Vision	Meta	11B	bfloat16	0.1	128	42	Local
LLaVA-OneVision-1.5-8B-Instruct	LMMS-Lab	8B	bfloat16	0.1	128	42	Local
InternVL2_5-26B	OpenGVLab	26B	bfloat16	0.1	128	42	Local
Kimi-VL-A3B-Instruct	Moonshot AI	3B	auto	0.2	128	42	Local

Table 5: Model configuration summary. All models use greedy decoding (temperature ≤ 0.2). "Local" denotes inference on NVIDIA GH200 with CUDA.

- **Architecture:** `AutoModel` (requires `trust_remote_code=True`)
- **Dynamic Resolution:** Custom preprocessing with aspect ratio preservation
- **Image Preprocessing:** BICUBIC interpolation to 448×448 tiles (min_num=1, max_num=12)
- **Thumbnail Strategy:** Appended for multi-tile images (use_thumbnail=True)
- **Generation:** Native `.chat()` method with pixel_values tensor stack
- **Flash Attention:** Enabled via `use_flash_attn=True`

Kimi-VL-A3B-Instruct:

- **Architecture:** `AutoModelForCausalLM`
- **Processor:** Dual tokenizer setup (main tokenizer + processor tokenizer for chat template)
- **Chat Template:** Applied via `processor.tokenizer.apply_chat_template()`
- **Input Structure:** Type-annotated content list: `{"type": "image"}`, `{"type": "text"}`
- **Temperature:** 0.2
- **Token Management:** `pad_token_id` fallback to `eos_token_id` if unavailable

API Models (gpt-4o, gemini-2.5-flash, claude-sonnet-4, grok-4-1-fast-reasoning):

- **Image Encoding:** Base64 JPEG encoding (RGB conversion applied if input is RGBA/other modes)
- **gpt-4o:** OpenAI SDK with `data : image/jpeg;base64` URL format
- **gemini-2.5-flash:** Google Generative AI SDK with `[prompt_text, pil_image]` input list
- **Gemini Safety:** All harm categories set to `BLOCK_NONE` to prevent refusal on research data
- **claude-sonnet-4:** Anthropic Vertex AI client with base64 source in content blocks
- **grok-4-1-fast-reasoning:** OpenAI-compatible SDK (`base_url="https://api.x.ai/v1"`) with image_url format

B.3 Reproducibility Safeguards

Random Seeds: All local models initialize with `seed=42`:

```
torch.manual_seed(42)
torch.cuda.manual_seed_all(42)
```

API Seeds: Where supported (gpt-4o, gemini-2.5-flash, grok-4-1-fast-reasoning), `seed=42` parameter passed to API calls.

Incremental Saving: Results saved immediately after each prediction via CSV append mode to prevent data loss on system crashes.
Resume Capability: All evaluators check for existing CSV files and skip completed instance IDs, enabling seamless resumption after interruptions.
Deterministic Decoding: Temperature ≤ 0.2 with `do_sample=(temperature > 0)` ensures greedy/near-greedy decoding for reproducibility.

C Complete Results Tables

C.1 Experiment 1: Per-Script Accuracy Details

Table 6 presents the complete accuracy results across all three scripts for each model.

Model	Gur.	Shah.	Rom.	SCR
gpt-4o	90.93	86.67	86.93	78.13
gemini-2.5-flash	93.87	88.53	92.27	85.60
claude-sonnet-4	91.47	84.76	85.83	77.09
grok-4-1-fast-reasoning	92.00	84.53	90.13	79.73
Qwen2-VL-7B-Instruct	68.00	58.40	65.07	41.33
Llama-3.2-11B-Vision	62.13	45.87	60.27	27.47
LLaVA-OneVision-1.5-8B-Instruct	71.20	63.47	61.07	41.33
InternVL2_5-26B	58.93	45.60	59.73	28.00
Kimi-VL-A3B-Instruct	51.20	48.53	59.73	24.80
Qwen2-VL-72B-Instruct	83.54	79.43	80.25	54.67
Average	**76.32**	**68.58**	**74.13**	**53.82**

Table 6: Zero-shot accuracy (%) across scripts. SCR = Script Consistency Rate.

Confidence Intervals. Wilson score 95% confidence intervals on per-script accuracy across 375 instances are narrow throughout: for frontier models (accuracy >84%), the margin is ±1.8–3.6 percentage points; for open-weights models (accuracy 45–72%), ±2.5–5.1 points. These intervals do not overlap for the Gurmukhi–Shahmukhi pairs flagged as significant by McNemar's test, confirming that reported deltas exceed sampling variation.

C.2 Experiment 3: Transfer Efficiency Matrix

Table 9 shows Transfer Efficiency (TE) values for all few-shot transfer conditions.

D Full Experimental Prompts

This section provides the exact prompt templates used across all experiments. To ensure high-fidelity evaluation, we enforced strict formatting constraints.

D.1 Notation and Placeholder Definitions

To maintain clarity across the templates provided in this appendix, we use the following placeholders to denote dynamically injected content:

- `{question}`: The Punjabi query translated into the specific target script (Gurmukhi, Shahmukhi, or Roman).
- `{formatted_options}`: A list of four multiple-choice candidates. Each candidate is presented on a new line without alphabetical prefixes (A, B, C, D) or numerical markers.
- `{script_name}`: The English name of the target script (e.g., "Gurmukhi", "Shahmukhi", or "Roman").
- `{Script Instruction}`: A script-specific directive used in Experiments to trigger internal script-specific reasoning. The values used were:
  - **Gurmukhi:** "ਗੁਰਮੁਖੀ ਵਿੱਚ"
  - **Shahmukhi:** "وچ مکھی شاہ"
  - **Roman:** "in Roman script"

D.2 Design Rationale: Text-Based vs. Letter-Based Selection

In all experiments, models were instructed to output the *exact text* of the correct option rather than a single identifier (e.g., "A", "B"). This design choice was made for two primary reasons:

1. **Script Comprehension Verification:** A model might correctly guess a letter (25% probability) without truly processing the script. Forcing the model to reproduce the script-specific text ensures it can parse and generate the target orthography.
2. **Failure Mode Analysis:** By requiring text output, we could identify cases where

models produced "hallucinated" characters or mixed scripts, data that would be lost if restricted to single-letter outputs.

D.3 Experiment 1: Baseline Script Gap

The primary benchmark used the following template with English instructions and script-specific constraints.

Exp 1: Standard Prompt

```
Question: {question}

Options:
{formatted_options}

Task: Identify the correct option based on the image.

Constraint: {Answer in [Script Name] script.} Copy the exact text of the correct option. Do not explain.

CRITICAL RULES:

1. Answer MUST be in the SAME script as the question
2. Copy EXACTLY one option from above - character by character
3. NO explanations, NO extra words, NO English translations
4. NO letters like A), B), C) or numbers
5. Output ONLY the option text, nothing else

Your answer (copy exact text from options):
Answer:
```

D.4 Experiment 2: Native Instruction Prompting

This experiment used similar prompts as experiment 1, without the image input to test model performance and establish a baseline.

Exp 2: Native Script Instructions

```
Question {Script Instruction}: {question}

Options:
{formatted_options}

CRITICAL RULES:

1. Answer MUST be in the SAME script as the question
2. Copy EXACTLY one option from above - character by character
3. NO explanations, NO extra words, NO English translations
4. NO letters like A), B), C) or numbers
5. Output ONLY the option text, nothing else

Your answer (copy exact text from options):
```

D.5 Experiment 3: System Prompting (Few-Shot)

For experiments involving system-level instructions, the following persona-based prompt was utilized.

Exp 3: System Prompt

```
You are a precise answering assistant.
You will be given a visual question and options.
You must output ONLY the exact text of the correct option in {Script Name} script.

CRITICAL RULES:
1. NO reasoning.
2. NO explanations.
3. NO option letters (like A, B).
4. Output strictly the option text.
```

E Error Classification Details

Table 7 reports the per-script breakdown of error types across all 11,250 model responses (10 models $\times$ 375 instances $\times$ 3 scripts), confirming that the generative evaluation design measures comprehension rather than output formatting difficulty.

Script	Comp. Failures	Empty	Formatting
Gurmukhi	99.8%	0.2%	0.0%
Shahmukhi	99.3%	0.7%	0.0%
Roman	99.7%	0.3%	0.0%
Aggregate	**99.59%**	**0.41%**	**0.00%**

Table 7: Error classification across all model responses by script. Comprehension failures are complete wrong-option selections; formatting artifacts include mixed-script or hallucinated characters.

At the model level, claude-sonnet-4 showed the lowest comprehension failure rate at 96.55%, with 3.45% empty responses constituting the sole notable outlier. All other models showed comprehension failure rates at or above 98.77%.

F McNemar Test Results

Table 8 reports McNemar's test results for all pairwise script comparisons across all 10 models. We apply McNemar's test treating per-instance correctness as paired binary observations across 375 instances. For each model, we test three script pairs: Gurmukhi vs. Shahmukhi, Gurmukhi vs. Roman, and Shahmukhi vs. Roman. The chi-squared variant is used when discordant pairs exceed 25; the exact variant otherwise.

Several patterns emerge. First, the Gurmukhi–Shahmukhi gap is the most consistently significant pair across models, confirming Shahmukhi as the most undertrained script. Second, the Gurmukhi–Roman gap is significant for some models (claude-sonnet-4, LLaVA) but not others, suggesting variable Roman script coverage across model families. Third, Qwen2-VL-72B-Instruct is the only model where no pair reaches significance, consistent with its more balanced cross-script performance observed in Table 6. These results validate that the Script Gap reported in Section 5.1 is statistically robust and not an artifact of the dataset size.

Model	Script Pair	p-value	Sig.
gpt-4o	Gur vs Shah	0.025	*
gpt-4o	Gur vs Rom	0.041	*
gpt-4o	Shah vs Rom	0.883	ns
gemini-2.5-flash	Gur vs Shah	0.001	**
gemini-2.5-flash	Gur vs Rom	0.146	ns
gemini-2.5-flash	Shah vs Rom	0.038	*
claude-sonnet-4	Gur vs Shah	<0.001	***
claude-sonnet-4	Gur vs Rom	0.003	**
claude-sonnet-4	Shah vs Rom	0.892	ns
grok-4-1-fast-reasoning	Gur vs Shah	<0.001	***
grok-4-1-fast-reasoning	Gur vs Rom	0.265	ns
grok-4-1-fast-reasoning	Shah vs Rom	0.006	**
Qwen2-VL-7B-Instruct	Gur vs Shah	<0.001	***
Qwen2-VL-7B-Instruct	Gur vs Rom	0.343	ns
Qwen2-VL-7B-Instruct	Shah vs Rom	0.044	*
Llama-3.2-11B-Vision	Gur vs Shah	<0.001	***
Llama-3.2-11B-Vision	Gur vs Rom	0.608	ns
Llama-3.2-11B-Vision	Shah vs Rom	<0.001	***
LLaVA-OneVision-1.5-8B-Instruct	Gur vs Shah	0.004	**
LLaVA-OneVision-1.5-8B-Instruct	Gur vs Rom	<0.001	***
LLaVA-OneVision-1.5-8B-Instruct	Shah vs Rom	0.415	ns
InternVL2_5-26B	Gur vs Shah	<0.001	***
InternVL2_5-26B	Gur vs Rom	0.857	ns
InternVL2_5-26B	Shah vs Rom	<0.001	***
Kimi-VL-A3B-Instruct	Gur vs Shah	0.430	ns
Kimi-VL-A3B-Instruct	Gur vs Rom	0.006	**
Kimi-VL-A3B-Instruct	Shah vs Rom	0.001	**
Qwen2-VL-72B-Instruct	Gur vs Shah	0.131	ns
Qwen2-VL-72B-Instruct	Gur vs Rom	0.208	ns
Qwen2-VL-72B-Instruct	Shah vs Rom	0.892	ns

Table 8: McNemar's test across all script pairs and models. *** $p<0.001$, ** $p<0.01$, * $p<0.05$, ns = not significant.

Model	S→G	R→G	G→S	R→S	G→R	S→R	Avg. TE
gpt-4o	100.85	101.70	100.88	100.29	101.45	100.87	**101.01**
gemini-2.5-flash	98.04	98.88	99.12	100.00	99.15	100.00	99.20
claude-sonnet-4	96.80	99.42	101.25	101.56	99.71	100.54	99.88
grok-4-1-fast-reasoning	100.29	100.87	100.31	99.38	89.52	95.81	97.70
Qwen2-VL-7B-Instruct	97.56	97.56	102.76	96.77	100.00	98.35	98.83
Llama-3.2-11B-Vision	101.86	101.86	67.22	97.51	87.85	100.00	92.72
LLaVA-OneVision-1.5-8B-Instruct	103.07	100.38	102.06	97.12	100.00	100.85	100.58
InternVL2_5-26B	96.51	100.58	102.76	100.00	100.00	100.00	99.98
Kimi-VL-A3B-Instruct	107.87	110.11	101.70	105.68	76.23	88.34	98.32
Qwen2-VL-72B-Instruct	154.76	153.17	81.20	97.37	88.70	100.33	112.59
Average	**105.76**	**106.45**	**95.93**	**99.57**	**94.26**	**98.51**	**100.08**

Table 9: Transfer Efficiency (%) across script pairs. Notation: G=Gurmukhi, S=Shahmukhi, R=Roman.