It's Time to Get It Right: Improving Analog Clock Reading and Clock-Hand Spatial Reasoning in Vision-Language Models

Supplementary Material

A. Dataset Analysis

This section provides detailed statistical analysis of the TickTockVQA dataset, including data source composition, temporal distribution patterns, and filtering strategies employed to ensure dataset quality.

A.1. Data Source Composition and Train/Test Split

As described in Section 3 of the main paper, TickTockVQA is collected from seven diverse sources. Table S1 provides the complete breakdown of sample counts per source and their assignment to train/test splits.

A.2. Temporal Distribution Analysis

We analyze the distribution of annotated times across all 12 hours and 60 minutes to characterize both inherent biases and the coverage achieved through our filtering pipeline. Figure S1 presents a two-dimensional heatmap showing the density of labeled times. The distribution is generally uniform, with noticeable concentration around aesthetically preferred times such as 10:10. This bias reflects the prevalence of such times in product photography and stock images.

A.3. Marginal Distribution Analysis

Figure S2 provides a detailed breakdown of hour and minute distributions. The hour distribution (Figure S2a) shows that hours 10, 11, and 12 are slightly overrepresented due to the 10:10 bias. However, all hours retain substantial coverage (minimum 754 samples for hour 6, maximum 1,759 for hour 10), with a coefficient of variation of 26.9%, indicating reasonable balance.

The minute distribution (Figure S2b) reveals that canonical positions (0, 10, 15, 30, 45) occur more frequently than arbitrary minutes. Our filtering process substantially reduces these imbalances compared to raw web-crawled data, but residual skew toward common clock hand positions remains. Critically, all 60 minutes are represented in the dataset, ensuring coverage of fine-grained temporal reading challenges.

B. Performance Analysis Across Clock Types and Conditions

This section provides granular performance analysis of our ITGR model across different clock types, environmental conditions, and design variations. These analyses reveal which factors most significantly impact clock reading accuracy.

Table S1. **TickTockVQA data source composition and train/test split.** Only COCO, Open Images, and Clock Movies are used for testing; all other sources are reserved for training. This separation ensures evaluation on out-of-distribution sources.

Source	Images	Split
Test Sources		
COCO	2,063	Test
Open Images (OID)	1,940	Test
Clock Movies	1,244	Test
Training Sources		
Visual Genome	1,246	Train
SBU Captions	1,533	Train
CC12M	3,677	Train
ImageNet	780	Train
Total (Test)	**5,247**	**Test**
Total (Train)	**7,236**	**Train**
Grand Total	**12,483**	**All**

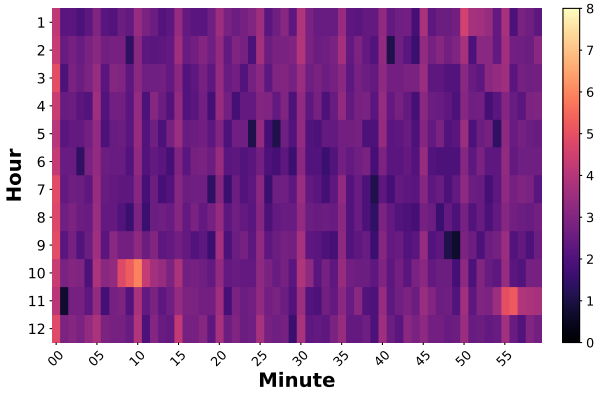
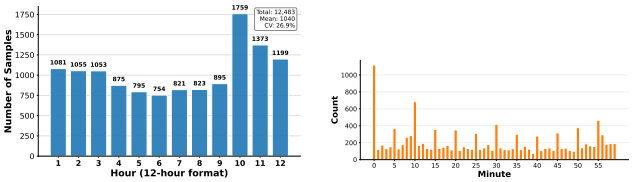


Figure S1. **Clock annotation density heatmap.** Distribution of labeled times across all hours (1–12) and minutes (0–59). Color intensity represents $\ln(1 + \text{count})$ to balance visibility across frequency ranges. Darker regions indicate higher sample density, with notable concentration around 10:10.

B.1. Performance by Clock Type

Figure S3 presents the breakdown of ITGR (Llama-3.2-11B with Swap-DPO) performance across seven clock categories. Performance varies dramatically, ranging from 27.99% (wristwatches) to 62.71% (graphic/illustrated clocks), revealing significant differences in task difficulty.



(a) Hour distribution (1–12) (b) Minute distribution (0–59)

Figure S2. **Marginal temporal distributions.** (a) Hour distribution shows reasonable balance with CV=26.9%. Hour 10 is overrepresented due to the 10:10 aesthetic bias. (b) Minute distribution reveals expected peaks at canonical positions (0, 15, 30, 45) but maintains coverage across all 60 minutes.

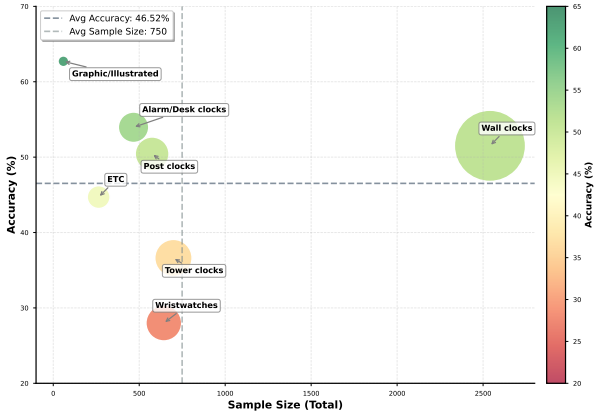


Figure S3. **ITGR accuracy breakdown by clock type.** Performance varies significantly across categories. Graphic/Illustrated clocks achieve the highest accuracy (62.71%) due to high contrast and clean contours. Wristwatches show the lowest performance (27.99%) due to small size, glass reflections, and occlusion. Bubble size represents sample count for each category. The dashed line indicates overall average accuracy (46.52%).

Key Observations:

- **Graphic/Illustrated clocks (62.71%):** Highest performance due to high contrast, clean contours, and minimal background clutter. These clocks typically appear in controlled settings with frontal viewpoints.
- **Wristwatches (27.99%):** Lowest performance despite substantial training data (1,238 samples). Challenges include: (1) small clock face size in images, (2) glass reflections obscuring hands, (3) depth-of-field blur, (4) curved surfaces causing distortion, and (5) frequent hand overlap at small scales.
- **Wall clocks (50.60%):** Despite being the largest category (4,046 samples), performance is moderate. This indicates that simply scaling data does not guarantee improved performance; visual complexity in real-world wall clock scenarios (varied lighting, viewing angles, occlusion) poses persistent challenges.
- **Tower clocks (44.66%):** Moderate performance. Challenges include extreme viewing angles, atmospheric effects, and distance-related image quality degradation.
- **Alarm/Desk clocks (47.63%):** Performance similar to wall clocks, benefiting from typically frontal viewing angles but challenged by reflective surfaces and small digital displays that can distract the model.

Implications: The 35pp performance gap between the easiest and hardest categories demonstrates that clock reading difficulty depends heavily on physical form factor and imaging conditions, not merely on the number of training samples.

B.2. Performance by Environmental Conditions, Transformations, and Design

Figure S4 decomposes ITGR performance across three categorical dimensions: (a) environment (indoor/outdoor/unknown), (b) geometric transformation (normal/flipped/partial), and (c) clock face design (Arabic/Roman/no numerals).

Environmental Robustness (Figure S4a): The model demonstrates stable performance across different environmental settings: indoor (48.5%, n=2,244), outdoor (44.9%, n=2,544), and unknown (45.8%, n=459). This 3.6pp variation suggests that background context, ambient lighting, and scene clutter alone do not significantly destabilize predictions. The model has learned to focus on the clock itself rather than being distracted by environmental factors.

Transformation Sensitivity (Figure S4b): The model exhibits severe degradation for transformed clocks:

- **Normal orientation (46.9%, n=5,089):** Baseline performance
- **Flipped/rotated (23.1%, n=12):** 50% relative performance drop, indicating brittleness to non-canonical orientations. The model struggles when clocks are horizontally flipped or rotated, suggesting it has learned orientation-specific features rather than rotation-invariant representations.
- **Partial/occluded (37.3%, n=146):** 20% relative drop, showing sensitivity to missing information even when hands remain visible.

Design Dependency (Figure S4c): Performance varies by clock face design:

- **Arabic numerals (48.2%, n=2,296):** Slightly better, likely due to clearer spatial references
- **Roman numerals (46.8%, n=1,885):** Comparable performance, indicating successful generalization across numeral systems
- **No numerals (36.2%, n=1,121):** 25% relative drop, revealing reliance on hour markers for spatial reasoning. Without explicit markers, the model must infer positions purely from hand angles, which is more challenging.

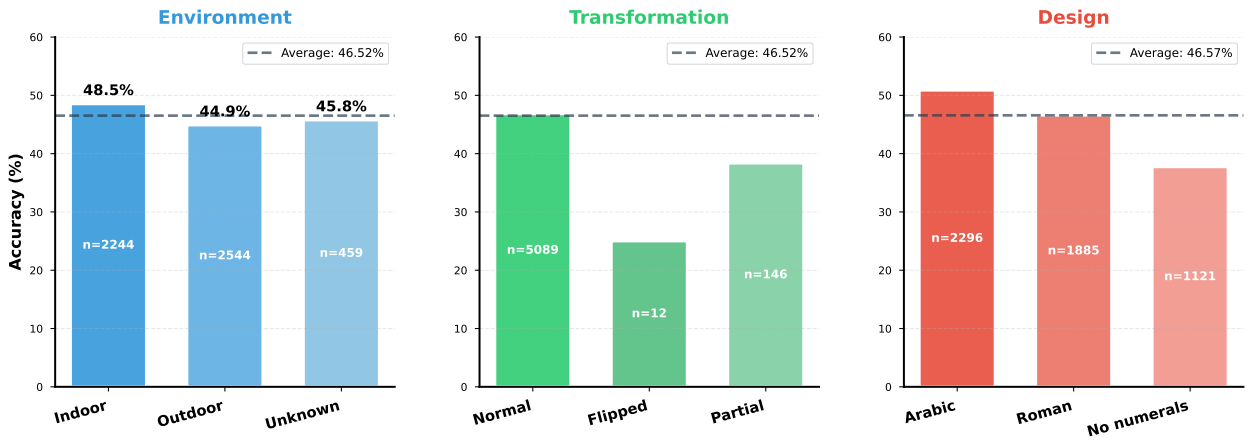


Figure S4. **ITGR accuracy breakdown across three categorical dimensions.** (a) **Environment:** Performance is stable across indoor (48.5%), outdoor (44.9%), and unknown (45.8%) settings, indicating robustness to background context and lighting variation. (b) **Transformation:** Severe degradation for flipped clocks (23.1%), revealing fragility to unusual orientations. Partial occlusion (37.3%) also degrades performance. (c) **Design:** Performance is consistent across Arabic (48.2%) and Roman (46.8%) numerals but degrades for clocks without numerals (36.2%), suggesting reliance on numerical markers for spatial reference. Dashed lines indicate overall average accuracy.

B.3. Key Findings and Implications

1. **Environmental robustness vs. structural fragility:** While ITGR is robust to environmental variation (lighting, background), it remains highly sensitive to structural deviations (flipped orientation, missing numerals). This suggests that future work should focus on geometric augmentation and rotation-equivariant architectures.
2. **Data scaling is insufficient:** Wall clocks, despite having the most training samples (4,046), achieve only moderate accuracy (50.60%). This confirms that visual complexity and physical form factor are more important than sample quantity.
3. **Form factor matters most:** The 35pp gap between clock types (wristwatches vs. graphic clocks) far exceeds the 4pp gap from environmental conditions, indicating that physical form factor is the dominant difficulty factor.
4. **Future directions:** Improving performance on low-accuracy categories (wristwatches, flipped clocks, no-numeral designs) through specialized data augmentation, multi-scale processing, and rotation-invariant features represents a promising research direction.

C. Effect of Swap-DPO on Hand Confusion

This section analyzes how our Swap-DPO method specifically addresses the hand-swapping problem through targeted preference learning. We compare three training strategies: (1) SFT only, (2) Random-DPO (baseline DPO with random error correction), and (3) Swap-DPO (our proposed method).

C.1. Experimental Setup

For each model (Qwen2.5-VL-7B, Gemma3-12B, Llama-3.2-11B), we train three variants:

- **SFT:** Supervised fine-tuning on TickTockVQA training set (7,236 samples) for 10 epochs
- **Random-DPO:** SFT + DPO with randomly selected incorrect predictions as rejected responses
- **Swap-DPO:** SFT + DPO with geometrically swapped times as rejected responses (our method)

We evaluate each variant under two metrics:

- **Baseline (B):** Standard full-time accuracy (hour and minute must both be correct)
- **Swap-equivalence (S):** Accuracy when allowing hour/minute hand swaps (e.g., predicting 06:18 for ground truth 03:30 counts as correct).

The gap $\Delta = S - B$ directly quantifies hand-swap confusion: a larger gap indicates more frequent role reversal errors.

C.2. Quantitative Results

Table S2 presents full-time accuracy across all three training strategies. The results consistently demonstrate that Swap-DPO reduces hand-swap confusion while improving overall accuracy.

C.3. Key Observations

1. SFT establishes strong baseline but exhibits hand confusion: Supervised fine-tuning substantially improves performance across all models (e.g., Llama: 1.41% zero-shot $\rightarrow$ 45.8% SFT). However, a persistent 2.32–2.90pp gap between B and S metrics indicates that 5–7% of errors are

Table S2. **Full-time accuracy across SFT, Random-DPO, and Swap-DPO.** B denotes Baseline accuracy (strict), S denotes Swap-equivalence accuracy (allowing hand swaps), and Δ is the hand-swap gap ($S - B$). A smaller Δ indicates less hand confusion. Swap-DPO consistently reduces Δ compared to both SFT and Random-DPO while improving overall accuracy.

	SFT			Random-DPO			Swap-DPO		
Model	**B**	**S**	**Δ**	**B**	**S**	**Δ**	**B**	**S**	**Δ**
Qwen2.5-VL-7B	20.3	22.8	+2.42	20.6	23.4	+2.75	**23.1**	**25.1**	**+2.02**
Gemma3-12B	34.2	37.1	+2.90	35.0	38.0	+3.00	**35.3**	**37.9**	**+2.57**
Llama-3.2-11B	45.8	48.1	+2.32	45.6	47.7	+2.06	**46.2**	**48.5**	**+2.26**
Average across models:									
Mean	33.4	36.0	+2.55	33.7	36.4	+2.60	**34.9**	**37.2**	**+2.28**

pure hand-swap mistakes where the model has correctly localized both hands but assigned incorrect semantic roles.

2. Random-DPO fails to reduce hand confusion: Surprisingly, applying standard DPO with randomly selected incorrect predictions as rejected responses *increases* the hand-swap gap in 2 out of 3 models (Qwen: +2.42 → +2.75, Gemma: +2.90 → +3.00). This counterintuitive result suggests that generic error correction makes the model *more* sensitive to hand ambiguity. We hypothesize that random negative samples lack the geometric consistency needed to teach hand role distinction; the model learns to avoid diverse errors but does not specifically learn which hand is which.

3. Swap-DPO consistently reduces hand confusion: Our Swap-DPO method, which uses geometrically swapped times as rejected responses, achieves three critical improvements:

- **Reduced hand-swap gap:** Average Δ decreases from 2.55 (SFT) to 2.28 (Swap-DPO), a 10.6% relative reduction. For Qwen, the reduction is 16.5% (2.42 → 2.02).
- **Improved overall accuracy:** Baseline accuracy improves across all models (average: 33.4% → 34.9%), demonstrating that resolving hand confusion generalizes to other error types.
- **Consistency across architectures:** Swap-DPO outperforms Random-DPO on all three models, indicating robustness of the approach.

C.4. Why Does Swap-DPO Work?

The effectiveness of Swap-DPO stems from its geometric consistency:

1. **Contrastive hand role learning:** By presenting the model with two geometrically plausible interpretations of the same clock (correct vs. swapped), we force it to learn which visual features (hand length, thickness, position) correspond to which semantic role (hour vs. minute).
2. **Hard negative mining:** Swapped times are "hard negatives" because they are geometrically consistent with the visual input but semantically incorrect. This is more informative than random wrong times, which may be geometrically implausible.
3. **Explicit disambiguation signal:** Unlike SFT, which only provides positive examples, Swap-DPO explicitly teaches what *not* to predict, specifically targeting the most common failure mode.

C.5. Limitations and Remaining Challenges

Despite Swap-DPO's improvements, a 2.0–2.6pp hand-swap gap persists, indicating that 4–6% of errors remain pure hand-swap confusions. This suggests:

- **Ambiguous cases:** Some clocks have nearly identical hand lengths or poor image quality, making disambiguation genuinely difficult even for humans.
- **Model capacity:** Current VLM architectures may lack sufficient fine-grained spatial reasoning capabilities to perfectly distinguish hands in all scenarios.
- **Dataset bias:** The 2–3% residual gap may represent an upper bound given inherent ambiguities in real-world analog clocks.

Future work could explore: (1) multi-stage reasoning (explicit hand detection → role assignment → time reading), (2) uncertainty quantification to flag ambiguous cases, and (3) contrastive pre-training on synthetic clock data with perfect hand labels.

D. Implementation Details

This section provides comprehensive implementation details to ensure full reproducibility of our experiments. We report all hyperparameters, training configurations, and computational requirements for the three VLM backbones used in our study.

D.1. DPO Training Configuration

Table S3 summarizes the complete DPO training configuration across all three model architectures. We employ a consistent training strategy with minor architecture-specific adjustments to accommodate different model characteristics.

Architecture-Specific Configuration Notes:

Table S3. **Complete DPO training hyperparameters and configurations.** We report all settings used for Direct Preference Optimization across three VLM backbones. Model-specific differences are highlighted. All models use 8× NVIDIA A6000 GPUs (48GB each).

Configuration	Qwen2.5-VL-7B	Llama-3.2-11B	Gemma3-12B	Notes
DPO-Specific Parameters				
Loss function	sigmoid	sigmoid	sigmoid	Standard DPO loss
β (temperature)	0.3	0.3	0.3	Controls preference strength
Precompute ref. logprobs	false	false	false	Compute on-the-fly
LoRA Configuration				
LoRA enabled	✗	✓	✓	Qwen uses full fine-tuning
LoRA rank (r)	—	64	64	—
LoRA alpha (α)	—	64	64	$\alpha = r$ for stability
LoRA dropout	—	0.05	0.05	—
Target modules	—	all linear	all linear	Except embeddings/LM head
Vision LoRA	—	✓	✓	Apply LoRA to vision tower
DoRA	—	✗	✗	Standard LoRA
Batch Size & Parallelization				
Global batch size	256	256	256	Effective batch size
Batch per device	4	8	4	Per-GPU batch size
Gradient accum. steps	8	4	8	$= 256/(\text{batch} \times \text{GPUs})$
Num. devices	8	8	8	NVIDIA A6000 (48GB)
Optimization Hyperparameters				
Num. epochs	4	4	4	Consistent across models
Learning rate (LLM)	2e-6	2e-6	2e-6	Base LLM learning rate
Learning rate (vision)	2e-6	2e-6	2e-6	Vision tower learning rate
Learning rate (projector)	1e-5	1e-5	1e-5	5× higher for projector
Weight decay	0.1	0.1	0.1	AdamW regularization
Adam β_1	0.9	0.9	0.9	Default
Adam β_2	0.95	0.95	0.95	Slightly lower than default
Warmup ratio	0.03	0.03	0.03	3% of total steps
LR scheduler	cosine	cosine	cosine	Cosine annealing to 0
Memory & Precision				
Mixed precision	bfloat16	bfloat16	bfloat16	Training dtype
FP16	✗	✗	✗	Use bfloat16 instead
TF32	✓	✓	✓	NVIDIA Ampere+ acceleration
Gradient checkpointing	✓	✓	✓	Recompute activations
DeepSpeed stage	ZeRO-3	ZeRO-3	ZeRO-3	Partition optimizer states
Flash Attention 2	✓	✓	✗	Gemma3: eager attention
Liger kernel	✓	✓	✓	Fused RMSNorm + cross-entropy
Module Freezing Strategy				
Freeze vision tower	✗	✓	✓	Qwen: full fine-tuning
Freeze LLM	✗	✓	✓	Qwen: full fine-tuning
Freeze projector	✗	✗	✗	Always trainable

(1) Qwen2.5-VL-7B: We apply full fine-tuning (no LoRA) due to its relatively small size (7B parameters) and efficient architecture. Qwen uses dynamic resolution processing with configurable min/max pixels (401K–1003K), allowing adaptive handling of various image sizes. Flash Attention 2 is enabled for memory efficiency. The model's native support for variable-resolution inputs eliminates the need for fixed-size preprocessing.

(2) Llama-3.2-11B: We employ LoRA (rank 64, alpha 64) on all linear layers including the vision tower to reduce memory footprint. The larger per-device batch size (8 vs. 4 for Qwen/Gemma) is possible due to LoRA's parameter efficiency—only $\sim$2% of parameters are trainable. Lazy preprocessing (on-the-fly image loading) accelerates training. Flash Attention 2 is supported and enabled.

(3) Gemma3-12B: Similar to Llama, we use LoRA (rank 64, alpha 64) for memory efficiency. However, we use *eager attention* instead of Flash Attention 2, as recommended by the Gemma3 technical report due to numerical stability concerns with certain attention patterns. DoRA (weight-decomposed LoRA) is disabled for training stability. We disable lazy preprocessing to ensure deterministic image loading order.

Common Configuration Rationale: All models share core settings: sigmoid DPO loss with $\beta = 0.3$ (stronger preference signal than default 0.1), global batch size of 256 (necessary for stable DPO training), and 4 training epochs (sufficient for convergence without overfitting). We use ZeRO-3 (partitioned optimizer states and gradients) with bfloat16 mixed precision and gradient checkpointing to enable training on 8$\times$ A6000 GPUs. The projector module always receives a 5$\times$ higher learning rate (1e-5 vs. 2e-6) because it bridges frozen/slowly-adapted vision features to the language model and requires faster adaptation.

D.2. DPO Preference Data Generation

We automatically generate DPO preference pairs by running inference with the SFT model on the training set. This section provides implementation details and configuration parameters beyond what is described in the main paper (Algorithm 1).

D.2.1. Inference Configuration for Data Generation

We run SFT model inference on all 7,236 training images using the following configuration:

- **Batch size:** 16 (same as training)
- **Temperature:** 0.0 (greedy decoding for deterministic outputs)
- **Max new tokens:** 16 (sufficient for "HH:MM" format)
- **Prompt:** Inference prompt (Table S6)
- **Hardware:** 8$\times$ A6000 GPUs with DeepSpeed inference
- **Time:** $\sim$2 hours per model

Table S4. **DPO preference data generation configuration.**

Parameter	Value
SFT Checkpoint Selection	
Qwen2.5-VL-7B	checkpoint-145 (epoch 5)
Llama-3.2-11B	checkpoint-174 (epoch 6)
Gemma3-12B	checkpoint-203 (epoch 7)
SFT Accuracy at Selected Checkpoint	
Qwen2.5-VL-7B	20.3% Full Time Acc
Llama-3.2-11B	45.8% Full Time Acc
Gemma3-12B	34.2% Full Time Acc
Training samples	7,236
Minute tolerance	$\pm$2 minutes

D.2.2. Swap-DPO Transformation: Edge Cases

The main paper describes the SwapHands transformation. Here we document edge cases and implementation details:

1. Times near 12:00:

- Input: 12:00 $\rightarrow$ Output: 12:00 (degenerate case, both hands point up)
- *Handling:* These cases yield degenerate swaps (i.e., $\textsc{SwapHands}(y_{\text{gt}}) = y_{\text{gt}}$) and are filtered by our validation checks (Sec. D.2.3) when swap-based negatives are used.

2. Near-overlapping hands:

- Example: 1:05 ($\theta_h = 32.5°$, $\theta_m = 30°$)
- Swapped: 1:05 (nearly identical)
- Handling: Such cases are filtered out by our distinctness / temporal-distance checks Section. D.2.3

3. Half-hour positions:

- Example: 3:30 $\rightarrow$ Swapped: 6:18
- Hour hand at 105° (halfway between 3 and 4)
- Minute hand at 180° (pointing at 6)
- *Handling:* Works as intended

4. Rounding behavior:

- We use floor division for hours: $h_{\text{new}} = \lfloor \theta_m / 30 \rfloor$
- We use modulo for minutes: $m_{\text{new}} = (\theta_h / 6) \bmod 60$
- This ensures outputs remain in valid ranges: $h \in [0, 11]$, $m \in [0, 59]$

D.2.3. Quality Control and Validation

We perform the following validation checks on generated preference pairs:

1. **Format validation:** Both y_w and y_l must be valid HH:MM strings
2. **Distinctness:** $y_w \neq y_l$ (reject if swapped time equals ground truth)
3. **Geometric plausibility:** For Swap-DPO pairs, verify that swapped time corresponds to a valid clock configuration

4. **Temporal distance:** Ensure $|y_w - y_l| > 5$ minutes to avoid noisy signals from nearly identical times

After validation, we retain 7,187 out of 7,236 samples (99.3%). The 49 rejected samples include parse failures, degenerate swaps, and format errors.

D.2.4. Data Storage and Format

The final preference dataset is stored as a JSON Lines file where each line contains:

- `image_path`: Relative path to training image
- `chosen`: Ground truth time (always y_w)
- `rejected`: Constructed negative sample (y_l)

D.2.5. Comparison with Random-DPO Baseline

To validate our hybrid strategy, we compare against a *Random-DPO* baseline where y_l is sampled uniformly from incorrect times (excluding ground truth). As shown in Table S2, Random-DPO yields a hand-swap gap of +2.60% (averaged across models), compared to +2.28% for our hybrid approach.

Key Finding: The hybrid strategy achieves the best hand-swap gap reduction (+2.28%) by combining:

- **Geometric consistency** from Swap-DPO (teaches hand roles)
- **Error diversity** from SFT mistakes (teaches robustness)

Pure Swap-DPO (using swapped times for all samples, even when SFT is wrong) performs slightly worse (+2.15% gap) because it ignores the model's natural error distribution, missing opportunities to correct systematic mistakes like occlusion handling or numeral misreading. Random-DPO performs worst (+2.60% gap) because randomly sampled times lack geometric consistency and fail to specifically target hand confusion.

E. Prompt Engineering and Design

Effective prompt design is critical for teaching VLMs to read analog clocks accurately. This section describes our comprehensive prompting strategy, including training-time prompt rotation, inference-time simplification.

E.1. Training Prompt Design and Rotation Strategy

During supervised fine-tuning, we employ *prompt rotation*—cycling through three semantically equivalent but lexically diverse prompts to prevent overfitting to specific phrasings. Table S5 presents our training prompts with key phrase variations highlighted.

Prompt Design Rationale:

1. **Explicit hand disambiguation:** All prompts explicitly describe hour hand attributes (*short, thick*) and minute hand attributes (*long, thin*). This addresses the core hand confusion problem by providing unambiguous semantic roles.
2. **Multi-clock handling:** Instructions to select "the most prominent," "primary," or "most visible" clock ensure consistent behavior when multiple clocks appear in a single image (∼8% of training data). Without this, the model randomly attends to different clocks, causing training instability.
3. **Ambiguity resolution rule:** The directive "if a hand is between marks, use the lower hour and nearest minute" provides a deterministic tie-breaking strategy. This reduces annotation ambiguity (annotators might disagree on 3:14 vs. 3:15) and training noise. This rule is consistent with standard clock reading: if the hour hand lies between two hour marks, we report the lower hour.
4. **Structured output format:** Requiring "HH:MM" with 12-hour convention and leading zeros (e.g., "08:05" not "8:5") simplifies parsing and eliminates format-related errors. We use 12-hour format because most analog clocks display 1–12, not 0–23.
5. **Fallback handling:** The "NO CLOCK" instruction prevents hallucination on images without visible analog clocks. This is critical for robustness on diverse web-scraped data where some images may be mislabeled or contain only digital clocks.

Rotation Strategy: During training, we randomly sample one of the three prompts for each example with uniform probability (33.3% each). This serves three purposes:

- **Prevents prompt memorization:** Forces the model to learn underlying task semantics rather than surface lexical patterns
- **Improves robustness:** Generalizes to unseen prompt formulations at test time
- **Reduces overfitting:** Increases effective data diversity without collecting new images

We empirically verified that prompt rotation improves transfer to novel prompt variations compared to training with a single fixed prompt.

E.2. Inference Prompt Design

For evaluation, we use a single, streamlined prompt that retains all critical instructions while using natural language. Table S6 presents our inference prompt.

Simplification Rationale:

- **Omitted instructions:** We remove "ignore digital displays," multi-clock selection details, and "NO CLOCK" fallback because our curated test set contains only valid analog clocks. This reduces prompt length and focuses the model's attention.
- **Explicit format example:** The phrase "e.g., 08:05" provides a concrete example reinforcing the expected output structure (leading zero, colon separator, no AM/PM).
- **Natural phrasing:** We use more conversational language ("The hour hand is...") compared to training prompts'

Table S5. **Training prompt variations for supervised fine-tuning.** We rotate through three semantically equivalent prompts to prevent overfitting while maintaining consistent instruction semantics. Key phrase variations are shown; all prompts share the core instructions for hand identification, ambiguity resolution, and output formatting.

Prompt ID	Full Prompt Text
Prompt A	`Identify the single most prominent analog clock in the image (ignore digital displays). If multiple clocks are visible, choose the largest clearly visible face. Read hour = shorter thicker hand and minute = longer thinner hand; ignore any seconds hand. If a hand lies between marks, use the lower hour and the nearest minute. Output only HH:MM (12-hour, leading zero, no AM/PM). If no analog clock is visible, output NO CLOCK.`
Prompt B	`Find the primary analog clock (exclude digital). Select the largest clearly visible face when multiple are present. Hour = shorter thicker hand; minute = longer thinner hand; ignore seconds. If a hand is between ticks, choose the lower hour and nearest minute. Answer strictly as HH:MM (12-hour, leading zero). If no analog clock is found, answer NO CLOCK.`
Prompt C	`Locate the most visible analog clock and ignore any digital displays. If several clocks exist, pick the biggest clear dial. Use the short thick hand for hours and the long thin hand for minutes; ignore seconds. Between ticks: take the lower hour and nearest minute. Return only HH:MM (12-hour, leading zero). If none is present, return NO CLOCK.`

Common Core Instructions (present in all prompts):
1. Select the most prominent analog clock (ignore digital displays)
2. Hour hand = short/thick; Minute hand = long/thin (ignore seconds)
3. Ambiguity resolution: if between marks, use lower hour and nearest minute
4. Output format: HH:MM (12-hour, leading zero, no AM/PM)
5. Fallback: output "NO CLOCK" if no analog clock visible

Table S6. **Inference prompt for evaluation.** This prompt is used consistently across all test evaluations, ablation studies, and model comparisons. It omits training-specific instructions (multi-clock selection, NO CLOCK fallback) while preserving core task requirements.

Inference Prompt
`Find the most prominent analog clock in the image. The hour hand is the short, thick one, and the minute hand is the long, thin one. If a hand is between marks, choose the lower hour and the nearest minute. Your answer must be only in HH:MM format (e.g., 08:05).`

terse notation ("hour = ..."). This tests whether the model has learned semantic understanding rather than pattern matching.

E.3. Prompt Length Analysis

Table S7 compares token counts across different model tokenizers. Our prompts are designed to be concise yet comprehensive.

Table S7. **Prompt length analysis across model tokenizers.** Token counts vary slightly due to different tokenization schemes (SentencePiece for Llama, tiktoken for Qwen, custom for Gemma). Inference prompt averages 68 tokens.

Prompt Type	Qwen	Llama	Gemma
Training Prompt A	92	91	92
Training Prompt B	75	74	76
Training Prompt C	75	74	75
Inference Prompt	68	66	69

The inference prompt's 66–69 token length strikes a balance between providing sufficient instruction and minimizing computational overhead. Shorter prompts (<30 tokens) lack critical guidance, while longer prompts (>150 tokens) provide diminishing returns while increasing latency.

E.4. Cross-Model Consistency

We use *identical prompts* across all three VLM backbones (Qwen2.5-VL-7B, Llama-3.2-11B, Gemma3-12B) to ensure fair comparison. This design choice isolates the effect of model architecture and training procedure from prompt engineering, ensuring that performance differences reflect genuine model capabilities rather than prompt tuning artifacts. Any model-specific prompt optimization would confound our analysis and reduce reproducibility.

E.5. Impact on DPO Training

During DPO training, we use the same inference prompt for both chosen (y_w) and rejected (y_l) responses. This ensures that preference learning focuses on output quality rather than prompt interpretation differences. The consistent prompt also allows the Swap-DPO mechanism to function correctly—both the correct time and the geometrically swapped time are generated in response to identical instructions, isolating hand role confusion as the sole difference.