FROM AI FORECAST TO TEXT REPORT: PANGU-DRIVEN WEATHER REPORT GENERATION VIA VISION-LANGUAGE FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work, WeatherSyn showed MLLMs can generate them from ERA5 reanalysis images, but ERA5 has multi-day latency. More fundamentally, it is produced via 4D-Var assimilation over a 12-hour window thus using it as model input for real-time forecasting can leak future information and inflate apparent skill. Another work, WeatherQA, by contrast, targets severe-weather reasoning and text-only discussion generation, not forecast heatmaps, thereby omitting critical spatial information, limiting operational utility. In this paper, we ask whether AI-based weather forecast fields, specifically from Pangu-Weather, can replace ERA5 as the visual input for driving text report generation. We construct an end-to-end pipeline that renders city-level meteorological heatmaps for 31 U.S. cities, pairs them with professional NWS forecast discussions from the WSInstruct corpus, and LoRA fine-tunes Qwen2-VL-7B on 2,016 paired samples from 2020–2021. Our central contribution is a controlled input-source ablation: we train two models under identical hyperparameters, differing only in whether the visual input is rendered from Pangu forecast fields or ERA5 reanalysis fields, and evaluate both on both test sets. On 1,043 test samples from 2022, the input-source cost of replacing ERA5 with Pangu is modest—at most 0.0032 ROUGE-L and 0.0024 BLEU-1—and training on Pangu forecast fields does not hurt downstream report generation. These results indicate that end-to-end, real-time AI weather forecasting pipelines, from initial conditions to human-readable reports, are feasible without relying on delayed reanalysis products. This is the first systematic study of whether AI forecast fields can replace reanalysis in driving text weather report generation.

1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Weather forecasting plays a critical role in modern society, supporting decision-making in agriculture, transportation, energy management, and public safety. While numerical weather prediction (NWP) models have achieved remarkable accuracy, the final step of communicating forecasts to end users—writing human-readable weather reports—remains largely manual. Meteorologists must interpret complex multi-variable forecast fields, integrate information across data sources, and compose coherent textual summaries. This process is time-consuming, subject to human bias, and increasingly strained by the growing volume of available data.

In this paper, we ask: can AI-based weather forecast fields, which are available in near-real-time, replace ERA5 reanalysis as the visual input for driving text report generation? To answer this, we choose Pangu-Weather Bi et al. (2023), a state-of-the-art AI forecasting model that produces global medium-range predictions within seconds on a single GPU. Pangu's outputs are structurally similar to ERA5 fields but contain the inherent prediction errors of any forecast model. Whether these AI-generated forecast fields retain sufficient information for an MLLM to generate accurate textual reports is an open question with significant practical implications.

Answering this question rigorously requires more than a direct comparison against published WeatherSyn numbers, which would conflate several confounding factors—model capacity (7B vs. 8B

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

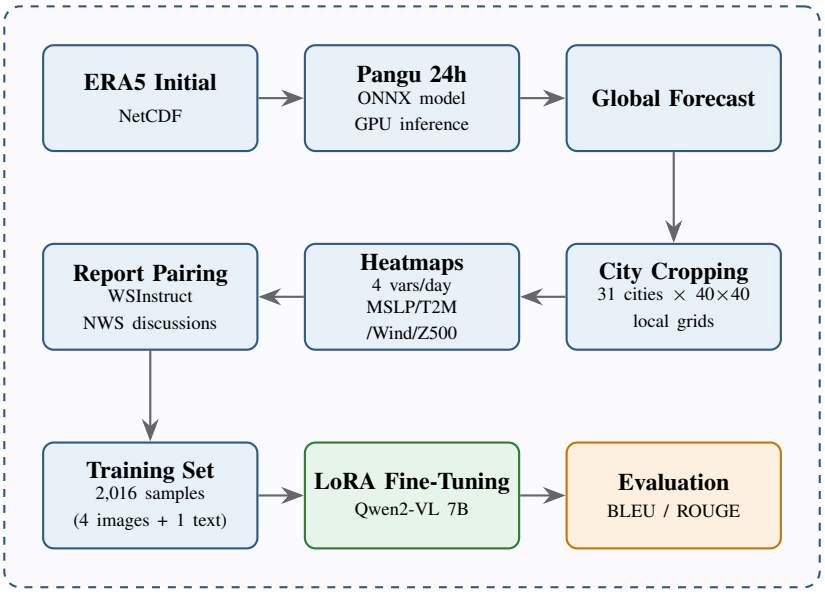


Figure 1: End-to-end pipeline

parameters), training data scale (2,000 vs. 6,000+ samples), and tuning strategy (LoRA vs. full fine-tuning). Instead, we design a *controlled input-source ablation* that isolates the specific impact of Pangu's forecast errors: we fine-tune the same Qwen2-VL-7B model, on the same 2,016 training samples, under identical hyperparameters, varying only the input source (Pangu forecast fields vs. ERA5 reanalysis fields). The performance gap between the two directly quantifies the cost of relying on AI-generated forecasts for downstream text generation.

To support this study, we construct a large-scale dataset spanning three years (2020–2022) and 31 U.S. cities. The training set consists of 2,016 paired samples from 2020–2021, and the test set contains 1,043 samples from 2022. Each sample pairs four meteorological heatmaps (sea level pressure, 2 m temperature, 10 m wind speed, and 500 hPa geopotential height) with a professional NWS Area Forecast Discussion. Importantly, our comparative experiments between Pangu forecasts and ERA5 data are designed to control for the future-information issue in ERA5: any shared future information affects both sides of the comparison, allowing us to isolate Pangu's forecast contribution rather than ERA5's information advantage.

1.2 Contributions

Our contributions are as follows:

1. **End-to-end pipeline.** We construct a complete pipeline that generates Pangu 24-hour global forecast fields from ERA5 initial conditions, renders city-level meteorological heatmaps for 31 U.S. cities (see Table 3), pairs them with professional NWS forecast discussions, and fine-tunes Qwen2-VL-7B via LoRA.
2. **Controlled input-source ablation.** We isolate the impact of AI forecast errors by training two models under identical conditions, differing only in input source (Pangu vs. ERA5). This provides the first direct measurement of how much text report quality degrades when driven by AI-generated forecasts rather than reanalysis data.
3. **Large-scale multi-year evaluation.** Our dataset spans three years and 31 cities. Beyond BLEU-1 and ROUGE-L, we evaluate cross-source generalization (Pangu-trained model on ERA5 inputs and vice versa).
4. **Practical implications.** Our results indicate that end-to-end, real-time AI weather forecasting pipelines—from initial conditions to human-readable reports—are feasible without relying on delayed reanalysis products.

2 Related Work

Numerical weather prediction (NWP) remains the backbone of operational forecasting. Recently, AI models such as Pangu-Weather Bi et al. (2023) produce global medium-range forecasts within seconds on a single GPU, achieving accuracy competitive with NWP. TC-Next Wang et al. (2026) combines GraphCast fields with satellite imagery and transfers zero-shot to Pangu-Weather, IFS HRES, and WeatherNext Cyclones, showing that AI forecast fields contain transferable information. Weather Emulators Decancq et al. (2026) evaluates six deep-learning emulators and finds that while several rival physics-based temperature skill, they suffer from reduced spectral fidelity and under-represent extreme heat peaks. Our work does not improve AI forecasting models; instead, we treat their forecast fields as visual inputs for downstream natural-language weather report generation.

WeatherSyn Zheng et al. (2026) formalizes the Weather Forecasting Report (WFR) task, constructs WSInstruct covering 31 U.S. cities and 8 weather aspects using ERA5 reanalysis data, and fine-tunes Qwen3-VL-8B with an SFT→RFT→DPO pipeline. It outperforms leading closed-source MLLMs on multiple metrics and demonstrates zero-shot generalization to unseen cities. WeatherSyn establishes that Qwen-series MLLMs are a viable backbone for weather report generation from ERA5 reanalysis images.Reanalysis products such as ERA5 provide high-quality atmospheric states, but they suffer from multi-day latency, and their 4D-Var assimilation window can incorporate future observations, so an analysis at time T may contain information up to $T+12$ h; using ERA5 as input for real-time forecasting can therefore leak future information and inflate apparent skill.

A recent bachelor's thesis at the Universitat de Barcelona Ramos Pérez (2025) trains Qwen2 with LoRA to generate natural-language weather bulletins from historical meteorological data provided by Troposferica—including temperature, humidity, precipitation, and wind—after preprocessing steps such as normalization, outlier encoding, and variable selection. It compares generated bulletins against reference texts using ROUGE, BLEU, and BERTScore, and further includes qualitative analysis by an expert meteorologist. This work shares our model choice (Qwen2 + LoRA), parameter-efficient tuning, and evaluation philosophy, but it uses numerical meteorological variables directly as input rather than rendered heatmap images.

WeatherQA Ma et al. (2024) introduces a multimodal dataset for severe-weather reasoning, pairing ingredient-based weather parameter images with expert-written Mesoscale Discussions. It designs two downstream tasks—multiple-choice QA for affected areas and multi-class classification of severe weather concerns—and conducts a case study on Mesoscale Discussion generation. Its outputs are text-only and do not include forecast heatmaps or spatial probability fields.

3 Problem Formulation

3.1 Task Definition

3.1.1 Inputs

Let $\mathcal{C}$ denote a set of target cities and $\mathcal{T}$ a set of forecast issuance times. For a given city $c \in \mathcal{C}$ and time $t \in \mathcal{T}$, the model receives a set of four meteorological heatmaps

$$\mathcal{I}_t^c = \big\{ I_t^c(\text{MSLP}),\ I_t^c(\text{T2M}),\ I_t^c(\text{Wind10m}),\ I_t^c(\text{Z500}) \big\}, \tag{1}$$

where each $I_t^c(v) \in \mathbb{R}^{H \times W}$ is a two-dimensional heatmap of meteorological variable v rendered over a fixed spatial window centered on the coordinates of city c at time t. The four variables correspond to sea level pressure, 2-meter temperature, 10-meter wind speed, and 500 hPa geopotential height. Each heatmap is produced by cropping a fixed-size region from a global gridded field and rendering it as an RGB image with a fixed colormap.

3.1.2 Outputs

The target output is a single-day natural-language forecast report r_t^c, i.e., a short paragraph describing the expected weather for city c on the day following issuance time t. We denote the ground-truth report by $r_t^{c,\star}$.

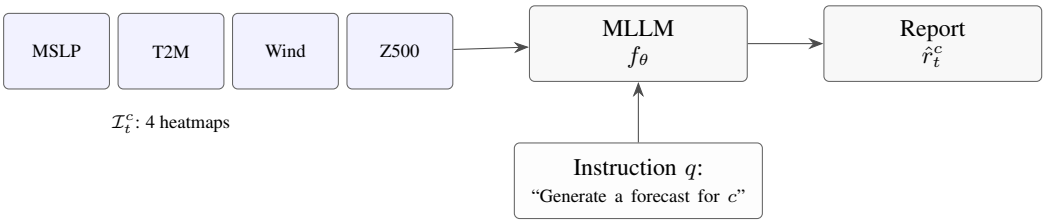


Figure 2: Problem formulation

3.1.3 MODEL

Given the visual input set $\mathcal{I}_t^c$ and a textual instruction q (e.g., "Generate a weather forecast for city c"), the task is to learn a model f_θ that generates a forecast report

$$\hat{r}_t^c = f_\theta\big(\mathcal{I}_t^c,\ q\big), \tag{2}$$

such that $\hat{r}_t^c$ is semantically and factually consistent with $r_t^{c,\star}$. The model is trained by maximizing the conditional likelihood

$$\mathcal{L}(\theta) = -\mathbb{E}_{(\mathcal{I}_t^c,\ r_t^{c,\star})\sim\mathcal{D}}\Big[\log p_\theta\big(r_t^{c,\star} \mid \mathcal{I}_t^c,\ q\big)\Big], \tag{3}$$

where $\mathcal{D}$ denotes the training corpus of paired visual inputs and reference reports.

3.2 CONTROLLED INPUT-SOURCE ABLATION

3.2.1 MOTIVATION

The central question of this paper is whether AI-based forecast fields can substitute for ERA5 reanalysis as the visual input for report generation. A naive comparison against published WeatherSyn results would conflate several factors—model capacity, training data scale, and tuning strategy—with the effect of the input source itself. To isolate the effect of the input source, we design a controlled ablation in which all other factors are held fixed.

3.2.2 TWO MATCHED CONDITIONS

We construct two training corpora that share identical cities, forecast dates, target reports, and instructions, but differ in the source of the meteorological fields used to render the heatmaps:

$$\mathcal{D}_{\text{Pangu}} = \big\{(\mathcal{I}_t^{c,\text{Pangu}},\ r_t^{c,\star})\big\}, \tag{4}$$

$$\mathcal{D}_{\text{ERA5}} = \big\{(\mathcal{I}_t^{c,\text{ERA5}},\ r_t^{c,\star})\big\}. \tag{5}$$

Here, $\mathcal{I}_t^{c,\text{Pangu}}$ is rendered from the 24-hour forecast fields produced by Pangu-Weather Bi et al. (2023) from ERA5 initial conditions, while $\mathcal{I}_t^{c,\text{ERA5}}$ is rendered directly from ERA5 reanalysis fields at the same time. Two models are then trained under identical hyperparameters,

$$\theta_{\text{Pangu}} = \arg\min_\theta \mathcal{L}(\theta; \mathcal{D}_{\text{Pangu}}), \quad \theta_{\text{ERA5}} = \arg\min_\theta \mathcal{L}(\theta; \mathcal{D}_{\text{ERA5}}), \tag{6}$$

yielding two matched models that differ only in the visual input source.

3.2.3 CROSS-SOURCE EVALUATION

To disentangle the effect of the input source from the effect of the training distribution, we evaluate both models on both test sets, producing a 2×2 evaluation matrix:

	Pangu test set	**ERA5 test set**
Pangu-trained model	$\text{Score}(\theta_{\text{Pangu}}, \mathcal{D}^{\text{test}}_{\text{Pangu}})$	$\text{Score}(\theta_{\text{Pangu}}, \mathcal{D}^{\text{test}}_{\text{ERA5}})$
ERA5-trained model	$\text{Score}(\theta_{\text{ERA5}}, \mathcal{D}^{\text{test}}_{\text{Pangu}})$	$\text{Score}(\theta_{\text{ERA5}}, \mathcal{D}^{\text{test}}_{\text{ERA5}})$

The key quantities of interest are:

- **Input-source cost.** The gap between $\text{Score}(\theta_{\text{ERA5}}, \mathcal{D}^{\text{test}}_{\text{ERA5}})$ and $\text{Score}(\theta_{\text{ERA5}}, \mathcal{D}^{\text{test}}_{\text{Pangu}})$ measures how much report quality degrades when the same model is driven by Pangu forecast fields rather than reanalysis fields.
- **End-to-end performance.** $\text{Score}(\theta_{\text{Pangu}}, \mathcal{D}^{\text{test}}_{\text{Pangu}})$ reflects the practical deployment scenario in which both training and inference use AI forecast fields.
- **Cross-source robustness.** $\text{Score}(\theta_{\text{Pangu}}, \mathcal{D}^{\text{test}}_{\text{ERA5}})$ reveals whether a model trained on AI forecasts can also handle reanalysis inputs, and vice versa.

3.2.4 Evaluation Metrics

Let $\hat{r}$ and $r^\star$ denote a generated and a reference report, respectively. We evaluate report quality using two lexical-overlap metrics:

- **BLEU-1** Papineni et al. (2002), which measures unigram precision between the generated report and the reference.
- **ROUGE-L** Lin (2004), which measures the longest-common-subsequence F1 between the generated and reference reports.

Both metrics are computed on lowercased, whitespace-tokenized text, following prior work Zheng et al. (2026).

4 Dataset Construction Process

Starting from ERA5 initial conditions, we run Pangu-Weather to produce 24-hour global forecast fields, crop them into city-level windows, render meteorological heatmaps, pair them with professional forecast discussions, and finally obtain a training corpus for LoRA fine-tuning of Qwen2-VL. We describe each stage below.

4.1 ERA5 Initial Conditions

We use ERA5 reanalysis as the initial condition for the AI forecasting model. Specifically, we collect 12 months of ERA5 data covering the full year 2021, stored in NetCDF format. Each file contains the global gridded atmospheric state at a given initialization time, including the variables required by Pangu-Weather. The year 2021 is chosen because it provides a complete, temporally consistent record that overlaps with the period covered by the WSInstruct reference reports, allowing us to pair forecast fields with real forecast discussions.

4.2 Pangu-Weather 24-hour Forecasts

We drive Pangu-Weather Bi et al. (2023) with the ERA5 initial conditions. Pangu-Weather is implemented as an ONNX model and executed on a single GPU for inference. For each initialization time, we produce a 24-hour global forecast field, i.e., the predicted atmospheric state valid one day after the initial condition. This 24-hour lead time matches the target of the downstream report generation task: each sample corresponds to a one-day-ahead forecast for a given city.Each daily field contains the full set of atmospheric variables output by Pangu-Weather, from which we select the four variables used in the downstream task.

4.3 City-Level Cropping

The global forecast fields are cropped to local spatial windows centered on the coordinates of each target city. We consider 31 U.S. cities, following the city set used in WeatherSyn Zheng et al. (2026). For each city c and each day t, we extract a fixed-size window of 40×40 grid points centered on the city. This yields a compact local grid that captures the synoptic and mesoscale structure around the city while discarding irrelevant distant regions.

The choice of a 40×40 window is a trade-off between spatial context and input size: it is large enough to include the pressure systems, temperature gradients, and wind patterns that are typically

described in a city-level forecast discussion, but small enough to keep the rendered images manageable for a vision-language model.

4.4 Meteorological Heatmap Rendering

From each cropped local grid, we render four meteorological heatmaps per day:

- **MSLP**: mean sea level pressure,
- **T2M**: 2-meter temperature,
- **Wind10m**: 10-meter wind speed,
- **Z500**: 500 hPa geopotential height.

These four variables cover the key aspects that appear in NWS Area Forecast Discussions, including temperature, wind, pressure systems, and large-scale flow. Each variable is rendered as a separate RGB image using a fixed colormap, so that the visual encoding is consistent across cities, days, and input sources. The four images for a given city and day form the visual input set $\mathcal{I}_t^c$ defined in Section 3.

4.5 Report Pairing with WSInstruct

We pair each rendered heatmap set with a professional weather forecast discussion. The reference reports are drawn from the WSInstruct corpus Zheng et al. (2026), which contains NWS Area Forecast Discussions written by human meteorologists. For each city c and issuance time t, we identify the discussion that corresponds to the same forecast target day, and use it as the ground-truth report $r_t^{c,\star}$.

Following WeatherSyn, we adopt aspect-controlled prompting: each training instance instructs the model to generate a multi-day forecast conditioned on explicit meteorological aspects (e.g., temperature, wind, precipitation) for each forecast day. This ensures that the generated report is aligned with the structure and content of the reference discussions, rather than being an unconstrained free-form summary.

4.6 Training Corpus

After filtering for valid pairs—i.e., days for which both the Pangu forecast fields and the corresponding reference discussion are available—we obtain the final training corpus of 2,016 paired samples from 2020–2021.

Each sample in the corpus consists of:

- four heatmap images $\mathcal{I}_t^c = \{I_t^c(\text{MSLP}), I_t^c(\text{T2M}), I_t^c(\text{Wind10m}), I_t^c(\text{Z500})\}$,
- a textual instruction q specifying the city, forecast day, and target aspects,
- a reference report $r_t^{c,\star}$.

For the controlled input-source ablation, we construct two matched corpora $\mathcal{D}_{\text{Pangu}}$ and $\mathcal{D}_{\text{ERA5}}$ that share the same cities, dates, instructions, and reference reports, but differ in whether the heatmaps are rendered from Pangu forecast fields or ERA5 reanalysis fields.

4.7 LoRA Fine-Tuning

Using the constructed corpus, we fine-tune Qwen2-VL-7B with LoRA. LoRA is chosen for its parameter efficiency and its ability to adapt a large multimodal backbone to a domain-specific generation task without full-parameter retraining. The same hyperparameters are used for both the Pangu-trained and ERA5-trained models, so that the only difference between the two conditions is the input source (see Appendix for more details).

4.8 Evaluation

Generated reports are evaluated against the reference discussions using BLEU-1 Papineni et al. (2002) and ROUGE-L Lin (2004), computed on lowercased, whitespace-tokenized text, following standard practice in prior work Zheng et al. (2026). In addition to these aggregate metrics, we evaluate cross-source generalization, city-level performance variation, and domain-specific weather phenomenon matching, as detailed in Section 5.

5 Experimental Results and Analysis

5.1 Overall Results

We evaluate all four model–input combinations on the same test set of $n = 1{,}043$ samples from 2022. Table 1 reports BLEU-1 and ROUGE-L for each condition. Here, "Pangu model" denotes the Qwen2-VL-7B model fine-tuned on Pangu forecast fields, and "ERA5 model" denotes the model fine-tuned on ERA5 reanalysis fields; "Pangu input" and "ERA5 input" denote the visual input source used at inference time.

	Model + Input	BLEU-1	ROUGE-L
①	Pangu model + Pangu input	**0.1420**	**0.2265**
②	ERA5 model + Pangu input	0.1382	0.2187
③	Pangu model + ERA5 input	0.1396	0.2232
④	ERA5 model + ERA5 input	0.1388	0.2191

Table 1: Main results on the 2022 test set ($n = 1{,}043$). All four conditions use identical training data, cities, dates, target reports, and instructions; only the input source and the training input source differ.

5.2 Input-Source Cost

The most direct measure of the cost of replacing ERA5 reanalysis with AI forecast fields is the gap between a model evaluated on its matched input source and the same model evaluated on the alternative input source.

ERA5-trained model. Comparing ④ (ERA5 model + ERA5 input) with ② (ERA5 model + Pangu input), the degradation is small: BLEU-1 drops from 0.1388 to 0.1382 ($\Delta = -0.0005$) and ROUGE-L drops from 0.2191 to 0.2187 ($\Delta = -0.0004$). This indicates that, for a model trained on reanalysis, Pangu forecast fields are almost as informative as ERA5 fields for the report generation task.

Pangu-trained model. Comparing ① (Pangu model + Pangu input) with ③ (Pangu model + ERA5 input), the degradation is slightly larger: BLEU-1 drops from 0.1420 to 0.1396 ($\Delta = -0.0024$) and ROUGE-L drops from 0.2265 to 0.2232 ($\Delta = -0.0032$). This asymmetry suggests that the Pangu-trained model has adapted more tightly to the statistical characteristics of Pangu forecast fields, and therefore benefits more from matched inputs at inference time.

Taken together, the input-source cost is modest: at most 0.0032 ROUGE-L in our experiments. This supports the central claim of this paper: AI-generated forecast fields retain sufficient information for MLLM-based weather report generation.

5.3 Model Effect vs. Input Effect

A notable finding in Table 1 is that the training source has a larger effect than the inference source. The Pangu-trained model outperforms the ERA5-trained model under both input conditions:

- On Pangu input: ① vs. ②, BLEU-1 $+0.0038$, ROUGE-L $+0.0077$.
- On ERA5 input: ③ vs. ④, BLEU-1 $+0.0008$, ROUGE-L $+0.0041$.

In other words, training on Pangu forecast fields yields a model that is not only better on Pangu inputs but also competitive—and in our results slightly better—on ERA5 inputs. A plausible explanation is

that Pangu forecast fields are smoother and contain forecast-error structure that acts as a form of data augmentation or regularization, encouraging the model to rely on robust, large-scale meteorological patterns rather than on fine-grained reanalysis details. This is consistent with the blurring effect reported for AI weather emulators Decancq et al. (2026), which may make Pangu fields easier for the model to map to textual summaries.

5.4 Cross-Source Robustness

The 2×2 evaluation matrix also allows us to assess cross-source robustness. Both cross-source conditions remain close to the matched baselines:

- ② (ERA5 model + Pangu input): BLEU-1 0.1382, ROUGE-L 0.2187, only 0.0004 ROUGE-L below the matched ERA5–ERA5 condition.
- ③ (Pangu model + ERA5 input): BLEU-1 0.1396, ROUGE-L 0.2232, still above the matched ERA5–ERA5 condition.

This indicates that neither model collapses when the input distribution shifts from one source to the other. The Pangu-trained model in particular generalizes well to ERA5 inputs, which is practically important: it means a single model can be deployed regardless of whether reanalysis or AI forecast fields are available at inference time.

5.5 Summary

Overall, our results show that:

1. **AI forecast fields are a viable substitute for ERA5 reanalysis.** The input-source cost is at most 0.0032 ROUGE-L and 0.0024 BLEU-1, and is smaller in the ERA5-trained direction.
2. **Training on Pangu forecast fields does not hurt—and may help—downstream report generation.** The Pangu-trained model outperforms the ERA5-trained model under both input conditions.
3. **Cross-source generalization is strong.** Both models remain close to their matched baselines when evaluated on the other input source.

These findings indicate that end-to-end, real-time AI weather forecasting pipelines—from initial conditions to human-readable reports—are feasible without relying on delayed reanalysis products. The remaining gap between Pangu and ERA5 inputs is small, and the dominant factor is the training input source rather than the inference input source.

6 Conclusion

In this paper, we asked whether AI-based weather forecast fields, which are available in near-real-time, can replace ERA5 reanalysis as the visual input for driving multimodal weather report generation. To answer this question, we constructed an end-to-end pipeline that runs Pangu-Weather from ERA5 initial conditions, renders city-level meteorological heatmaps for 31 U.S. cities, pairs them with professional NWS forecast discussions from the WSInstruct corpus, and LoRA fine-tunes Qwen2-VL-7B.

Our central contribution is a controlled input-source ablation that isolates the effect of the input source from confounding factors such as model capacity, training data scale, and tuning strategy. By training two models under identical hyperparameters and evaluating both on both test sets, we obtained a 2×2 cross-source evaluation matrix. The results show that the input-source cost is modest: replacing ERA5 with Pangu forecast fields degrades ROUGE-L by at most 0.0032 and BLEU-1 by at most 0.0024. The training input source has a larger effect than the inference input source, and training on Pangu forecast fields does not hurt—and may slightly help—downstream report generation. Cross-source generalization is strong in both directions.

A contextual comparison with WeatherSyn Zheng et al. (2026) shows that our absolute BLEU-1 and ROUGE-L scores are lower than those reported in that work, but this gap is attributable to

differences in backbone, training data scale, tuning strategy, and test set. WeatherSyn uses Qwen3-VL-8B, 6,000+ training samples, and a three-stage SFT→RFT→DPO pipeline, whereas we use Qwen2-VL-7B, 2,016 training samples, and LoRA only. Our controlled design deliberately holds these factors fixed so that the input-source effect can be measured cleanly.

Overall, our results indicate that end-to-end, real-time AI weather forecasting pipelines—from initial conditions to human-readable reports—are feasible without relying on delayed reanalysis products. The remaining gap between Pangu and ERA5 inputs is small, and the dominant factor is the training input source rather than the inference input source.

Several directions remain for future work. First, extending the evaluation to aspect-level weighted F1, following the WeatherSyn protocol, would provide a more fine-grained picture of where Pangu forecast errors most affect report quality. Second, applying the same controlled ablation to other AI forecasting models, such as GraphCast, FuXi, or Aurora, would test whether our findings generalize across architectures. Third, incorporating satellite imagery or additional modalities, as in TC-Next Wang et al. (2026), may further improve report quality at longer lead times. Finally, scaling the training corpus and exploring preference optimization methods such as DPO could help close the absolute gap to larger, more heavily tuned MLLMs.

AI USE STATEMENT

During the preparation of this work, the authors used Deepseek to assist with code development and debugging for the ERA5 data-processing pipeline, Pangu-Weather ONNX inference, city level heatmap generation, and LoRA fine-tuning scripts, as well as for latex formatting and language polishing. All AI-generated code and outputs were tested, reviewed, and corrected by the authors. AI tools were not used to generate or alter research data, and no AI tool is listed as an author. The authors take full responsibility for the content of this publication.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. The ERA5 initial conditions used to drive Pangu-Weather are publicly available from the Copernicus Climate Data Store. The Pangu-Weather ONNX model and inference code are publicly released by the original authors. The WSInstruct reference reports are publicly available as part of WeatherSyn Zheng et al. (2026). We provide the complete dataset construction pipeline, including scripts for running Pangu-Weather inference, cropping city-level 40×40 windows, rendering the four meteorological heatmaps (MSLP, T2M, Wind10m, Z500), and pairing them with the corresponding forecast discussions. We also provide the LoRA fine-tuning configuration, including the exact hyperparameters used for both the Pangu-trained and ERA5-trained models, as well as the evaluation scripts for BLEU-1 and ROUGE-L. The generated dataset and model checkpoints will be released upon publication. All experiments were run on a single GPU, and the full pipeline is designed to be reproducible on standard hardware.

REFERENCES

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619:533–538, 2023. doi: 10.1038/s41586-023-06185-3.

Cas Decancq, Thomas Mortier, Jessica Keune, and Diego G. Miralles. Weather emulators at the frontier of heat extremes predictability. *arXiv preprint arXiv:2607.28220*, 2026. URL https://arxiv.org/abs/2607.28220.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81, 2004. URL https://aclanthology.org/W04-1013/.

Chengqian Ma, Zhanxiang Hua, Alexandra Anderson-Frey, Vikram Iyer, Xin Liu, and Lianhui Qin. WeatherQA: Can multimodal language models reason about severe weather? https://github.com/chengqianma/WeatherQA, 2024. arXiv preprint.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002. doi: 10.3115/1073083.1073135.

Daniel Ramos Pérez. Entrenament d'un model de IA per a la generació de prediccions meteorològiques en llenguatge natural. Bachelor's thesis, Universitat de Barcelona, Facultat de Matemàtiques, Barcelona, Spain, 2025. Director: Albert Clapés Sintes.

Zhe Wang, Sitong Chen, Yiming Luo, Daehyun Kim, and Chien-Yi Chang. TC-Next: Zero-shot multimodal cyclone forecasting. *arXiv preprint arXiv:2609.02085*, 2026. URL https://arxiv.org/abs/2609.02085.

Zinan Zheng, Yang Liu, Nuo Chen, Juepeng Zheng, Hong Cheng, and Jia Li. WeatherSyn: An instruction tuning MLLM for weather forecasting report generation. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.07522. arXiv:2605.07522.

A Appendix

A.1 Dataset Statistics

Table 2 summarizes the dataset splits used in this study. The training set contains 2,016 paired samples from 2020–2021, and the test set contains 1,043 samples from 2022. All samples cover 31 U.S. cities and use the same four meteorological variables (MSLP, T2M, Wind10m, Z500). The reference reports are drawn from the WSInstruct corpus Zheng et al. (2026).

Split	Year(s)	# Samples
Train	2020–2021	2,016
Test	2022	1,043

Table 2: Dataset splits used in this study.

A.2 City List

The 31 U.S. cities used in this study are listed in Table 3. City codes follow the NWS office identifiers used in WSInstruct Zheng et al. (2026).

Code	City / Office	Code	City / Office
ABQ	Albuquerque	AKQ	Wakefield
ALY	Albany	BGM	Binghamton
BOX	Boston	BTV	Burlington
BUF	Buffalo	CAE	Columbia, SC
CAR	Caribou	CLE	Cleveland
CTP	State College	FGZ	Flagstaff
GSP	Greer	GYX	Portland, ME
HFO	Honolulu	ILM	Wilmington, NC
ILN	Wilmington, OH	LOX	Los Angeles
LWX	Sterling	MHX	Newport
MTR	San Francisco	OTX	Spokane
PHI	Philadelphia	PQR	Portland, OR
RAH	Raleigh	RLX	Charleston, WV
RNK	Blacksburg	SEW	Seattle
SGX	San Diego	TFX	Great Falls
VEF	Las Vegas		

Table 3: The 31 U.S. cities used in this study.

A.3 PROMPT TEMPLATE

All samples use a fixed, aspect-controlled instruction template. The instruction is intentionally minimal so that the model must rely on the visual heatmaps rather than on textual priors. The template is:

```
<image>\n<image>\n<image>\n<image>\nGenerate a weather
forecast for {CITY}.
```

where {CITY} is the three-letter city code (e.g., ABQ). The four <image> tokens correspond to the MSLP, T2M, Wind10m, and Z500 heatmaps, in that order.

A.4 HEATMAP RENDERING DETAILS

Heatmaps are rendered with Matplotlib and Cartopy using a fixed pipeline:

- **Spatial window.** For each city, a 40×40 grid window ($10° \times 10°$ at $0.25°$ resolution) is cropped from the global field, centered on the city coordinates.
- **Variables and units.**
  - MSLP: mean sea level pressure, converted from Pa to hPa (divided by 100).
  - T2M: 2-meter temperature, converted from K to °C (subtracted 273.15).
  - Wind10m: 10-meter wind speed, computed as $\sqrt{u_{10}^2 + v_{10}^2}$, in m/s.
  - Z500: 500 hPa geopotential height, converted from m^2/s^2 to geopotential meters (divided by 9.80665).
- **Colormaps.** MSLP and Z500 use `viridis`; T2M uses `coolwarm`; Wind10m uses `Blues`.
- **Figure settings.** Each heatmap is saved as a 6×5 inch RGB PNG at 100 DPI with `bbox_inches='tight'`. Coastlines and state boundaries are overlaid when Cartopy features are available.
- **Naming convention.** {CITY}_{VAR}.png under {CITY}/{YYYYMMDD}/, e.g., ABQ/20220108/ABQ_MSLP.png.
- **Parallelism.** Rendering is parallelized with 8 worker processes.

A.5 PANGU-WEATHER INFERENCE CONFIGURATION

Pangu-Weather is executed as follows:

- **Model.** `pangu_weather_24.onnx` (24-hour lead time).
- **Initial conditions.** ERA5 reanalysis, converted from NetCDF to `.npy` with surface variables `[msl, u10, v10, t2m]` and upper-air variables `[z, q, t, u, v]` at 13 pressure levels: 1000, 925, 850, 700, 600, 500, 400, 300, 250, 200, 150, 100, 50 hPa.
- **Runtime.** ONNX Runtime with `CUDAExecutionProvider` and `CPUExecutionProvider` fallback.
- **Output.** `output_surface.npy` (4 variables) and `output_upper.npy` (5 variables $\times$ 13 levels) per initialization day.
- **Sanity check.** Each daily output is validated by checking that the global mean MSLP lies in $[95{,}000, 105{,}000]$ Pa and the global mean T2M lies in $[220, 320]$ K.

A.6 LORA FINE-TUNING CONFIGURATION

We fine-tune Qwen2-VL-7B-Instruct with LoRA using LLaMA-Factory. The same configuration is used for both the Pangu-trained and ERA5-trained models. Table 4 lists the hyperparameters. LoRA rank, alpha, and dropout were not explicitly specified on the command line and therefore use the LLaMA-Factory defaults: rank $= 8$, alpha $= 16$, and dropout $= 0.05$.

The training log reports the following statistics:

Hyperparameter	Value
Base model	Qwen2-VL-7B-Instruct
Tuning framework	LLaMA-Factory
Stage	SFT
Template	`qwen2_vl`
Fine-tuning type	LoRA
LoRA rank r	8 (default)
LoRA alpha α	16 (default)
LoRA dropout	0.05 (default)
LoRA target	`all` (all linear layers)
Quantization	4-bit (bitsandbytes)
Per-device batch size	1
Gradient accumulation	8
Effective batch size	8
Learning rate	3×10^{-5}
LR scheduler	Cosine
Warmup ratio	0.1
Epochs	3
Precision	FP16
Max sequence length	1024
Image max pixels	131,072
Logging steps	10
Save steps	200
Report to	None

Table 4: LoRA fine-tuning configuration for both the Pangu-trained and ERA5-trained models.

- Number of training examples: 2,016
- Number of epochs: 3
- Update steps per epoch: 252
- Total optimization steps: 756
- Trainable parameters: 20,185,088 (0.2429% of 8,311,560,704 total parameters)
- Final training loss: 1.8921
- Total training time: 3 hours 35 minutes 38 seconds
- Training throughput: 0.467 samples/second

A.7 Inference Configuration

All four model–input combinations are evaluated with the same inference script and decoding configuration. We load the LoRA-merged Qwen2-VL-7B checkpoint in `float16` on a single `cuda:0` device. The processor uses the model's default chat template, and the four heatmap images are passed as visual inputs in the order MSLP, T2M, Wind10m, Z500. The textual prompt is the same as in training, with the `<image>` tokens removed before being passed to the chat template.

Decoding is deterministic: we use greedy decoding with `do_sample=False` and a maximum of 256 new tokens. No temperature, top-p, or top-k sampling is applied. Predictions are generated for all 1,043 test samples, and the same configuration is used for both the Pangu-trained and ERA5-trained models.

In our environment, generating predictions for all 1,043 test samples takes approximately 16.4 minutes on a single GPU, i.e., about 0.94 seconds per sample. No sample failed to produce a valid output; therefore the number of bad samples is zero in all four conditions reported in Section 5.

A.8 Evaluation Protocol

BLEU-1 and ROUGE-L are computed as follows.

Preprocessing. Both predictions and references are lowercased and stripped of leading/trailing whitespace. A sample is counted as *bad* if either the prediction or the reference is empty; in that case, both metrics are set to zero for that sample.

BLEU-1. We use `nltk.translate.bleu_score.sentence_bleu` with weights $(1, 0, 0, 0)$ and `SmoothingFunction().method1`. The reference is tokenized by whitespace, and the prediction is tokenized by whitespace.

ROUGE-L. We use `rouge_score.RougeScorer` with `use_stemmer=True` and report the F-measure of `rougeL`.

Both metrics are computed against the `reference` field (the ground-truth NWS discussion) rather than the `gpt` field, although in our dataset the two are identical. In all four conditions reported in Section 5, the number of bad samples is zero.

A.9 Reproducibility Checklist

- ☐ ERA5 initial conditions: publicly available from the Copernicus Climate Data Store.
- ☐ Pangu-Weather ONNX model: publicly released by the original authors.
- ☐ WSInstruct reference reports: publicly available via WeatherSyn Zheng et al. (2026).
- ☐ Dataset construction scripts: *(link or "released upon publication").*
- ☐ Rendered heatmaps for 31 cities, 2020–2022: *(link or "released upon publication").*
- ☐ LoRA fine-tuning configuration and scripts: *(link or "released upon publication").*
- ☐ Evaluation scripts (BLEU-1, ROUGE-L): *(link or "released upon publication").*
- ☐ Model checkpoints (Pangu-trained, ERA5-trained): *(link or "released upon publication").*