A Training-Free Single-Pass Vision–Language Pipeline for Sentence-Level Visual Chinese Grammatical Error Correction

Lexuan Zhang
KTH Royal Institute of Technology

Abstract

Handwritten text by language learners couples recognition errors with grammatical errors, which makes automatic correction of such text harder than either subproblem alone. Visual Chinese grammatical error correction asks a system to read a handwritten sentence image and to emit both a fully corrected sentence and a list of localized visual edits. We describe a training-free pipeline that makes one call to a vision–language model (VLM) per image and jointly requests a verbatim transcription, a minimal-edit correction, and a typed bounding-box edit list in a single JSON object; a deterministic packer enforces schema and coverage, and a second engine fills only the test images for which the first produces no valid record. A paired 19-image development comparison under a proxy of the official $F_{0.5}$ metric finds that two VLMs obtain the same edit true-positive count while one emits nearly twice as many edits; the more conservative engine has higher edit precision on this slice. The system obtains an official score of 0.5905 on the hidden sentence-level test set of the VisCGEC-v2.0 shared task. The result suggests that a single-pass training-free VLM pipeline is viable for this task, and that edit precision is a useful engine-selection signal on the measured development slice.

1 Introduction

Grammatical error correction (GEC) is usually defined over already-tokenized text (Ng et al., 2014), and Chinese GEC has an established shared-task lineage over learner text (Zhao et al., 2018; Rao et al., 2020; Chang et al., 2023). Visual Chinese GEC (Wang et al., 2025) additionally requires the system to read a real handwritten image produced by a learner of Chinese as a foreign language, so recognition errors, malformed characters, and grammatical errors are entangled. The VisCGEC-v2.0 shared task of Multi-LLL 2026 (Multi-LLL Organizers, 2026) makes that setting operational: the official input is the image; the public test file contains only an identifier and a path; and each prediction must contain a corrected string `predict_text` together with a list of localized edits.

Each edit is a pixel box, a corrected span, and one of four operation types (`Insert`, `Delete`, `Substitute`, `WordOrder`). The official score is the mean of a per-sample combination of text-level $F_{0.5}$ and edit-level $F_{0.5}$, so a fluent rewrite that does not match the annotators' boxes is not enough.

This paper describes the sentence-level system we submitted, a real system that obtained an official score of 0.5905 on the hidden test set (Section 5). Our contributions are:

- a training-free, single-pass pipeline in which one VLM call per image jointly produces a transcription, a correction, and typed bounding-box edits, backed by a deterministic packer that hard-fails on any schema or coverage violation (Section 3);
- an offline, self-validated proxy of the official metric with explicit fidelity limits, used for engine selection on the 245-image development split (Section 5);
- a paired development observation: two engines tie on edit true positives while one emits nearly twice as many edits, making edit precision a useful engine-selection signal on this slice (Section 6).

We did not train or fine-tune any model. We release the prompt template, the offline scorer and submission packer, and both submitted prediction files in an anonymized repository.[1]

[1] `https://anonymous.4open.science/r/viscgec-sentence-pipeline-F466`

Proceedings of the 1st Workshop on Multi-modality and Multi-lingual Language Learning, pages 77–83
November 9, 2026 ©2026 Association for Computational Linguistics

2 Data

Released splits. Sentence-level VisCGEC-v2.0 provides 1,960 training images, 245 development images, and 246 hidden-test images. Training and development records contain `img_id`, `img_path`, a gold corrected string `target_text`, and a gold `edit_list`. They do *not* contain the learner's original string. The test file contains only `img_id` and `img_path`. Systems must therefore recognize text from the image; they cannot assume an OCR transcript.

Annotation shape. On the 1,960 training sentences the gold `edit_list` has mean length 1.80 (median 1, max 14); six sentences have an empty list. Edit types occur in the ratio Substitute 1,693 : Insert 1,079 : Delete 761. `WordOrder` occurs *zero* times in the released training annotations, and also zero times in the 245 development annotations (Substitute 212, Insert 134, Delete 89; mean 1.78 edits). Mean gold sentence length is 24.6 characters (development 25.5). Images are single-line handwriting strips (training width p50 1,494 px, height p50 85 px; test width p50 1,531 px, height p50 93 px).

Why we do not fine-tune. A text GEC trainer wants pairs $(\textit{source}, \textit{target})$. The public files give *target* and a list of *corrected* spans with boxes, but not the original spans, so *source* cannot be reconstructed from the JSON. Obtaining N training pairs would require first OCRing N training images and then trusting that OCR. We did not pay that cost, and we used the training split only (a) to compute the edit-type and edit-count statistics above and (b) as documentation for an optional convention-aware prompt variant. The development split is the only split with gold that we scored.

Scope. The shared task also defines a paragraph-level track over the same schema; we did not submit to it, and the rest of this paper is sentence-level only.

External resources. The only task data we used are the official public packs. Inference uses closed VLMs accessed through an API (Section 4); we did not add an external GEC corpus.

3 Methodology

Figure 1 shows the primary decoding path and the fallback condition. The pipeline operates as follows:

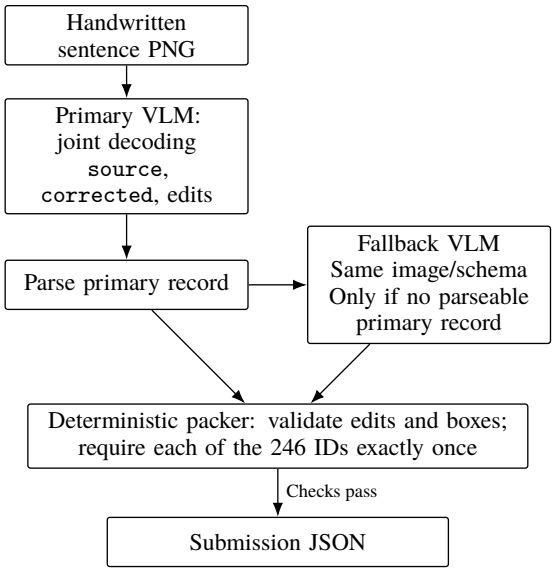


Figure 1: The primary record is retained when parseable. The fallback fills only missing records. The packer refuses to emit a submission when coverage or schema checks fail.

1. Pass one handwritten PNG to a VLM as visual input at original resolution. Invocation goes through a CLI harness: the prompt names the image file and its pixel dimensions, and the harness reads the file and attaches it to the API request.

2. Ask a VLM for a single JSON object with three fields: `source` (verbatim transcription, errors kept), `corrected` (minimal-edit fluent Chinese), and `edit_list` (the visual difference between the two).

3. Parse the last JSON object in the response that contains `source` and `corrected`. Drop edits whose box is non-integer or degenerate, whose type is outside the four-way inventory, or whose `correct_text` is not a string. Deletion text must be the empty string.

4. Set `predict_text` = `corrected`. Round boxes to integers and clip them to the image. Require exact coverage of the 246 test IDs (no missing, extra, or duplicate IDs).

5. If an image has no parseable record from the first engine, fill it from a second engine; a record that already parsed is never overwritten.

6. Write the submission file under the official sample filename

(`sample_submission.json`), packaged as the platform requires.

Prompt. The production template used for both submissions is a short instruction that (i) forbids reading any annotation file, (ii) states the four edit types, (iii) requires Insert boxes to cover a full neighbouring character cell rather than a zero-width line, and (iv) demands one JSON line and nothing else. We also iterated on a longer convention-aware variant of the same template, which tells the model that the reference makes 1.8 edits per sentence, that `WordOrder` is unused in training, and that style improvements and extra connectives are not rewarded, and which quotes eight gold corrected sentences. All 246 archived test prompts instantiate the short template, so we make no claims for the variant. The schema—joint transcription, correction, and boxes—is the same in both templates and is what the packer consumes.

Why joint decoding. Text-level $F_{0.5}$ in the Chinese setting is a comparison of character edits *source* → *gold* against *source* → *prediction*, in the style of ChERRANT (Zhang et al., 2022), itself an adaptation of ERRANT (Bryant et al., 2017). If a system copied its own OCR into `predict_text` and emitted an empty edit list, the text-level term would be zero whenever the OCR differed from the hidden source, and the edit-level term would be zero by construction; the sample score would then be at most 0.5 in the empty-edit case and typically near 0. The same VLM call therefore has to both read and correct, and it has to emit boxes that line up with the annotators.

Deterministic shell. All coverage and schema checks run in a packer that refuses to write the submission file if any ID is missing or illegal. That shell is the only component we treat as bit-stable across engines.

4 Model

We do not train weights; all correction capability comes from commercial vision–language models accessed through an API. A submission in this task is a single file with one record per test image, and the platform allows each team only a small fixed number of scored submissions. We submitted twice: a *contrastive* configuration built first under time pressure, and a *primary* configuration built after development measurement; the platform keeps each team's best-scoring submission. Three engines appear across the two submissions, identified by the router names our access channel recorded at inference time (these are channel identifiers, not necessarily the vendors' public model names; where a channel exposed two aliases, we give both):

- **Qwen** (`qwen3.8-max`), the main engine of the contrastive submission;
- **GPT** (`gpt-5.6-sol / gpt:sol-medium`), the fallback engine in both submissions;
- **Claude** (`claude-opus-5 / claude:high`), the main engine of the primary submission.

A fourth engine (GLM 5.3) was probed on development images and rejected: an 18-image preview scored below the contrastive configuration, and that engine sometimes returned a tool trace instead of the required JSON.

Primary submission. 224 of the 246 records come from Claude. The remaining 22 records are the images for which Claude's output was missing (14), empty (4), or unparseable (4), and are filled by GPT. This fallback is a reliability backfill, not a learned router: a record that already parsed is never overwritten.

Contrastive submission. The earlier submission ran Qwen as the main engine over all 246 test images and applied the same fallback rule, yielding 119 Qwen records and 127 GPT records. Section 5 compares the two configurations on development measurements.

Implementation details that we can state. Each call passes one image (attached by the harness as visual input) and one prompt file; at least one engine's logs show an explicit image-analysis step, so we do not claim to know how each engine consumes the pixels internally. Batch inference used a per-call timeout of 900 seconds and concurrency between 4 and 8. We did not record sampling parameters or exact token limits, and we do not invent them. There is no reranking, no second-pass box refiner, and no lexical GEC model on top of OCR.

5 Evaluation

Official metric. The task defines

$$F_{0.5} = \frac{1.25\,PR}{0.25\,P + R},$$

$$\textit{sample} = 0.5\,\textit{text-}F_{0.5} + 0.5\,\textit{edit-}F_{0.5},$$

and takes the mean over samples. An edit is a true positive only if its box has IoU ≥ 0.5 with a gold box, its `correct_text` matches exactly, and its `edit_type` matches. The task description names ChERRANT as the text-level metric; its implementation, tokenization, and empty-gold convention were not released.

Local proxy. We reimplemented the published formulae as a standalone offline scorer, used only at development time for engine selection and mix control. Text-level $F_{0.5}$ compares character-level Levenshtein edits rather than ChERRANT, so we treat the number as a trend rather than a leaderboard forecast. The scorer is checked against hand-computed cases and negative controls: gold-as-prediction scores 1.0, an empty edit list caps the sample score at 0.5, and shifting every box by 30 px degrades edit $F_{0.5}$ to 0.25. Because we do not have the hidden source string, we compute two text-level branches on development images: an *optimistic* branch that uses the model's own `source` as the ChERRANT source, and a *pessimistic* branch that adds extra false positives for estimated OCR errors, $\max(0,\ d(\texttt{source}, \texttt{target}) - g)$, where d is the character edit distance between the model's transcription and the gold corrected text and g is the number of characters the gold edit list changes. Edit matching is greedy one-to-one, sorted by IoU; empty gold and empty prediction score 1.0. TIER2 is the mean sample score using both terms; TIER1 zeroes the edit term and is not used for selection.

Official result. The primary submission obtains an official overall score of 0.5905 on the hidden test set under the metric above. The platform reports only each team's best-scoring submission, so the contrastive submission has no separately visible official score, and itemized text-level and edit-level sub-scores are not exposed for any team; the remainder of this section therefore analyses the system on the development split. The local proxy numbers below are not directly comparable with the official score: the proxy uses character-level Levenshtein edits instead of ChERRANT, a different split, and a source-side approximation.

Development engine faces. Table 1 reports TIER2 on a 50-image development slice (Claude n=49; one item timed out twice at 600 s). All development figures are single runs. Claude is strongest; GPT is weakest; Qwen sits in between. A 19-image paired slice that holds the image set fixed (Table 2) shows a precision difference: both Claude and Qwen obtain 6 edit true positives against 29 gold edits, but Qwen emits 58 predicted edits against Claude's 31. The TIER2 gap is almost entirely edit-level precision.

Engine	n	TIER2 opt.	TIER2 pess.
Qwen	50	0.230	0.204
GPT	50	0.171	0.160
Claude	49	0.268	0.242

Table 1: Development-set TIER2 by engine. Optimistic/pessimistic refer to the source proxy for text-level $F_{0.5}$. Not hidden-test scores.

	TIER2 pess.	edit $F_{0.5}$	pred.	TP
Claude	0.213	0.232	31	6
Qwen	0.159	0.108	58	6

Table 2: Paired 19-image development slice (same gold, 29 gold edits). Equal edit true-positive counts accompany different predicted edit counts and precision. Unmatched edits may reflect correction, box, or type mismatches.

Mix-weighted control line. Applying the contrastive submission's engine weights (119 Qwen +127 GPT) to the 50-image faces gives a TIER2 of 0.199/0.181. The primary submission's weights (224 Claude +22 GPT) give 0.259/0.235 on the same faces (Table 3). All six recorded axes move up, including edit $F_{0.5}$ ($0.145 \rightarrow 0.184$) and predicted character accuracy ($0.773 \rightarrow 0.869$). A pure-GPT replacement (0.171/0.160) is worse than the contrastive mix and was not used. This arithmetic is a development-face linear mix; it is *not* a score of the 246 test predictions (the test gold is hidden), nor a measured evaluation of either complete fallback pipeline on the same development images. Because fallback is used on specific failures of the primary engine, its average development score need not describe those selected images.

6 Analysis

Edit precision as an engine-selection signal. The paired comparison supports selecting an engine by edit precision on the measured development slice. It does not establish over-editing as the dominant cause of failure: an unmatched edit can also reflect a bounding-box or edit-type mismatch. On a 20-image development sample, character disagreement between the model's transcrip-

	Contrastive	Primary	Δ
TIER2 opt.	0.199	0.259	+0.060
TIER2 pess.	0.181	0.235	+0.053
text $F_{0.5}$ opt.	0.254	0.334	+0.080
text $F_{0.5}$ pess.	0.218	0.285	+0.067
edit $F_{0.5}$	0.145	0.184	+0.040
pred. char acc.	0.773	0.869	+0.096

Table 3: Mix-weighted development faces. Contrastive $= (119/246)$ Qwen $+(127/246)$ GPT; Primary $= (224/246)$ Claude $+(22/246)$ GPT. Development estimates, not hidden-test scores.

tion and the gold corrected text, net of the gold edits themselves, is around 3%. This is a source-side proxy, not a manually measured transcription error rate, and it cannot rule out recognition errors as a major factor. One recorded example illustrates disagreement with the reference: on development item `1974_1`, the gold correction inserts the connective "但是随着" at the start of the strip, whereas the model inserts a sentence-final "了". This example motivates checking unnecessary edits, but it does not quantify their prevalence. Manually checked transcriptions and an analysis of unmatched edits would be needed to separate recognition, correction, box, and type errors.

Optimization trajectory. The two submissions are one measured iteration, not two systems. The contrastive submission was assembled first, under the platform's fixed submission quota. The paired development measurement favoured the engine with higher edit precision—engines tied on true positives yet differed nearly two-fold in emitted edits—so the primary submission re-ran the test set with the more conservative engine and refilled only its 22 missing records. The corresponding mix-weighted estimates are higher on every recorded development axis (Table 3); they do not establish the gain of the actual fallback pipeline. Two cheaper levers were probed and not used in the final calls: a convention-aware prompt variant that encodes training-split statistics (Section 3), and a fourth engine dropped after an 18-image preview scored below the contrastive configuration (Section 4). Outside the sentence track, a pure-geometry feasibility check—in the released paragraph-level training annotations, all 2,237 gold Insert boxes are identically 4 px wide, against which full-cell boxes cannot reach IoU ≥ 0.5—concentrated the entire submission budget on the sentence track.

Submitted edit inventories. On the 246 test predictions themselves (no gold) the contrastive submission contains 584 edits (mean 2.37 per sentence) and the primary submission 435 (mean 1.77). The training mean is 1.80. Type counts: contrastive Substitute 234 / Insert 155 / Delete 114 / `WordOrder` 81; primary Substitute 208 / Insert 111 / Delete 89 / `WordOrder` 27. `WordOrder` is unused in the released training and development gold, although it is permitted by the schema. The hidden-test gold is unavailable, so these counts alone do not establish that every such test prediction is incorrect. Reducing that count from 81 to 27 is a side-effect of preferring Claude, not a dedicated type filter—we did not post-hoc drop `WordOrder`. Five test items in each submission have an empty edit list (two IDs overlap).

Coverage mix versus fusion. The 22 GPT records in the primary submission exist only because a valid submission must cover all 246 test images. They are a reliability backfill, not an ensemble vote. GPT's average development TIER2 is the weakest of the three engines, but that does not determine its performance on the images where Claude produces no parseable record. The recorded linear mix is above the contrastive line on every local axis; the effect of fallback requires measuring complete pipelines on the same development images.

Boxes. Insert boxes on the sentence-level gold are real character cells (width p50 52 px), which is what the prompt asks for. We did not add a geometric refiner: illegal and degenerate boxes are dropped, and the primary submission dropped none.

What we did not try. We did not OCR the 1,960 training images to build GEC pairs; we did not fine-tune; we did not run a two-stage OCR-then-GEC pipeline as a submitted system; we did not enter the paragraph-level track.

7 Conclusion

We have described a training-free, single-pass vision–language pipeline for sentence-level visual Chinese grammatical error correction that obtains an official score of 0.5905 on the hidden test set. The design is minimal by choice: one model call per image, a rigid output contract, and a deterministic packer. On the paired 19-image development slice, the engine that emits fewer edits has the same

edit true-positive count and higher edit precision. This makes edit precision a useful selection signal for that comparison; it does not isolate recognition errors or establish the benefit of the complete fallback pipeline.

Limitations

The local scorer is not ChERRANT and cannot be claimed to match the official script. Development faces have n=50 (Claude n=49), and the mix table linearly combines those faces; it does not evaluate the 246 test predictions against hidden gold. We do not have official itemized scores. Decoding hyperparameters were not logged. Closed VLMs are opaque: we cannot report architecture, context length actually used, or pretraining data. The system is sentence-level only. API engines are a moving target; the identifiers above are those recorded at submission time.

Declaration on Generative AI

The system described in this paper *is* a generative AI system: each test image is corrected by a commercial vision–language model (Section 4) under a fixed prompt and a deterministic packer. No generative model was used to invent evaluation numbers; every figure in Section 5 is copied from run logs or recomputed from the submitted JSON. This manuscript was drafted with assistance from a coding agent that was instructed to quote only logged facts and to mark gaps rather than fill them. The authors take responsibility for the claims that remain.

References

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Hongxiang Chang, Yang Liu, Meng Xu, Yingying Wang, Cunliang Kong, Liner Yang, Yang Erhong, Maosong Sun, Gaoqi Rao, Renfen Hu, and Zhenghao Liu. 2023. CCL23-Eval 任务 7 总结报告: 汉语学习者文本纠错 (Overview of CCL23-Eval Task: Chinese Learner Text Correction). In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pages 239–249, Harbin, China. Chinese Information Processing Society of China.

Multi-LLL Organizers. 2026. The workshop on multimodality and multi-lingual language learning (Multi-LLL 2026). https://xiaoaugenstern.github.io/Multi-LLL-2026/. Shared task: VisCGEC-v2.0. Accessed 13 September 2026.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35, Suzhou, China. Association for Computational Linguistics.

Xiaoman Wang, Dan Yuan, Xin Liu, Yike Zhao, Xiaoxiao Zhang, Xizhi Chen, and Yunshi Lan. 2025. VisCGEC: Benchmarking the visual Chinese grammatical error correction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5054–5068, Albuquerque, New Mexico. Association for Computational Linguistics.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3128–3143, Seattle, United States. Association for Computational Linguistics.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the NLPCC 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing*, pages 439–445. Springer.

A Production prompt and output contract

The production prompt instantiates the following template once per image, with `<PATH>`, `<W>`, and `<H>` filled by the image path and its pixel dimensions:

> Read the image file at `<PATH>` with your Read tool. The image is exactly `<W>` pixels wide and `<H>` pixels tall. It is a cropped strip of handwritten Chinese, one sentence, written by a Chinese-as-a-second-language learner. The sentence contains grammatical errors (wrong character, missing character, redundant character, wrong word order). Produce three things:

(1) `source`: verbatim transcription of what the student wrote, keeping every error exactly as written; (2) `corrected`: the same sentence with all grammatical errors fixed into fluent standard Chinese, changing as little as possible; (3) `edit_list`: the localized corrections, each with `bbox` (integer pixel coordinates `start_x`, `start_y`, `end_x`, `end_y` in this image's original coordinate system, covering the character cell(s) involved; for an `Insert` the box covers a full character-cell-sized region at the insertion position, not a zero-width line), `correct_text` (the corrected text for that spot; the empty string for `Delete`), and `edit_type` (one of `Insert`, `Delete`, `Substitute`, `WordOrder`). Hard constraints: do not search the filesystem for any answer key, annotation file, dataset JSON, or ground truth—the only permitted input is the image itself; coordinates must be measured off this image; output exactly one line of JSON and nothing else.

The file-reading phrasing reflects the invocation harness described in Section 4: it reads the named file and attaches it to the API call as visual input.

The packer accepts the last JSON object in the response that contains `source` and `corrected`; drops edits with non-integer or degenerate boxes, out-of-inventory types, or non-string `correct_text`; requires an empty `correct_text` for `Delete`; rounds and clips boxes to the image; and refuses to emit the submission file unless the 246 test IDs are covered exactly once.