

WINNAR749 at SHROOM-Visions: Model-Agnostic Fine-Grained Hallucination Detection in VLMs via QLoRA Approach with Soft, Calibrated Targets

Ranjit Raut

Department of Artificial Intelligence
Kathmandu University
Dhulikhel, Nepal
rautranjit916@gmail.com

Abstract

Vision-Language Models (VLMs) are showing remarkable performance in connecting vision and language, closely intertwined. However, they are vulnerable to hallucination and alignment problems. To address these issues, this paper describes a simple system for the SHROOM-visions shared task on character-level hallucination detection in image-conditioned text. The system attaches two lightweight linear heads to a pretrained VLM backbone and fine-tunes them, together with low-rank adapters on the backbone. As a result, they predict a per-character hallucination probability and a per-character distribution over five hallucination classes. We train directly against soft targets as the released annotations are themselves soft (overlapping, disagreeing multi-annotator spans), rather than collapsing them into hard span labels and fit a post-hoc isotonic calibration map so that predicted probabilities track empirical annotator agreement, the quantity the task’s Confidence Calibration metric rewards. The system we submitted to the official shared-task evaluation built on Qwen2.5-VL-3B-Instruct contained two implementation issues that we identified only after the submission deadline; we report the official results as submitted, document the issues and their fixes, and separately report unofficial post-evaluation results obtained after correcting them and porting the pipeline to Qwen3.5-4B.¹

1 Introduction

Hallucination is one of the major issues for foundational models (Huang et al., 2025). It is an inherent aspect of those models when they are not able to generalize properly and result on producing false output plausibly. Given its natural tendency, it should be mitigate as much as possible in the models to improve alignment and reduce implausible outputs. In multimodal models, it becomes an

immediate topic to address to make models more explainable, interpretable, and responsible (Liu et al., 2024; Bai et al., 2025b). SHROOM-visions² (Vázquez et al., 2026) asks participants to label, for every character of an image-conditioned response, (a) the probability that the character is part of a hallucination, and (b) which of four classes: invention, mischaracterization, OCR problem, or miscounting, plus a catch-all “other”. Systems are ranked on two axes: Span Identification (character-level IoU against the gold mask) and Confidence Calibration (correlation between predicted probability and empirical multi-annotator agreement).

A close look at the released training data Mickus et al. (2026) shows that gold labels are not clean, non-overlapping spans, and annotators can flag overlapping but different-extent, different-class spans over the same characters, each with its own agreement rate. Thus, there is no single hard tag per character, only a graded, sometimes-contradictory annotation. We keep that gradedness rather than discard it, building continuous per-character targets directly from the released prob fields and training against them. Since, collapsing to hard 0/1 labels early would throw away exactly the disagreement signal the Confidence Calibration metric is scored against.

We are transparent that the system submitted to the official evaluation differs from the fully corrected pipeline this paper primarily describes. We discovered two implementation issues in the submitted system only after the deadline. Section 4 documents both the official submission and the corrected, unofficial re-run, and every result table is labeled *official* (frozen, submission-deadline numbers) or *post-evaluation* (obtained after the deadline, outside the shared-task ranking).

¹The code is available at <https://github.com/theranjitraut/SHROOM-Visions-2026>

²<https://helsinki-nlp.github.io/shroom/2026>

2 Related Works

Parameter-efficient fine-tuning. The full fine-tuning of a multi-billion-parameter VLM is impractical on modest hardware and, at the shared task’s per-language data scale, prone to overfitting. Low-Rank Adaptation (LoRA) (Hu et al., 2022) freezes the pretrained weights and injects small trainable low-rank matrices into selected projections, drastically cutting trainable parameters. QLoRA (Dettmers et al., 2023) extends this to a frozen backbone quantized to 4 bits via bitsandbytes (Dettmers et al., 2022), making it possible to adapt a multi-billion-parameter backbone on a single consumer GPU. We build directly on this line of work rather than fine-tuning the backbone in full precision.

Uncertainty and calibration. Guo et al. (2017) shows that forecasting probability estimates that accurately reflect the chance of correctness, i.e., confidence calibration, is crucial for classification models across many applications. MC-dropout (Gal and Ghahramani, 2016), treating dropout at inference time as an approximate Bayesian posterior, gives a cheap way to obtain a predictive distribution rather than a single point estimate. We use it since the Confidence Calibration metric explicitly rewards probabilities that track empirical agreement rather than raw model confidence. Isotonic regression (Zadrozny and Elkan, 2002) is a standard non-parametric post-hoc recalibration technique that we use to map raw sigmoid probabilities onto the empirical agreement scale without assuming a particular parametric miscalibration shape, unlike Platt scaling (Platt, 1999). We revisit the applicability of a monotonic recalibration technique under a rank-based evaluation metric in Section 3.4.

SHROOM and hallucination detection. SHROOM-vision is the vision-and-language extension of the *SHROOM shared task series, which began as a text-only hallucination span/label detection task (Mickus et al., 2024). The current dataset (Mickus et al., 2026) extends this with image-conditioned outputs, multiple languages, and fine-grained hallucination classes. It ships overlapping, disagreeing annotator spans rather than a single adjudicated gold span per hallucination, which motivates the soft-target formulation in Section 3.

3 Method

3.1 Data and Soft Targets

Each training example consists of an image, a prompt, a free-text response, and zero or more label spans {start, end, label, prob}, where start/end are character offsets into the response and prob is the fraction of annotators who flagged that exact span as that class. Responses with an empty label list are clean and non-hallucinated negatives.

Character-level targets. For every character c we compute two soft targets: (i) a scalar $\text{hallu}(c) \in [0, 1]$, obtained by *summing* the prob of every label span covering c and clipping at 1, mirroring how the official scorer accumulates per-span probability mass onto its own reference vector; and (ii) a distribution $\text{class}(c)$ over the five classes, obtained by normalizing the prob mass each class contributes at c . Characters with no covering span get an all-zero class vector and are excluded from the classification loss, since class identity is undefined where nothing was flagged.

The *sum*-based aggregation above is the corrected formulation used in our post-evaluation pipeline in Section 4. The system submitted to the official evaluation instead aggregated overlapping annotations by taking the *maximum* prob across covering spans, which systematically underestimates $\text{hallu}(c)$ whenever spans of different classes overlap at a character: a common occurrence in this dataset (e.g., a narrower OCR span nested inside a wider invention span, each independently supported by a subset of annotators). We ablate this design choice in Section D.

Token alignment. The backbone tokenizer’s subword units do not align with character boundaries, which matters because Span Identification is scored per character and one of the four task languages (Chinese) has no whitespace-delimited word boundaries. We tokenize the response with offset mapping, average the covered character targets onto every response token, and separately keep the character offsets so token-level predictions can be broadcast back onto individual characters at inference exactly as they were pooled going in. The image-plus-prompt prefix is tokenized through the model’s own processor (not the raw tokenizer) so the single image placeholder is expanded into the exact number of vision tokens the processor’s resolution cap will produce; skipping this causes a

shape mismatch between vision embeddings and image-token positions. We also cap image resolution via a minimum/maximum-pixel setting on the processor, since an unbounded input image can otherwise produce well over a thousand vision tokens on its own.

Per-token supervision arrays (hallucination and class targets) must span the *full* input sequence, image-and-instruction prefix plus response, and be index-aligned with the input-token and attention-mask arrays, with prefix positions carrying an ignore sentinel. In the officially submitted system, these target arrays instead covered only the response segment while the input arrays covered the full sequence; batch collation silently padded the shorter target arrays without realigning them to the correct token positions, so after padding, the real (non-ignored) target values ended up superimposed on prefix positions rather than response positions, leaving almost no valid supervision. We describe this issue, its fix, and its measured impact in Section 4.

3.2 Model

We attach two randomly initialized linear heads on top of the last hidden layer of an instruction-tuned VLM backbone: a scalar *hallucination head* and a 5-way *class head*. Our post-evaluation reference configuration uses Qwen3.5-4B³ (Qwen Team, 2026); the system submitted to the official evaluation used Qwen2.5-VL-3B-Instruct⁴ (Bai et al., 2025a). Section 4 details why these differ and what changed in porting the head-attachment mechanics between the two backbones.

The hallucination head is trained with binary cross-entropy against the soft `hallu(c)` target, pooled to the token level, over response tokens only. The class head is trained with soft cross-entropy against `class(c)`, restricted to tokens with any class support. The total loss is the unweighted sum of the two.

3.3 Training

We fine-tune with QLoRA: the backbone loads in 4-bit (NF4, double quantization, bf16 compute) via `bitsandbytes`, is prepared for k-bit training (norms upcast to fp32, gradient checkpointing enabled), and low-rank adapters are attached on top of the attention and MLP projections. Only the LoRA

adapters and the two heads are trainable; the quantized base weights stay frozen throughout, keeping the pipeline runnable on a single consumer GPU with 12 GB VRAM (an NVIDIA RTX 3080 Ti). We train one multilingual model over all four languages rather than four monolingual models, since each language individually has only a few thousand labeled examples; too little for full fine-tuning of a separate model per language using language only as an optional filter for ablations.

Because per-image resolution and hence sequence length vary, we use a per-step batch size of 1 with gradient accumulation for a larger effective batch size, rather than batching variable-resolution images together. Only the response tail is dropped on overlong examples; truncating the shared image/prompt prefix would silently misalign the number of image placeholder tokens with the number of vision encoder patches and crash the forward pass.

3.4 Calibration

Since the released test set is unlabeled, it cannot be used to fit a calibration curve. We instead carve a 10% held-out slice out of each language’s labeled training file, training only on the remainder after a separate 10% validation split. We fit an isotonic regression mapping raw sigmoid probability to empirical annotator-agreement rate on that held-out slice, using several stochastic forward passes per example (MC-dropout) rather than a single deterministic pass, since dropout-averaged probabilities correlate better with empirical agreement than a single overconfident sigmoid. The fitted map is stored as a dense lookup table so inference does not require reloading a fitting library.

Because isotonic regression is a monotonic, non-decreasing transform, it cannot, by construction, change the *rank ordering* of a set of predictions. If the official Confidence Calibration score is a Spearman rank correlation between predicted probability and empirical agreement, isotonic calibration leaves that score unchanged relative to the raw, uncalibrated probabilities, up to tie-breaking effects from the discretized lookup table, so it is fair to ask what calibration buys us. We identify two reasons it still matters here. First, our own per-epoch model-selection proxy in Section 3.6 uses a *Pearson* correlation rather than a rank correlation, since it is cheaper to compute per epoch than a full rank correlation over long sequences; Pearson correlation is not monotonic-invariant, so calibration can and

³<https://huggingface.co/Qwen/Qwen3.5-4B>

⁴<https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct>

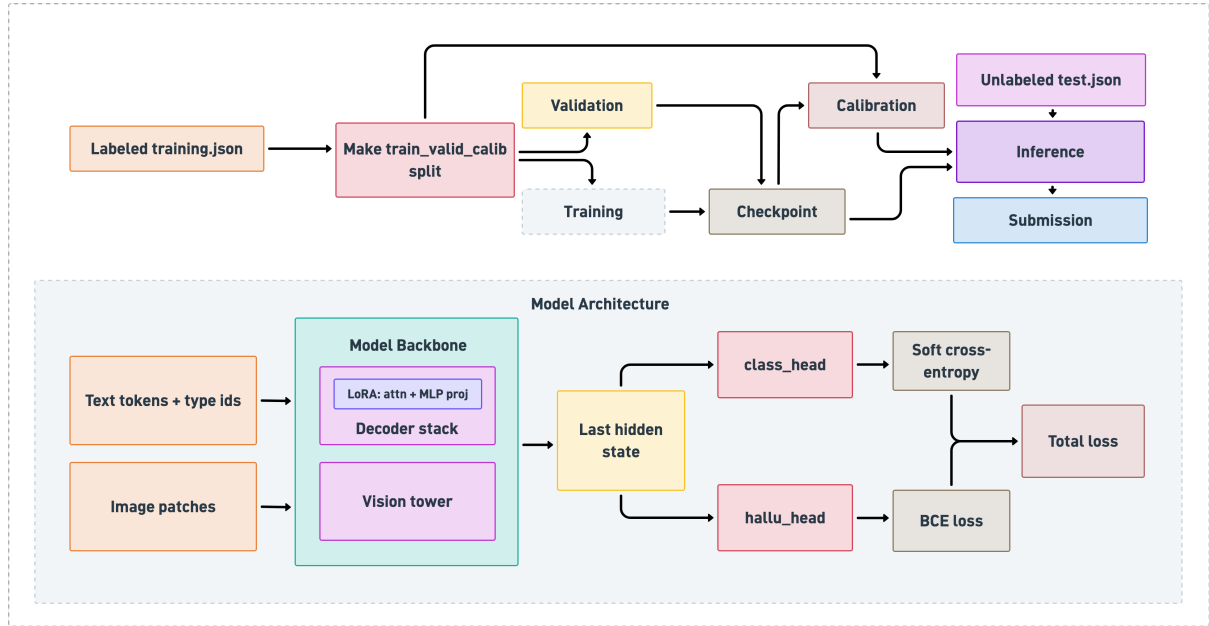


Figure 1: System workflow of the proposed system. Top: labeled data is split into training/validation/calibration subsets; the model is trained and checkpointed, then used both to fit the isotonic map and, with that map, to run inference on the unlabeled test set. Bottom: the backbone-plus-heads architecture used identically across training, calibration, and inference.

does move this proxy in Table 7. Second, and more important for the metric that is actually scored, our span-extraction step in Section 3.5 groups characters into predicted spans using a *fixed absolute threshold* on the calibrated probability. Thresholding is not rank-invariant: a systematically over- or under-confident raw model produces differently shaped spans at a fixed nominal threshold than a properly calibrated one, directly affecting Span Identification (IoU) even though it leaves a rank-based Confidence Calibration score untouched. We therefore justify calibration around span thresholding and our own Pearson-based model-selection proxy, rather than the official rank-based metric itself, and report the corresponding ablation in Section D.

3.5 Inference and Submission

At inference, we rerun the MC-dropout pass, broadcast token probabilities to characters, and apply the calibration map. Contiguous above-threshold characters form a predicted span, which also breaks wherever the predicted class changes, since the scorer builds a piecewise-constant curve per span and merging classes would flatten real variation. Class names are mapped to the kit’s exact required casing (only “OCR” differs) before writing one submission file per language.

3.6 Experiments

We evaluate on a per-epoch 10% held-out validation split (disjoint from both the training and calibration slices) using two token-level proxy metrics: IoU between thresholded predicted and gold hallucination masks, and Pearson correlation between predicted probability and empirical agreement rate, which are cheaper to compute per epoch than the official character-level metrics and are used only for model selection during development. Before reporting or submitting official numbers, we re-run calibration and scoring at the character level, since sub-word tokens do not align one-to-one with the characters the shared task actually scores. Throughout the results tables, Cor denotes character-level correlation between predicted hallucination probability and empirical annotator agreement over all characters in the response; Cor lbl denotes the same correlation restricted to characters that received a non-zero class label from at least one annotator, isolating calibration quality where the class head’s predictions are also defined.

4 Result and Analysis

The system evaluated on the official leaderboard was built on Qwen2.5-VL-3B-Instruct, fine-tuned with QLoRA as described in Section 3.3, using the head architecture of Section 3.2. Table 1 reports

its official rank and score for each language and metric, taken directly from the leaderboard.⁵

During post-submission analysis intended to understand unexpectedly weak validation behavior, we identified two implementation issues in the submitted system’s data pipeline, neither of which raised a runtime error. (1) The submitted system aggregated overlapping annotator spans at each character using the *maximum* reported agreement probability rather than the *sum* in Section 3.1, systematically underestimating hallu(c) wherever annotations of different classes overlap. (2) The submitted system’s per-token supervision arrays covered only the response segment of each example, while the corresponding input/attention-mask arrays covered the full (prefix + response) sequence. Because batch collation pads every tensor to a common length without realigning shorter ones to the correct starting offset, the real (non-ignored) supervision values ended up superimposed on prefix token positions after padding, while the true response positions received only the ignore-sentinel value. The loss and validation masks (requiring simultaneously non-ignored supervision *and* a response-token position) were consequently non-empty almost nowhere, so the model received essentially no effective gradient signal despite training completing without error, and validation/calibration correlations stayed near zero.

Both issues were confirmed empirically: a targeted diagnostic that inspects a single un-collated training example (bypassing the padding step that masked the second issue) reproduced a hard length mismatch between the supervision and input arrays, directly confirming issue (2); the effect of issue (1) was confirmed by comparing aggregate positive/negative target mass under each aggregation rule on the same data. We consider this level of detail important for reproducibility, given that neither issue produced an exception and both were consistent with, but not conclusively diagnosed by, the training-time symptoms alone (near-zero training loss, exactly zero validation loss, undefined IoU, and undefined calibration correlation on every validation batch).

After fixing both issues, we additionally ported the pipeline from Qwen2.5-VL-3B-Instruct to Qwen3.5-4B. The official results are reported in Table 1: frozen, submission-deadline numbers for

Qwen2.5-VL-3B-Instruct with both issues present, exactly as scored by the organizers; post-evaluation, unofficial results are in Table 4: obtained after the deadline, on Qwen3.5-4B, with both issues fixed, scored with the same official scorer script. Post-evaluation numbers are not directly comparable to the official leaderboard, since they reflect a different backbone, a corrected pipeline, and evaluation after the competition’s frozen test window. We report them only to document and quantify the impact of the two issues above and to give a corrected reference implementation. Post-evaluation results are in Section C.

Language	Cor	Cor lbl	IoU	Rank
en	0.1549	0.1549	0.1549	28/28
fr	0.1679	0.1679	0.1679	20/21
it	0.1635	0.1635	0.1635	20/21
zh	0.2554	0.2554	0.2554	21/22

Table 1: Official results, submitted system (Qwen2.5-VL-3B-Instruct, both issues present). Scores are numerically identical to the constant (“predict no hallucination”) baseline in Table 4 on all metrics and languages, consistent with issue (2) suppressing effective supervision.

5 Conclusion

We presented a simple recipe for the SHROOM-visions task. Instead of collapsing annotations into hard spans, we keep them as soft, per-character training targets. We fine-tune a VLM backbone with QLoRA plus two small heads, calibrate the probabilities afterward using a held-out labeled slice, and group calibrated per-character probabilities into spans at inference time. The system we submitted to the official evaluation contained two implementation issues that suppressed most of its effective training signal, and we report its official results transparently in Table 1; after correcting both issues and porting the backbone to Qwen3.5-4B, our unofficial post-evaluation results in Table 4 outperform both the constant baseline and the HaluShift++ baseline across all four languages and both metrics. We view the discrepancy between these two result sets, and its root causes, as a useful case study in why silent (non-crashing) implementation bugs in soft-label, multi-array training pipelines deserve targeted diagnostic tooling beyond monitoring aggregate loss curves alone.

⁵Tied ranks are reported as (position/total distinct scores among tied entries); see the official results for exact tie handling.

Limitations

Our token-level validation metrics in Section 3.6 are a fast stand-in for the official character-level Span Identification and Confidence Calibration metrics, not a replacement for them; we use them only to track training progress and rely on the official scorer for any number that decides which submission to send. We ablate our experiments on a smaller 6,000-example set than the full 15,102-example set given our single-GPU compute. As discussed in Section 3.4, isotonic calibration cannot, by construction, move a rank-based Confidence Calibration score, so its value in our pipeline rests specifically on span-threshold sensitivity and our own Pearson-based model-selection proxy rather than on the official metric directly; an important caveat for any system relying on post-hoc monotonic calibration under rank-based evaluation.

Training one multilingual model instead of one per language means more data per parameter but less specialization; languages that differ a lot from the rest of the mix, like Chinese with its lack of whitespace, might be expected to benefit less from the shared model, but our post-eval results in Table 4 show the opposite: Chinese scores highest of all four languages across every metric. One possible explanation is that the pre-trained model was exposed to a large, high-quality Chinese corpus spanning diverse domains during pre-training; if the tokenizer and subword alignment for Chinese are also comparatively well-optimized, the model may enter fine-tuning with stronger Chinese-specific representations than for the other languages.

Finally, our ablation results in Section D are pooled across all four languages on a reduced, stratified subsample rather than the full labeled set; given Chinese’s distinctive behavior noted above, per-language ablation effects may not be uniform, and some ablated components (sum aggregation, MC-dropout, the joint class head) show a reversed sign relative to our design expectations on this smaller slice. We discuss this further in Section D and leave a full-scale, per-language ablation re-run to future work given our single-GPU compute budget.

Acknowledgements

We thank the SHROOM-vision organizers for curating and releasing the dataset and the accompanying participant kit that this system was built against.

References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuezhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025a. [Qwen2.5-vl technical report](#). *Preprint*, arXiv:2502.13923.
- Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2025b. [Hallucination of multimodal large language models: A survey](#). *Preprint*, arXiv:2404.18930.
- Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*.
- Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). *Preprint*, arXiv:1706.04599.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM transactions on information systems*, 43(2):1–55.
- Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024. [A survey on hallucination in large vision-language models](#). *Preprint*, arXiv:2402.00253.
- Timothee Mickus, Claudio Savelli, Eduardo Calò, Emilio Raimond, Stella Frank, Hengyu Luo, Flavio Giobergia, Vincent Segonne, Chuyuan Li, Aman Sinha, Lorenzo Vaiani, Jörg Tiedemann, and Raúl Vázquez. 2026. [Can humans dream of electric sheep? human-written samples for fine-grained vision-and-language hallucination benchmarking](#). *Preprint*, arXiv:2608.01021.
- Timothee Mickus, Elaine Zosa, Raul Vazquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024.

SemEval-2024 task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1979–1993.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#). Accessed: August 7, 2026.

Raúl Vázquez, Aman Sinha, Chuyuan Li, Artem Shelmanov, Artem Vazhentsev, Claudio Savelli, Eduardo Calò, Emilio Raimond, Stella Frank, Hengyu Luo, Flavio Giobergia, Vincent Segonne, Lorenzo Viani, Jörg Tiedemann, and Timothee Mickus. 2026. [Overview of shroom-visions 2026: A shared task on hallucination detection in large vision-language models](#). *Preprint*, arXiv:2608.25662.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.

A Dataset

SHROOM-visions provides a curated dataset of 20,000 samples with multiple annotations and a fine-grained, span-level labeling scheme.

Dataset Split	Size
Training set	~15,200 samples [†]
Test set	~4,800 samples [†]

Table 2: Dataset splits by size and composition. The training set is composed of outputs from 5 diverse LVLMS, ~3,800 samples per language; the testing set has 1,200 samples per language. [†] Altogether, 15,102 training samples and 4,898 testing samples.

Split	%	No. of sample
Training	80%	12,080
Validation	10%	1,511
Calibration	10%	1,511
Total (labeled)	100%	15,102
Testing (unlabeled)	100%	4,898

Table 3: Training, validation, calibration, and testing split.

The 15,102 labeled examples are split 80/10/10 into training, validation, and calibration sets; the calibration split fits the isotonic-regression map

in Section 3.6. An additional 4,898 unlabeled examples form the test set used for the shared-task submission.

B Evaluation Metrics

Systems are scored at the *character level* against multi-annotator gold labels. For item d of length L_d , a system outputs a probability $Pr(c)$ for each character $c \in \{1, \dots, L_d\}$ that it belongs to a hallucination, giving prediction vector $\hat{r} \in \mathbb{R}^{L_d}$ ($\hat{r}_c = Pr(c)$); the gold vector $r \in \mathbb{R}^{L_d}$ is built the same way from the annotator labels. Three metrics are computed per item and averaged per language.

Unlabeled Correlation (Corr). Spearman correlation between gold and predicted vectors, rewarding confidence that tracks annotator agreement:

$$\text{Corr}(d) = \rho(r, \hat{r}) \quad (1)$$

If either vector is constant (no hallucination present/predicted), ρ is undefined and the item is instead scored on presence-vs-absence agreement.

Labeled Correlation (Corr_{lbl}). Same idea, but computed per label $\ell \in \mathcal{L}(d)$ (labels appearing in gold or prediction) and averaged, to check whether the hallucination *type* is also correct:

$$\text{Corr}_{\text{lbl}}(d) = \frac{1}{|\mathcal{L}(d)|} \sum_{\ell \in \mathcal{L}(d)} \rho(r_\ell, \hat{r}_\ell) \quad (2)$$

Items with an empty gold or predicted set score 1.0.

Intersection-over-Union (IoU). Measures span localization, ignoring confidence and class, by thresholding both vectors into character sets $R(d) = \{c : r_c > 0\}$ and $\hat{R}(d) = \{c : \hat{r}_c > 0\}$:

$$\text{IoU}(d) = \frac{|R(d) \cap \hat{R}(d)|}{|R(d) \cup \hat{R}(d)|} \quad (3)$$

Empty gold or predicted sets also score 1.0.

In short: IoU rewards precise localization, Corr rewards well-calibrated confidence, and Corr_{lbl} rewards correct categorization.

C Post-Evaluation (Unofficial) Results

Porting the pipeline from Qwen2.5-VL-3B-Instruct to Qwen3.5-4B required extending the set of LoRA-adapted projection names: Qwen3.5’s decoder alternates standard attention layers with Gated DeltaNet (linear-attention) layers whose

System	Language	Cor	Cor lbl	IoU
Constant baseline	en	0.1549	0.1549	0.1549
Constant baseline	fr	0.1679	0.1679	0.1679
Constant baseline	it	0.1635	0.1635	0.1635
Constant baseline	zh	0.2554	0.2554	0.2554
HalluShift++ baseline	en	0.2171	0.1967	0.1873
HalluShift++ baseline	fr	0.2154	0.2049	0.1931
HalluShift++ baseline	it	0.2473	0.2154	0.2139
HalluShift++ baseline	zh	0.3235	0.3009	0.2979
Our system (official, Qwen2.5-VL-3B)	en	0.1549	0.1549	0.1549
Our system (official, Qwen2.5-VL-3B)	fr	0.1679	0.1679	0.1679
Our system (official, Qwen2.5-VL-3B)	it	0.1635	0.1635	0.1635
Our system (official, Qwen2.5-VL-3B)	zh	0.2554	0.2554	0.2554
Our system (post-eval, unofficial, Qwen3.5-4B)	en	0.2536	0.3000	0.2380
Our system (post-eval, unofficial, Qwen3.5-4B)	fr	0.2837	0.3328	0.2737
Our system (post-eval, unofficial, Qwen3.5-4B)	it	0.2844	0.3377	0.2761
Our system (post-eval, unofficial, Qwen3.5-4B)	zh	0.3577	0.3976	0.3508

Table 4: Results per system and language. **Official**: frozen leaderboard results for Qwen2.5-VL-3B-Instruct (repeated from Table 1). **Post-eval, unofficial**: results after the deadline on the corrected Qwen3.5-4B pipeline in Section 4, not part of the ranking but scored on the same data.

query/key/value/gate projections use differently named modules, and the original LoRA configuration (targeting only standard attention and MLP projection names) silently adapted just the standard-attention layers, leaving the Gated DeltaNet layers unadapted. We also added the per-token modality identifiers that Qwen3.5’s rotary position encoding requires for multimodal inputs, with no equivalent requirement in Qwen2.5-VL.

Table 5 shows token-level proxy metrics recorded at each training epoch for the corrected pipeline. Training loss falls steadily across all three epochs, but validation loss improves only in epoch 2 before rising again in epoch 3, suggesting overfitting after epoch 2. Despite this, both task metrics keep improving through epoch 3: IoU rises from 0.1048 to 0.1386, and calibration correlation from 0.4263 to 0.4704. Since these are token-level proxies rather than official character-level scores, and token-to-character pooling can blur short spans together, the true character-level numbers may differ from this pattern, particularly for languages where a single token spans many characters.

Comparing Table 1 and the post-eval rows of Table 4 shows that, once both issues are corrected and the backbone upgraded, our system moves from statistically indistinguishable from the constant baseline on every language to outperforming both the constant baseline and HalluShift++ on every language and metric. On Chinese, post-eval IoU improves from the official 0.2554 to 0.3508, and post-eval Cor from 0.2554 to 0.3577. Against

Epoch	Train. loss	Val. loss	IoU	Calib corr
1	1.7328	1.5250	0.1048	0.4263
2	1.3223	1.4329	0.1256	0.4649
3	0.9324	1.6385	0.1386	0.4704

Table 5: Token-level proxy metrics, post-evaluation corrected pipeline (Qwen3.5-4B), computed on the full 15,102-example labeled dataset in Table 3. Not an official shared-task result; reported for model-selection and ablation purposes only.

HalluShift++, the margin is smaller but consistent: English IoU improves from 0.1873 to 0.2380, French from 0.1931 to 0.2737, Italian from 0.2139 to 0.2761, and Chinese from 0.2979 to 0.3508. This comparison documents the magnitude of the two implementation issues’ impact, not a shared-task ranking improvement, since the post-eval numbers were never submitted for official ranking.

Notably, Chinese achieves the strongest post-eval scores of any language across all three systems, running counter to our expectation that its lack of whitespace segmentation would make it the hardest language for token-to-character broadcasting.

We again emphasize that post-evaluation numbers are not directly comparable to the official leaderboard, since they reflect a different backbone, a corrected pipeline, and evaluation after the competition’s frozen test window; we report them only to document and quantify the impact of the two issues above and to give a corrected reference implementation.

D Ablations

We ablate five principal modeling choices: soft (vs. hard) targets, the overlapping-annotation aggregation rule, MC-dropout at inference, the joint class head, and isotonic calibration. The first four are reported together in Table 6; isotonic calibration is reported separately in Table 7, since it is evaluated with a different metric (character-level Pearson correlation on the held-out calibration slice, Section 3.4) rather than the token-level proxies used for the other four. All ablations use the corrected, post-evaluation pipeline (Qwen3.5-4B) and are evaluated with the token-level proxy metrics of Section 3.6 on 1500 randomly sampled examples per language, stratified, seed 42, with 10% held out for validation, 5% for calibration, the rest for training, and trained for a single epoch. This is smaller than the full 15,102-example set used for Table 5 and the main results in Table 4.

Configuration	IoU	Calib. corr
Full system	0.2768	0.2897
– soft targets (hard 0/1 labels)	0.2733	0.2837
– sum agg. (max agg. instead)	0.3003	0.3259
– MC dropout (single pass, $M=1$)	0.2982	0.3224
– class head (binary head only)	0.2838	0.3038

Table 6: Ablation of four principal modeling choices (isotonic calibration reported separately), evaluated on a 1500-example-per-language stratified sample. Removing sum aggregation, MC-dropout, or the class head numerically improves both metrics, but the reduced sample size and per-language variability suggest larger-scale, per-language ablation is needed before drawing firm conclusions.

Table 7 reports the ablation for character-level Pearson correlation between predicted probability and empirical annotator agreement, before and after isotonic calibration, on the held-out calibration slice of the full system.

Configuration	Pearson corr. (char-level)
Raw sigmoid probability	0.4614
Isotonic-calibrated probability	0.4992

Table 7: Effect of isotonic calibration on character-level Pearson correlation with empirical agreement (held-out slice, Qwen3.5-4B pipeline). While monotonic, it improves Pearson correlation (our proxy for model selection) and affects span-extraction thresholds, unlike rank-based scores.

D.1 Hyperparameter Sensitivity

We include a more systematic hyperparameter analysis, motivated in part by the differing evolution of IoU and calibration correlation across training in Table 5. We sweep: (i) the positive-class weight capping the BCE loss’s correction for the rarity of hallucinated characters, (ii) LoRA rank, and (iii) learning rate in Table 8, each evaluated on the token-level proxy metrics of Section 3.6 on the same reduced ablation sample described in Section D, to keep the sweep affordable within our compute budget. We expect the positive-class weight to be the most consequential of the three, since with it disabled entirely the model can trivially minimize loss by predicting the (overwhelmingly more common) negative class almost everywhere, a failure mode related to, but distinct from, the two implementation issues of Section 4: those issues left the model with no effective gradient signal at all, whereas an under-weighted positive class leaves the model with signal that is real but biased toward under-predicting hallucination. Each sweep is restricted to two settings given our single-GPU compute budget; we did not have resources to also include the exact hyperparameter values used for the main results in Table 5 within these reduced-sample sweeps, which we flag as a gap for future, larger-scale sweeps.

Hyperparameter	IoU	Calib. corr
Positive-class weight cap		
1 (unweighted)	0.2918	0.3110
10.0	0.3021	0.3323
LoRA rank		
8	0.2994	0.3240
16	0.2864	0.2981
Learning rate		
5×10^{-5}	0.2885	0.3108
0.0001	0.2979	0.3123

Table 8: Sensitivity sweeps over hyperparameters.