BlendCast: Teaching Vision-Language Models to Anticipate Member Skill in Weather Ensembles

Tao Han[1,2] **Fenghua Ling**[2] **Dazhao Du**[1] **Haoxi Li**[1]
Zhibin Wen[3] **Guangtao Zhang**[3] **Song Guo**[1,*] **Lei Bai**[2,*]
[1]The Hong Kong University of Science and Technology
[2]Shanghai Artificial Intelligence Laboratory
[3]Southern University of Science and Technology

Abstract

AI weather models are fast and skillful, yet their errors are strongly conditional because the best member changes with lead time, variable, initialization state, and atmospheric regime. Operational deterministic multi-model forecasting therefore requires more than averaging strong forecasters. It requires anticipating which member deserves trust before future analyses exist. **BlendCast** turns this anticipation problem into a vision-language forecasting task. Given meteorological maps and structured ensemble diagnostics, the model reads the current weather situation, reasons about prospective member reliability, and converts that judgment into deterministic per-variable ensemble weights. To teach this behavior, BlendCast first gives the model demonstrations of what member-skill reasoning and weight decisions should look like, then refines the resulting policy with multi-variable reinforcement learning so that its own analyses are rewarded when they lead to better, physically consistent forecasts. On held-out 2025 forecasts, BlendCast reduces WRMSE by **2.6%** over equal weighting and by **2.0%** over conventional ensemble post-processing, closing **63.2%** of the gap to oracle blending. These results show that, with multimodal weather evidence and reward-aligned training, VLMs can act as real-time forecasters of forecasters for deterministic AI weather ensembles.[2]

1 Introduction

Data-driven weather forecasting has turned medium-range numerical weather prediction (NWP) into a landscape of many strong forecasters rather than a single dominant model. GraphCast [1], Pangu-Weather [2], FengWu [3], FengWu-GHR [4], FuXi [5], Aurora [6], Stormer [7], and AIFS [8], together with newer systems such as VA-MoE [9], now achieve strong skill on benchmarks such as WeatherBench2 [10]. Yet their value is not arranged as a fixed leaderboard. Like classical error growth and forecast uncertainty, model reliability is flow-dependent [11], and a member that is excellent for one variable, lead, or weather regime may be ordinary for another. For an operational artificial intelligence (AI) ensemble, the central question therefore shifts from *which model is best on average* to *which member should be trusted for this forecast case*, using forecast-time information.

Figure 1 distills this complementarity into four observations that make prospective member-skill anticipation concrete. **Obs. 1, lead-time and variable dependence.** The strongest member changes across forecast horizons and target variables, so no model provides a uniformly reliable ordering of skill. **Obs. 2, initialization-state dependence.** even for a fixed target, the strongest member changes across initialization states, showing that skill is tied to the current forecast case rather than

*Corresponding authors: Song Guo (songguo@cse.ust.hk) and Lei Bai (bailei@pjlab.org.cn).
[2]Code is available at https://github.com/taohan10200/BlendCast.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

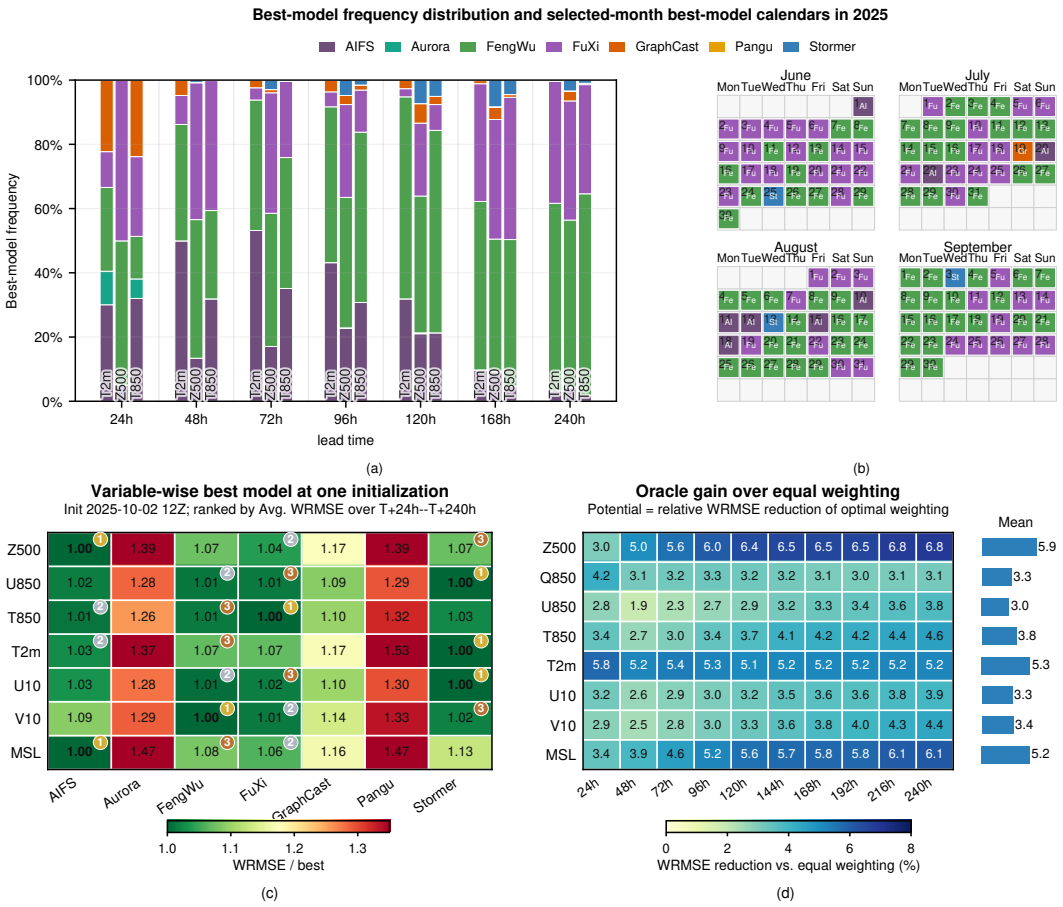


Figure 1: **Four empirical observations motivating BlendCast. (a) Lead-time and variable dependence.** The strongest member changes across forecast horizons and target variables, showing that model skill is not globally ordered. **(b) Initialization-state dependence.** Even for a fixed target, the strongest member changes across initialization states, indicating forecast-case-dependent skill. **(c) Variable-wise member specialization.** Within one forecast case, different variables favor different members, motivating per-variable rather than global weighting. **(d) Headroom beyond equal weighting.** Oracle variable-wise blending leaves measurable improvement over equal weighting, motivating a policy that assigns better weights from information available at forecast time.

to a static historical ranking. **Obs. 3, case-level variable dependence.** within one initialization, different variables can favor different members, motivating per-variable trust instead of a single global weight vector. **Obs. 4, headroom beyond equal weighting.** a retrospective oracle blend leaves measurable improvement over equal weighting, indicating that better weight allocation can further improve deterministic AI NWP ensembles. Here the oracle is a hindsight upper bound. After ECMWF's fifth-generation atmospheric reanalysis (ERA5) [12] verification is available, we solve the classical constrained linear forecast-combination problem for the non-negative per-variable weights that minimize weighted root-mean-square error (WRMSE) [13, 14]. Together, these observations show that member skill is a relationship between a forecaster and the evolving atmospheric state. Each member has moments of strength, but neither a single-model ranking nor universal equal weighting is sufficient. Because the best weights are not available in real-time operation, the inference-time challenge is to anticipate member-skill patterns from the meteorological state itself.

Existing ensemble methods offer useful but partial answers. Equal weighting remains a strong baseline because it is robust, transparent, and hard to overfit [11]. Classical post-processing methods such as Bayesian model averaging (BMA) [15], ensemble model output statistics (EMOS) [16], and regime-dependent calibration [17] improve reliability by learning from past errors. Recent AI ensemble systems push the complementary frontier of generating or emulating probabilistic members

with useful spread, but still commonly aggregate members symmetrically or emphasize distributional calibration rather than case-wise member fusion [18–23]. The missing piece is a forecast-time policy that assigns variable-dependent trust from the current meteorological evidence instead of averaging members uniformly. This is where vision-language models are appealing. Recent vision-language models (VLMs) such as LLaVA [24], MM1 [25], and Cambrian-1 [26] are not merely image classifiers with text output; they provide an interface for turning visual evidence and structured context into reasoned, executable decisions. Weather ensembles need exactly this kind of decision maker, a model that can forecast the forecasters.

We present BlendCast, a VLM-based policy for turning multimodal weather evidence into per-variable ensemble weights. For each forecast case, BlendCast reads meteorological images and ensemble diagnostics, forms a concise analysis of prospective member reliability, and outputs a normalized weight vector for each target variable. Supervised fine-tuning (SFT) provides the entry point. Oracle weights derived from ERA5 [12] teach the shape of good decisions, while teacher-generated meteorological rationales teach the language of member-skill reasoning. But imitation alone is too weak for a task whose essence is anticipation. The two SFT targets are paired in demonstrations, not produced by the same reasoning process; a model can learn to sound meteorological without learning whether its own diagnosis actually improves a forecast.

BlendCast therefore follows SFT with physically grounded multi-variable reinforcement learning, building on Group Relative Policy Optimization (GRPO)-style optimization in DeepSeekMath [27] and DeepSeek-R1 [28]. Multi-variable GRPO (MV-GRPO) turns each generated rationale and weight vector into an accountable forecast action. The weights are applied to frozen AI NWP members, the blended fields are scored offline against ERA5, and trajectories receive advantage only when their own decisions improve forecast quality while preserving valid weights and pressure–wind consistency. The result is a closed loop between seeing, reasoning, and weighting. At inference time, no future analyses are available; nevertheless, the policy has been trained to read the present atmospheric state as evidence about which members are likely to be reliable.

Our contributions are summarized below.

C1. A forecast-the-forecasters framework. We recast AI weather ensembling as prospective member-skill anticipation. The aim is not to build another forecaster, but to decide which forecasters to trust before the future is seen. BlendCast realizes this as an interpretable VLM policy that turns forecast-time evidence into per-variable trust over frozen NWP members.

C2. From multimodal weather evidence to executable weights. We design a compact state that lets the model view the ensemble like a forecaster through means, spread, per-member anomalies, and structured diagnostics. Split SFT teaches decision language and form, while MV-GRPO makes the model's reasoning answer to forecast quality, valid weights, and physical consistency.

C3. Approaching the oracle without future analyses. Across variables and lead times, BlendCast consistently moves the ensemble away from uniform averaging and toward the wait-and-see oracle reference, especially in cases where member skill is strongly regime- and lead-dependent. This supports the paper's central claim: with the right evidence and training signal, a VLM can learn forecaster-like judgment about member reliability before any verification field exists.

2 Related Work

AI weather forecasting and ensemble post-processing. Data-driven NWP has progressed from deterministic medium-range forecasters to broader forecasting systems, including Pangu-Weather [2], GraphCast [1], FengWu [3], FuXi [5], FengWu-GHR [4], Stormer [7], FengWu-4DVar [29], VA-MoE [9], NeuralGCM [30], AIFS [8], Aurora [6], Aardvark Weather [31], and FuXi Weather [32]. In parallel, ensemble forecasting and post-processing represent uncertainty and correct systematic error using multiple trajectories or historical calibration [11]. Bayesian Model Averaging (BMA) [15], ensemble model output statistics (EMOS) [16], and regime-dependent post-processing [17] learn combination or calibration rules from past cases, while recent AI ensemble work focuses on producing probabilistic members or emulating ensemble spread, including Graph-EFM [33], GenCast [18], SEEDS [19], NeuralGCM [30], and FuXi-ENS [23]. These lines make forecasts stronger and ensembles richer, but leave comparatively less attention to forecast-time, map-conditioned fusion of already available AI NWP members.

Vision-language reasoning and reinforcement learning. Vision-language models have progressed from CLIP [34] image-text representation learning and Flamingo [35] few-shot multimodal prompting to LLaVA [24] visual instruction following and multi-image reasoning in MM1 [25] and Cambrian-1 [26]. ReAct [36], Toolformer [37], DeepSeekMath [27], and DeepSeek-R1 [28] further connect model reasoning to executable decisions and outcome-based feedback. Meteorological multimodal work remains early, with WeatherQA [38] and CLLMate [39] focusing on severe-weather reasoning, event understanding, or weather/climate event forecasting benchmarks. BlendCast sits between these lines of work. It does not introduce another AI NWP forecaster or generate new stochastic members, but uses VLM reasoning to anticipate member skill from meteorological maps and assign per-variable trust over an existing deterministic multi-model AI NWP ensemble.

3 Method

We first cast ensemble weighting as a prospective approximation to a retrospective oracle blend in Section 3.1. Section 3.2 then introduces BlendCast as a VLM policy over weather evidence and describes how the VLM is trained as a blend-forecasting policy model.

3.1 The Prospective Blending Problem

The retrospective oracle. Let $\mathcal{M}$ be the set of AI NWP members and $\mathcal{V}$ the target weather variables (circulation, thermodynamics, winds, and pressure). For member m and variable v, the forecast field at initialization t_0 and lead h is $\hat{F}_{m,v} \in \mathbb{R}^{H\times W}$. The blended forecast is a per-variable linear combination

$$\hat{F}_v^{\,a}(x,y) = \sum_{m\in\mathcal{M}} w_{v,m}\,\hat{F}_{m,v}(x,y), \tag{1}$$

with $w_{v,m} \geq 0$ and $\sum_m w_{v,m} = 1$. If ERA5 verification F_v^{ERA5} is available, optimal weights can be computed exactly by solving a separate quadratic program (QP) for each variable.

$$w_v^* = \arg\min_{\substack{w_m\geq 0\\ \sum_m w_m = 1}} \sqrt{\frac{\sum_{x,y} c_y \big(\sum_m w_m \hat{F}_{m,v}(x,y) - F_v^{\mathrm{ERA5}}(x,y)\big)^2}{\sum_{x,y} c_y}}, \tag{2}$$

where $c_y = \cos(\phi_y)$ is the latitude weight (ϕ_y denotes the latitude of grid row y). Per-variable weights are necessary because fields have different error structures. A member leading on 500 hPa geopotential need not lead on near-surface temperature, so one global weight vector would be a category error. Since the objective is continuous and the constraint set $\Delta^{|\mathcal{M}|} = \{w \geq 0 \mid \sum_m w_m = 1\}$ is compact and convex, at least one oracle solution w_v^* exists [40]. The solution is unique only under the stronger condition that the corresponding quadratic error is strictly convex on the simplex, for example when the member forecast differences are linearly independent on the active affine hull.

The gap without future analyses. Equation (2) defines the ceiling of what blending can achieve, but w_v^* cannot be computed operationally because ERA5 is produced hours to days after the forecast valid time. This turns the optimization problem into a forecasting problem. The task is to anticipate w_v^* (or a close proxy) from information available at t_0, namely the ensemble forecasts themselves and the spatial structure of their disagreement.

This has the structure of a *two-stage stochastic program* [41]. The weight vector w_v must be committed at t_0 using only the ERA5-free state $s \in \mathcal{S}$, while the second-stage cost WRMSE_v is revealed only after the future verification field F_v^{ERA5} is observed. Let Π_v denote the set of measurable maps from $\mathcal{S}$ to $\Delta^{|\mathcal{M}|}$. The optimal prospective policy solves

$$\pi_v^* = \arg\min_{\pi_v\in\Pi_v} \mathbb{E}_{s,\,F_v^{\mathrm{ERA5}}}\Big[\mathrm{WRMSE}_v\Big(\pi_v(s),\ \hat{\mathbf{F}}_v,\ F_v^{\mathrm{ERA5}}\Big)\Big], \tag{3}$$

where $\hat{\mathbf{F}}_v = \{\hat{F}_{m,v}\}_{m\in\mathcal{M}}$ collects the member forecast fields and the expectation is over the joint distribution of current ensemble states and the future verification they will verify against. The oracle Eq. (2) is the *wait-and-see* counterpart. It observes F_v^{ERA5} before committing, so it solves a separate QP per realization. By information monotonicity, the oracle can never be outperformed by any causal policy, and the gap

$$\Delta_v(\pi) = \mathbb{E}[\mathrm{WRMSE}_v(\pi_v(s))] - \mathbb{E}[\mathrm{WRMSE}_v(w_v^*)] \ \geq\ 0 \tag{4}$$

is the expected value of perfect information (EVPI) for variable v [41]. BlendCast is a parameterized prospective policy π_θ trained to drive $\Delta_v(\pi_\theta)$ toward zero across all $v \in \mathcal{V}$.

Proposition 1 (Oracle dominance)**.** *For any measurable prospective policy π_v mapping $\mathcal{S}$ to $\Delta^{|\mathcal{M}|}$, $\Delta_v(\pi_v) \geq 0$.*

Proof. By definition, w_v^* minimizes $\mathrm{WRMSE}_v(\cdot,\, F_v^{\mathrm{ERA5}})$ over $\Delta^{|\mathcal{M}|}$ for every realization, so $\mathrm{WRMSE}_v(\pi_v(s),\, F_v^{\mathrm{ERA5}}) \geq \mathrm{WRMSE}_v(w_v^*,\, F_v^{\mathrm{ERA5}})$ holds pointwise. Taking expectations preserves the inequality.

3.2 Vision-Language as Prospective Policy

Equation (3) defines the approximation target but leaves open the choice of function class for π_v. The mapping from $\mathcal{S}$ to $\Delta^{|\mathcal{M}|}$ imposes three coupled demands. It must perceive spatially structured signal from two-dimensional ensemble anomaly maps. It must reason explicitly about whether observed member deviations reflect genuine predictive skill or initialization noise, a step requiring chains of meteorological inference rather than direct pixel-to-weight regression. And it must produce complete, normalized weight vectors for every target variable. Systems such as LLaVA [24], Qwen2.5-VL [42], and Cambrian-1 [26] demonstrate strong visual-to-text reasoning capabilities that address all three demands. The visual encoder perceives map structure from pixels, the language model supplies meteorological domain knowledge encoded during pretraining, and the chain-of-thought decoder forms explicit reasoning steps before committing to a structured weight decision. We therefore parameterize π_θ as a VLM with state $s = (\mathbf{I}, \mathbf{x})$. The policy generates a *trajectory* consisting of a reasoning chain $\mathbf{r}_\theta$ followed by a structured weight decision $\mathbf{w}_\theta$.

$$\pi_\theta(s) = (\mathbf{r}_\theta,\, \mathbf{w}_\theta), \qquad \mathbf{w}_\theta = \{w_{\theta,v}\}_{v\in\mathcal{V}}, \quad w_{\theta,v} \in \Delta^{|\mathcal{M}|}. \tag{5}$$

The joint distribution over trajectories factorizes autoregressively as

$$p_\theta(\mathbf{r}, \mathbf{w} \mid s) = \underbrace{p_\theta(\mathbf{r} \mid s)}_{\text{reasoning tokens}} \cdot \underbrace{p_\theta(\mathbf{w} \mid s,\, \mathbf{r})}_{\text{decision tokens}}, \tag{6}$$

which makes the causal structure explicit. Weight tokens are generated after and conditioned on the reasoning chain, so the decision is a downstream consequence of the stated meteorological argument.

Observable state. The image component $\mathbf{I}$ contains $|\mathcal{V}| \times (2 + |\mathcal{M}_v|)$ maps per forecast case, with one ensemble mean, one spread map, and $|\mathcal{M}_v|$ per-model anomaly maps for each target variable, where $\mathcal{M}_v \subseteq \mathcal{M}$ is the subset of members that provide forecasts for variable v. The text component $\mathbf{x}$ is a structured descriptor

$$\mathbf{x} = \Big(t_0,\; h,\; \mathcal{M},\; \{\mu_{v,r},\, \sigma_{v,r},\, \{\delta_{m,v,r}\}_{m\in\mathcal{M}}\}_{v\in\mathcal{V},\, r\in\mathcal{R}}\Big), \tag{7}$$

where $\mu_{v,r}$ is the ensemble mean of variable v over region $r \in \mathcal{R}$, $\sigma_{v,r}$ is the ensemble spread, and $\delta_{m,v,r} = \bar{F}_{m,v,r} - \mu_{v,r}$ is the signed deviation of member m from the ensemble mean, and $\bar{F}_{m,v,r}$ denotes the spatial average of $\hat{F}_{m,v}$ over region r. Regions cover the main verification domains: Global, Northern Hemisphere (NH), Europe (EU), and East Asia (EA), denoted $\mathcal{R} = \{\mathrm{Global},\, \mathrm{NH},\, \mathrm{EU},\, \mathrm{EA}\}$; all entries are computable from member forecasts alone. The per-model anomaly maps are the primary decision signal. A spatially coherent deviation pattern distinguishes genuine member skill from initialization noise in ways that regional scalar summaries cannot capture. The oracle weights w_v^* never enter the observable state. They appear only in the training metadata as SFT labels and reinforcement learning (RL) reward normalizers (see Appendix G for map taxonomy and a fully annotated sample).

Supervised warm-start. We initialize π_θ via multimodal SFT, pairing each ERA5-free state s with oracle weights w_v^* and a teacher-generated meteorological rationale. The teacher rationale is produced by a large frozen model from the same observable maps and statistics. It is a post-hoc diagnostic explanation of the oracle decision, not a causal derivation. SFT therefore teaches the policy domain vocabulary and the weight-output format, a necessary prior, but cannot verify that a particular visual diagnosis *leads to* better weights. The two output parts demand different precision, so BlendCast applies a split cross-entropy (CE) objective.

$$\mathcal{L}_{\mathrm{SFT}} = \mathrm{CE}_{\alpha_r}(\text{rationale tokens}) + \mathrm{CE}_{\alpha_w}(\text{weight tokens}), \qquad \alpha_r > \alpha_w, \tag{8}$$

with softer label smoothing on rationale tokens and stricter supervision on weight tokens to preserve the numerical precision required for valid ensemble decisions.

Closing the causal loop via MV-GRPO. Outcome-based reinforcement learning turns the SFT policy from a mimic of demonstrations into a policy whose own decisions are judged by forecast quality [27, 28, 43]. We adopt Group Relative Policy Optimization (GRPO) [27] and extend it to multi-variable weather blending. For each training case, the policy samples G rationale–weight trajectories; the weights are applied to frozen AI NWP members and scored offline against ERA5, which is never a model input. The main reward is the fraction of oracle-attainable WRMSE improvement recovered by the trajectory.

$$R_{i,v}^{\mathrm{wrmse}} \;=\; \frac{\mathrm{WRMSE}_{\mathrm{eq},v} - \mathrm{WRMSE}_{i,v}}{\max\!\big(\,\mathrm{WRMSE}_{\mathrm{eq},v} - \mathrm{WRMSE}_{\mathrm{opt},v},\;\; 10^{-3}\,\mathrm{WRMSE}_{\mathrm{eq},v}\big)}, \tag{9}$$

where $\mathrm{WRMSE}_{\mathrm{eq},v}$ is the equal-weight baseline, $\mathrm{WRMSE}_{\mathrm{opt},v} = \mathrm{WRMSE}_v(w_v^*)$ is the oracle, and oracle normalization prevents easy cases from dominating the RL signal. We supplement it with two structural signals, namely a validity reward for complete, non-negative, normalized weights and a geostrophic-consistency reward that discourages pressure–wind mismatches. Writing $\mathcal{V}^+ = \mathcal{V} \cup \{\mathrm{valid}, \mathrm{geo}\}$, MV-GRPO normalizes each component within the rollout group and aggregates them with lead-time weighting.

$$A_i \;=\; \phi(h) \sum_{v \,\in\, \mathcal{V}^+} \omega_v \,\frac{R_{i,v} - \bar{R}_v}{\hat{\sigma}_v + \epsilon}, \tag{10}$$

where $\omega_v > 0$ are per-component weights, $\bar{R}_v$ and $\hat{\sigma}_v$ are within-group mean and standard deviation, and $\phi(h)$ concentrates the signal at 72–120 h where oracle gap and inter-member spread are largest. The policy is then updated with the standard KL-regularized clipped GRPO objective relative to the SFT reference model, preserving tool-call format while letting forecast outcomes reshape the reasoning–weighting policy. Detailed reward definitions, lead-time weights, and optimization hyperparameters are given in Appendices I and J.

4 Experiments

4.1 Experimental Setup

Data and metrics. Both SFT and reinforcement learning (RL) use real AI NWP forecasts from 2025 paired with ERA5 analyses for oracle-weight construction and offline reward computation. The training split comprises the 06Z and 18Z Coordinated Universal Time (UTC) initialization times, yielding 9,501 samples across 235 unique dates and ten lead times from T+24 h to T+240 h. The 00Z and 12Z UTC initializations are held out entirely as the test set and are never seen during training. All metrics are computed offline after the forecast valid time, and prompts as well as teacher rationales never expose future ERA5 fields or evaluation scores. The primary metric is global latitude-weighted WRMSE following WeatherBench2 [10], evaluated over eight variables: 500 hPa geopotential height (Z500), 850 hPa specific humidity (Q850), 850 hPa zonal wind (U850), 850 hPa temperature (T850), 2-meter temperature (T2M), 10-meter zonal wind (U10), 10-meter meridional wind (V10), and mean sea-level pressure (MSL). Tables 1 and 2 report the main even-day results (D2/D4/D6/D8/D10 = 48/96/144/192/240 h) for upper-air, thermal, and pressure variables; even-day surface winds and all odd-day results appear in Appendix D.

Training details. BlendCast uses Qwen3.5-9B [44] as the VLM policy. We first warm-start it with SFT, using ERA5-derived oracle weights as decision labels and Qwen3.5-397B [44] rationales as meteorological language supervision, then refine it with MV-GRPO rewards computed from ERA5-verified forecast errors. All training ran on four H20 graphics processing units (GPUs). The selected SFT checkpoint required about 138 hours, and the MV-GRPO refinement used here required about 160 hours. Appendix J reports training and reward settings and distinguishes the updated SFT budget in the released configuration.

Baselines. We compare seven individual AI NWP members (Pangu-Weather, Stormer, FengWu, FuXi, GraphCast, Aurora, AIFS), three classical ensemble methods (equal-weight averaging, BMA,

EMOS), three proprietary VLMs deployed as ensemble weight predictors (Kimi-K2.5, InternVL-S1-Pro, Qwen3.5-397B), and two BlendCast variants (SFT-only and SFT+MV-GRPO). For the proprietary VLM baselines, if an application programming interface (API) call fails or the returned response cannot be parsed into a complete valid weight object, the corresponding forecast case falls back to Equal-weight rather than being dropped from WRMSE evaluation.

4.2 Main Results

Table 1: 2025 test-set results, upper-air variables, even-day leads (D2/D4/D6/D8/D10 = 48/96/144/192/240 h). Global latitude-weighted root-mean-square error (WRMSE, lower is better). Among non-oracle methods, bold marks the best result in each column; underlining marks the second best. †FuXi does not output Q850.

	Z500 ($m^2\,s^{-2}$)↓					Q850 ($10^{-3}\,kg\,kg^{-1}$)↓					U850 ($m\,s^{-1}$)↓				
Method	D2	D4	D6	D8	D10	D2	D4	D6	D8	D10	D2	D4	D6	D8	D10
Individual AI Members															
Pangu-Weather [2]	102.4	263.0	478.1	691.8	862.0	1.19	1.63	1.96	2.19	2.35	2.03	3.39	4.78	5.92	6.72
Stormer [7]	85.0	206.1	379.9	565.0	722.9	1.03	1.28	1.51	1.71	1.88	1.88	2.83	3.94	4.98	5.75
FengWu [3]	76.0	190.3	351.6	512.0	633.2	1.00	1.25	1.45	1.62	1.74	1.71	2.67	3.68	4.52	5.07
FuXi† [5]	75.9	192.7	355.4	516.3	641.0	—	—	—	—	—	1.71	2.78	3.70	4.54	5.15
GraphCast [1]	90.9	214.6	392.7	583.3	739.3	0.98	1.30	1.56	1.79	1.96	1.79	2.87	4.06	5.14	5.93
Aurora [6]	98.5	238.9	444.0	667.7	848.1	1.11	1.55	1.90	2.16	2.34	1.86	3.17	4.61	5.87	6.75
AIFS [8]	79.9	196.4	373.5	572.5	740.4	0.96	1.25	1.52	1.76	1.96	1.73	2.77	4.00	5.20	6.08
Ensemble Baselines															
Equal-weight [11]	71.6	180.8	337.0	497.8	625.3	0.95	1.22	1.44	1.61	1.74	1.64	2.58	3.57	4.44	5.04
BMA [15]	71.1	179.6	335.2	495.2	621.9	0.95	1.22	1.43	1.60	1.73	1.63	2.57	3.56	4.43	5.02
EMOS [16]	70.2	178.0	333.4	492.9	618.2	0.95	1.22	1.44	1.61	1.74	1.64	2.58	3.57	4.44	5.04
Proprietary VLMs															
Kimi-K2.5 [45]	71.1	181.0	339.1	495.4	623.4	0.93	1.21	1.42	1.60	1.73	1.64	2.58	3.59	4.43	5.04
InternVL-S1-Pro [46]	71.9	181.7	338.5	497.7	625.0	0.93	1.22	1.45	1.59	1.75	1.63	2.60	3.56	4.46	5.04
Qwen3.5-397B [44]	71.6	178.3	337.4	498.3	626.0	0.94	1.23	1.46	1.59	1.76	1.64	2.56	3.58	4.46	5.05
BlendCast-SFT (ours)	69.7	177.0	330.7	491.0	618.6	**0.92**	**1.19**	**1.40**	**1.57**	**1.70**	**1.61**	**2.53**	3.52	4.40	5.00
BlendCast (SFT+MV-GRPO)	**69.4**	**173.0**	**324.7**	**472.7**	**597.9**	0.93	**1.19**	1.41	1.58	1.71	1.62	2.54	**3.49**	**4.33**	**4.91**
Optimal (for reference) [13, 14]	68.0	169.9	314.9	465.0	582.4	0.92	1.18	1.39	1.56	1.68	1.61	2.51	3.46	4.29	4.85

Tables 1 and 2 show that BlendCast turns the broad ensemble advantage of Equal-weight into case-dependent skill. Across 30 even-day columns, BlendCast (SFT+MV-GRPO) achieves or ties the best non-oracle result in **21 columns** and reduces WRMSE by **2.85%** on average relative to Equal-weight. The gains are largest where the oracle has room to improve over uniform averaging: for example, Z500 at D10 drops from **625.3 to 597.9** against a 582.4 reference, and MSL at D8 drops from **470.4 to 455.7** against a 442.9 reference. Averaged over the two main tables, MV-GRPO closes **62.6%** of the Equal-weight-to-Optimal gap, indicating that the policy learns a useful fraction of the wait-and-see member-selection signal without access to future verification fields.

By contrast, the proprietary VLM baselines remain close to Equal-weight. Their mean absolute relative deviations from Equal-weight in Tables 1 and 2 are only 0.68% for Kimi-K2.5, 0.98% for InternVL-S1-Pro, and 0.49% for Qwen3.5-397B, so their WRMSE values mostly appear as small perturbations around the uniform ensemble rather than systematic improvements. Appendix Table 6 and Tables 7–9 show the same pattern on surface winds and odd-day leads. Appendix G.2 illustrates the mechanism in one concrete inference case: the raw proprietary VLM outputs assign visually and numerically near-average weights, whereas BlendCast emits more case-dependent allocations.

4.3 Ablations

Weather evidence. Table 3 first compares weather evidence at D5 and D10 by removing one input stream at a time. The full multimodal input is consistently better than either single-stream variant. Removing structured diagnostics increases WRMSE by roughly **2.5–4.0%**, while removing map images increases WRMSE by roughly **2.0–3.0%**. This indicates that spatial map patterns and compact ensemble diagnostics provide complementary evidence for case-dependent member selection.

Table 2: 2025 test-set results, thermal and pressure variables, even-day leads (D2/D4/D6/D8/D10 = 48/96/144/192/240 h). Global WRMSE (lower is better); among non-oracle methods, bold marks the best result and underlining marks the second best in each column.

	T850 (K) ↓					T2M (K) ↓					MSL (Pa) ↓				
Method	D2	D4	D6	D8	D10	D2	D4	D6	D8	D10	D2	D4	D6	D8	D10
Individual AI Members															
Pangu-Weather [2]	1.06	1.75	2.60	3.38	3.99	1.12	1.68	2.29	2.82	3.24	117.2	266.0	458.4	639.9	773.3
Stormer [7]	0.95	1.41	2.06	2.75	3.32	1.18	1.43	1.81	2.25	2.62	101.1	214.2	375.2	542.8	675.2
FengWu [3]	0.86	1.32	1.91	2.48	2.91	0.86	1.18	1.58	1.97	2.27	87.2	196.5	345.6	485.8	584.5
FuXi† [5]	0.86	1.36	1.92	2.50	2.97	0.87	1.24	1.60	1.98	2.30	90.1	203.4	349.3	489.4	593.3
GraphCast [1]	0.91	1.43	2.11	2.83	3.43	0.89	1.26	1.75	2.27	2.73	99.8	217.6	382.2	551.2	683.3
Aurora [6]	0.96	1.62	2.42	3.24	3.88	0.99	1.51	2.08	2.65	3.08	97.1	230.3	424.1	623.3	772.9
AIFS [8]	0.86	1.34	2.02	2.77	3.39	0.86	1.19	1.65	2.15	2.58	92.2	203.9	369.8	548.6	692.2
Ensemble Baselines															
Equal-weight [11]	0.81	1.26	1.84	2.41	2.87	0.81	1.12	1.52	1.90	2.21	81.4	185.4	329.2	470.4	574.3
BMA [15]	0.81	1.25	1.83	2.40	2.85	0.80	1.11	1.50	1.89	2.20	81.0	184.6	327.9	468.6	571.9
EMOS [16]	0.81	1.25	1.83	2.40	2.84	0.80	1.11	1.50	1.88	2.18	81.3	185.3	329.1	470.4	574.3
Proprietary VLMs															
Kimi-K2.5 [45]	0.83	1.28	1.84	2.40	2.87	0.82	1.14	1.54	1.89	2.19	81.5	185.1	331.6	468.4	572.8
InternVL-S1-Pro [46]	0.87	1.26	1.83	2.43	2.85	0.87	1.11	1.51	1.90	2.22	80.9	187.4	330.5	469.9	572.9
Qwen3.5-397B [44]	0.80	1.26	1.82	2.42	2.88	0.81	1.12	1.52	1.90	2.20	81.9	185.5	331.5	469.0	576.6
BlendCast-SFT (ours)	**0.79**	**1.22**	1.80	2.39	2.85	0.78	**1.08**	1.49	1.89	2.21	79.5	183.1	325.2	469.4	576.6
BlendCast (SFT+MV-GRPO)	0.80	1.23	**1.79**	**2.36**	**2.79**	**0.77**	1.09	**1.47**	**1.84**	**2.15**	**79.4**	**177.9**	**317.3**	**455.7**	**554.9**
Optimal (for reference) [13, 14]	0.79	1.21	1.76	2.31	2.73	0.76	1.06	1.44	1.80	2.10	78.3	175.7	310.2	442.9	539.0

Table 3: Ablation results across variables and lead times. Entries report global WRMSE at D5 and D10 (120/240 h) on the held-out test set; lower is better. Bold marks the best result in each column; underlining marks the second best. Variable units follow Tables 1–2.

	Z500		Q850		U850		T850		T2M		MSL		U10		V10	
Configuration	D5	D10	D5	D10	D5	D10	D5	D10	D5	D10	D5	D10	D5	D10	D5	D10
Weather evidence																
Images + diagnostics	**242.5**	**597.9**	1.31	1.71	3.04	**4.91**	**1.50**	**2.79**	**1.27**	**2.15**	**245.1**	**554.9**	**2.08**	**3.39**	**2.14**	**3.55**
Images only	248.7	617.7	1.36	1.75	3.12	5.03	1.54	2.87	1.30	2.21	252.5	574.0	2.15	3.52	2.20	3.66
Diagnostics only	249.2	613.9	1.35	1.75	3.11	5.01	1.54	2.86	1.30	2.19	252.0	567.3	2.14	3.47	2.20	3.64
Training design																
SFT only	250.1	618.6	**1.30**	**1.70**	**3.03**	5.00	**1.50**	2.85	1.28	2.21	251.4	576.6	2.10	3.43	2.15	3.60
RL only	259.9	637.3	1.35	1.77	3.16	5.20	1.57	2.99	1.32	2.32	261.8	594.3	2.19	3.60	2.25	3.75
No Split-Loss	257.1	630.5	1.32	1.73	3.09	5.14	1.54	2.89	1.29	2.26	256.1	593.6	2.12	3.48	2.21	3.67

Training design. The second block removes one learning component at a time. RL is most useful as a post-SFT refinement rather than as a replacement for supervised grounding. RL-only is worse than SFT-only by about **3–5%**, showing that the WRMSE reward alone is too sparse to learn stable weight prediction from scratch. No Split-Loss is also worse than SFT-only by about **1–3%**, supporting the need to smooth teacher-generated rationales and separate rationale learning from weight prediction.

4.4 Analysis

Member-count scaling and oracle headroom. We test whether adding AI NWP members creates more than a stronger equal-weight ensemble by enumerating all member subsets of size K and comparing equal-weight and retrospective oracle WRMSE. Figure 2 shows that both curves improve with more members, but the oracle improves faster, so the exploitable gap widens. For Z500, the equal-weight-to-oracle gap grows from **8.60** to **16.33** $\mathbf{m^2\,s^{-2}}$ at T+120 h as K increases from 2 to 7, and from **24.10** to **44.90** $\mathbf{m^2\,s^{-2}}$ at T+240 h. This widening headroom shows that larger AI ensembles contain complementary case-dependent skill that uniform averaging increasingly dilutes.

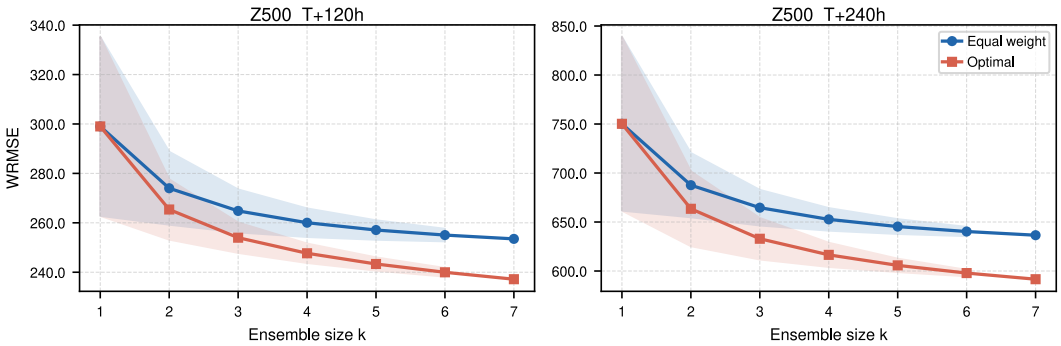


Figure 2: **Growing headroom with more members.** Blue shows equal weighting and red shows the hindsight oracle. The widening separation indicates that larger AI ensembles contain increasingly useful case-dependent complementarity that uniform averaging cannot fully exploit.

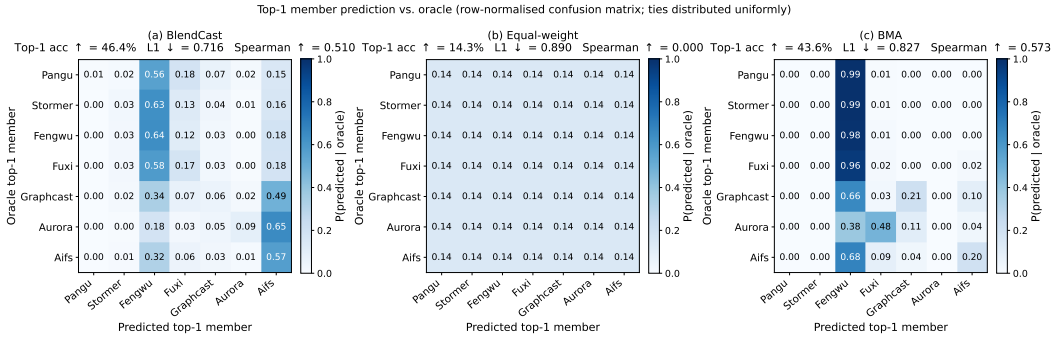


Figure 3: **Predicted member trust compared with the oracle.** Rows denote the oracle top-1 member and columns denote the predicted top-1 member. More diagonal mass means the method better recovers the oracle's case-dependent trust pattern; summary scores above each panel report top-1 accuracy, weight distance, and rank correlation.

Similarity to oracle weight distributions. We next ask whether BlendCast learns oracle-like member trust rather than only producing a blended field that scores well. For each case, lead, and variable, we compare the oracle top-1 member with the member assigned the largest predicted weight; diagonal mass in Figure 3 therefore indicates agreement with oracle trust. BlendCast identifies the oracle top-1 member in **46.4%** of cases (Spearman $\rho = 0.51$, $\ell_1 = 0.72$), far above the tie-aware equal-weight baseline of **14.3%**. BMA remains competitive as a static historical skill-ranking method (**43.6%** top-1 accuracy, $\rho = 0.57$), but its larger mean ℓ_1 distance (0.83) shows less accurate coefficient magnitude. This supports the premise that forecast-time visual state contains information beyond uniform averaging while complementing static climatological weighting. Appendix G.4 reports the same top-1 diagnostic for the three proprietary VLM baselines, whose top-1 accuracies remain near the tie-aware equal-weight baseline. Qualitative rationale and decision-output comparisons are deferred to Appendix G.2.

5 Conclusion

BlendCast reframes deterministic AI NWP ensemble weighting as a prospective reliability-forecasting problem: before verification exists, decide which frozen forecasters to trust for each variable and lead time. The method combines meteorological maps, structured ensemble diagnostics, split SFT, and MV-GRPO into an auditable policy that emits executable simplex weights rather than another forecast field. On held-out 2025 forecasts, this policy improves over equal weighting and conventional statistical post-processing while closing a substantial fraction of the retrospective oracle gap.

The experiments support three broader conclusions. First, the value of a large AI weather ensemble lies not only in a stronger average, but also in case-dependent member complementarity that uniform

weighting cannot fully use. Second, generic proprietary VLMs do not automatically discover this structure: without task-specific training, their weights remain close to the uniform simplex. Third, BlendCast's gains depend on both sides of the training recipe. SFT supplies valid tool use, meteorological language, and an initial member-selection prior, while MV-GRPO turns that prior into a forecast-quality-optimized decision policy. Together, these results suggest that VLMs can serve as real-time, interpretable forecasters of forecasters for deterministic AI weather ensembles.

Acknowledgments and Disclosure of Funding

This research was supported by funding from the Hong Kong RGC General Research Fund (152228/23E, 162161/24E, 162116/25E, 162180/25E), the National Natural Science Foundation of China (NSFC) Key Program (No. 62532005), the Collaborative Research Fund (No. C1042-23GF, No. 5097-25G), the NSFC/RGC Collaborative Research Scheme (Grant No. 62461160332 & CRS_HKUST602/24), the Research Impact Fund (No. R5011-23), and the Areas of Excellence Scheme (AoE/E-601/22-R).

The authors declare no competing interests.

References

[1] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, et al. GraphCast: Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

[2] 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.

[3] Kang Chen, Tao Han, Fenghua Ling, Junchao Gong, Lei Bai, Xinyu Wang, Jing-Jia Luo, Ben Fei, Wenlong Zhang, Xi Chen, Leiming Ma, Tianning Zhang, Rui Su, Yuanzheng Ci, Bin Li, Xiaokang Yang, and Wanli Ouyang. The operational medium-range deterministic weather forecasting can be extended beyond a 10-day lead time. *Communications Earth & Environment*, 6:518, 2025. doi: 10.1038/s43247-025-02502-y.

[4] Tao Han, Song Guo, Fenghua Ling, Kang Chen, Junchao Gong, Jing-Jia Luo, Junxia Gu, Kan Dai, Wanli Ouyang, and Lei Bai. FengWu-GHR: Learning the kilometer-scale medium-range global weather forecasting. *arXiv preprint arXiv:2402.00059*, 2024. doi: 10.48550/arXiv.2402.00059.

[5] Lei Chen, Xiaohui Zhong, Feng Zhang, et al. FuXi: A cascade machine learning forecasting system for 15-day global weather forecast. *npj Climate and Atmospheric Science*, 6:190, 2023.

[6] Cristian Bodnar, Wessel Bruinsma, et al. Aurora: A foundation model for the earth system. *Nature*, 641:1180–1187, 2025. doi: 10.1038/s41586-025-09005-y.

[7] Tung Nguyen, Rohan Shah, Hritik Bansal, Troy Arcomano, Romit Maulik, Veerabhadra Kotamarthi, Ian Foster, Sandeep Madireddy, and Aditya Grover. Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2196.

[8] Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C. A. Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, Zied Ben Bouallegue, Ana Prieto Nemesio, Peter D. Dueben, Andrew Brown, Florian Pappenberger, and Florence Rabier. AIFS: ECMWF's data-driven forecasting system. *arXiv preprint arXiv:2406.01465*, 2024. doi: 10.48550/arXiv.2406.01465.

[9] Hao Chen, Tao Han, Song Guo, Jie Zhang, Yonghan Dong, Yunlong Yu, and Lei Bai. VA-MoE: Variables-adaptive mixture of experts for incremental weather forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7915–7924, 2025.

[10] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel, et al. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 2024. arXiv:2308.15560.

[11] Martin Leutbecher and Tim N. Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008. doi: 10.1016/j.jcp.2007.02.014.

[12] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, Andras Horanyi, Joaquin Munoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.

[13] J. M. Bates and C. W. J. Granger. The combination of forecasts. *Operational Research Quarterly*, 20(4):451–468, 1969. doi: 10.1057/jors.1969.103.

[14] Allan Timmermann. Forecast combinations. In Graham Elliott, Clive W. J. Granger, and Allan Timmermann, editors, *Handbook of Economic Forecasting*, volume 1, pages 135–196. Elsevier, 2006. doi: 10.1016/S1574-0706(05)01004-9.

[15] Adrian E. Raftery, Tilmann Gneiting, Fadoua Balabdaoui, and Michael Polakowski. Using Bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, 133(5): 1155–1174, 2005.

[16] Tilmann Gneiting, Adrian E. Raftery, Anton H. Westveld III, and Tom Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, 133(5):1098–1118, 2005.

[17] Sam Allen, Christopher A. T. Ferro, and Frank Kwasniok. Regime-dependent statistical post-processing of ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 145 (725):3535–3552, 2019. doi: 10.1002/qj.3638.

[18] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637:84–90, 2025. doi: 10.1038/s41586-024-08252-9.

[19] Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13): eadk4489, 2024. doi: 10.1126/sciadv.adk4489.

[20] Simon Lang, Mihai Alexe, Mariana C. A. Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D. Dueben, Sara Hahner, Pedro Maciel, Ana Prieto-Nemesio, Cathal O'Brien, Florian Pinault, Jan Polster, Baudouin Raoult, Steffen Tietsche, and Martin Leutbecher. AIFS-CRPS: Ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *npj Artificial Intelligence*, 2(18), 2026. doi: 10.1038/s44387-026-00073-7.

[21] Simon Lang and Linus Magnusson. AIFS-ENS becomes operational. *ECMWF Newsletter*, 185: 19–23, 2025. doi: 10.21957/hg1z-pe65.

[22] Martin Andrae, Tomas Landelius, Joel Oskarsson, and Fredrik Lindsten. Continuous ensemble weather forecasting with diffusion models. In *International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=ePEZvQNFDW`.

[23] Xiaohui Zhong, Lei Chen, Hao Li, Roberto Buizza, Jun Liu, Jie Feng, Zijian Zhu, Xu Fan, Kan Dai, Jing-Jia Luo, Jie Wu, and Bo Lu. FuXi-ENS: A machine learning model for efficient and accurate ensemble weather prediction. *Science Advances*, 11(44):eadu2854, 2025. doi: 10.1126/sciadv.adu2854.

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[25] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Anton Belyi, Haotian Zhang, Karanjeet Singh, Doug Kang, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. MM1: Methods, analysis and insights from multimodal LLM pre-training. In *Computer Vision – ECCV 2024*, pages 304–323. Springer, 2024. doi: 10.1007/978-3-031-73397-0_18.

[26] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2771.

[27] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[28] DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[29] Yi Xiao, Lei Bai, Wei Xue, Hao Chen, Kun Chen, Kang Chen, Tao Han, and Wanli Ouyang. Towards a self-contained data-driven global weather forecasting framework. In *International Conference on Machine Learning*, 2024.

[30] Dmitrii Kochkov, Janni Yuval, Ian Langmore, et al. Neural general circulation models for weather and climate. *Nature*, 632:1060–1066, 2024. doi: 10.1038/s41586-024-07744-y.

[31] Anna Allen, Stratis Markou, Will Tebbutt, James Requeima, Wessel P. Bruinsma, Tom R. Andersson, Michael Herzog, Nicholas D. Lane, Matthew Chantry, J. Scott Hosking, and Richard E. Turner. End-to-end data-driven weather prediction. *Nature*, 641:1172–1179, 2025. doi: 10.1038/s41586-025-08897-0.

[32] Xiuyu Sun, Xiaohui Zhong, Xiaoze Xu, Yuanqing Huang, Hao Li, J. David Neelin, Deliang Chen, Jie Feng, Wei Han, Libo Wu, and Yuan Qi. A data-to-forecast machine learning system for global weather. *Nature Communications*, 16:6658, 2025. doi: 10.1038/s41467-025-62024-1.

[33] Joel Oskarsson, Tomas Landelius, Marc Peter Deisenroth, and Fredrik Lindsten. Probabilistic weather forecasting with hierarchical graph neural networks. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1316.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[35] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736, 2022. doi: 10.52202/068431-1723.

[36] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

[37] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[38] Chengqian Ma, Zhanxiang Hua, Alexandra Anderson-Frey, Vikram Iyer, Xin Liu, and Lianhui Qin. WeatherQA: Can multimodal language models reason about severe weather? *arXiv preprint arXiv:2406.11217*, 2024. doi: 10.48550/arXiv.2406.11217.

[39] Haobo Li, Zhaowei Wang, Jiachen Wang, Yueya Wang, Alexis Kai Hon Lau, and Huamin Qu. CLLMate: A multimodal benchmark for weather and climate events forecasting. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17536–17562, 2025. doi: 10.18653/v1/2025.emnlp-main.886.

[40] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

[41] John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 2 edition, 2011. doi: 10.1007/978-1-4614-0237-4.

[42] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Li, Jianqiang Huang, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[43] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. arXiv:2501.17161.

[44] Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[45] Moonshot AI. Kimi K2: Open agentic intelligence, 2025. Technical Report. `https://moonshotai.github.io/Kimi-K2/`.

[46] Zhe Chen et al. InternVL3: Exploring advanced training strategies for multimodal large language models. *arXiv preprint arXiv:2504.10479*, 2025.

[47] Hugging Face. TRL GRPOTrainer: Logged metrics. `https://huggingface.co/docs/trl/main/grpo_trainer`, 2026. Accessed 2026-05-06.

[48] Dieter Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt, Cologne, Germany, 1988.

Appendix Contents

A Full Evaluation Protocol.

B Metric Definitions and Oracle Gap.

C Members, Variable Availability, and Baselines; BMA and EMOS; Proprietary VLM Baselines.

D Extended Results and Ablations; Odd-Day Leads; Additional Regions; Ablation Reporting.

E Oracle Blending Details.

F Visual State Construction.

G Training Sample and Output Schema; Structured Decision Constraints; Inference Input–Output Visualization; External VLM Reliability and Weight Confusion.

H Teacher Narratives and SFT Data Construction; Oracle Weight Computation; Teacher Narrative Generation; Record Assembly; Training Coverage.

I MV-GRPO Reward Details; Lead-Time Weights.

J Training Hyperparameters and Compute.

K Discussion.

A Full Evaluation Protocol

All held-out numbers in the main paper are computed on the 2025 test period, covering all 00Z and 12Z initializations from 2025-01-01 through 2025-12-31. Each initialization is evaluated at ten daily lead times from T+24 h to T+240 h. The main reporting domain is global, with latitude weighting applied in all spatial averages. Verification uses ERA5 analyses at the corresponding valid time; ERA5 fields are used only for offline labeling, reward computation, and evaluation, never as part of the inference-time prompt. The target variables are Z500, Q850, U850, T850, T2M, U10, V10, and MSL. All methods are evaluated on the same forecast cases, leads, variables, grids, and verification fields.

Abbreviation	Full variable name	Unit
Z500	500 hPa geopotential height	$\mathrm{m^2\,s^{-2}}$
Q850	850 hPa specific humidity	$10^{-3}\,\mathrm{kg\,kg^{-1}}$
U850	850 hPa zonal wind	$\mathrm{m\,s^{-1}}$
T850	850 hPa temperature	K
T2M	2-meter temperature	K
U10	10-meter zonal wind	$\mathrm{m\,s^{-1}}$
V10	10-meter meridional wind	$\mathrm{m\,s^{-1}}$
MSL	Mean sea-level pressure	Pa

Table 4: Target variables evaluated in BlendCast.

B Metric Definitions and Oracle Gap

For a forecast field $\hat{F}_v$ and ERA5 verification field F_v, the global latitude-weighted root-mean-square error (WRMSE) is

$$\mathrm{WRMSE}_v(\hat{F}, F) = \sqrt{\frac{\sum_{x,y} c_y \left(\hat{F}_v(x,y) - F_v(x,y)\right)^2}{\sum_{x,y} c_y}}, \qquad c_y = \cos(\phi_y). \tag{11}$$

When reporting anomaly correlation coefficient (ACC), climatological means are removed from both forecast and verification fields before computing the latitude-weighted correlation. Relative improvement over equal weighting is

$$I_v(\mathrm{method}) = \frac{\mathrm{WRMSE}_{\mathrm{eq},v} - \mathrm{WRMSE}_{\mathrm{method},v}}{\mathrm{WRMSE}_{\mathrm{eq},v}}, \tag{12}$$

and the fraction of the equal-weight-to-oracle gap closed is

$$G_v(\text{method}) = \frac{\text{WRMSE}_{\text{eq},v} - \text{WRMSE}_{\text{method},v}}{\max(\text{WRMSE}_{\text{eq},v} - \text{WRMSE}_{\text{opt},v}, \epsilon)}. \tag{13}$$

The small constant ϵ prevents unstable ratios when the oracle improvement space is numerically negligible.

C Members, Variable Availability, and Baselines

The ensemble contains seven deterministic AI NWP members: Pangu-Weather, Stormer, FengWu, FuXi, GraphCast, Aurora, and AIFS. All members are used whenever they provide the target variable. If a member does not provide a variable, that member is excluded from the corresponding variable-wise simplex and the remaining weights are normalized over the available members.

Member	Z500	Q850	U850	T850	T2M	U10	V10	MSL
Pangu-Weather	yes	yes	yes	yes	yes	yes	yes	yes
Stormer	yes	yes	yes	yes	yes	yes	yes	yes
FengWu	yes	yes	yes	yes	yes	yes	yes	yes
FuXi	yes	no	yes	yes	yes	yes	yes	yes
GraphCast	yes	yes	yes	yes	yes	yes	yes	yes
Aurora	yes	yes	yes	yes	yes	yes	yes	yes
AIFS	yes	yes	yes	yes	yes	yes	yes	yes

Table 5: Variable availability used for BlendCast evaluation.

The primary ensemble baseline is equal weighting over the available members for each variable. Oracle blending is reported only as a retrospective upper bound because it uses ERA5 verification at the valid time.

C.1 Statistical Ensemble Baselines: BMA and EMOS

Both BMA and EMOS require an estimate of each member's historical error characteristics. To avoid any information leakage into the 00Z/12Z evaluation set, parameters are estimated exclusively from 06Z and 18Z initializations drawn from the same 2025 test period.

Bayesian Model Averaging (BMA). BMA assigns per-member weights that are inversely proportional to each member's historical root-mean-square error, conditioned on variable and lead time.

Parameter estimation. For each triple (v, τ, m) — variable v, lead time τ (hours), member m — we accumulate the latitude-weighted sum of squared errors (SSE) and sample count over training initializations $\mathcal{T}_{\text{train}}$:

$$\text{SSE}(v,\tau,m) = \sum_{t \in \mathcal{T}_{\text{train}}} \frac{\sum_{i,j} w_\phi(\phi_{ij}) \left(x^m_{ij}(v,\tau,t) - y_{ij}(v,\tau,t)\right)^2}{\sum_{i,j} w_\phi(\phi_{ij})}, \tag{14}$$

where $w_\phi(\phi) = \cos\phi / \langle\cos\rangle$ is the normalised cosine latitude weight and y denotes the ERA5 verification field.

Weight assignment. The per-member root-mean-square error (RMSE) and the softmax-like inverse-RMSE weight are:

$$\text{RMSE}(v,\tau,m) = \sqrt{\frac{\text{SSE}(v,\tau,m)}{|\mathcal{T}_{\text{train}}|}}, \tag{15}$$

$$w_m(v,\tau) = \frac{1/\text{RMSE}(v,\tau,m)}{\sum_{m'} 1/\text{RMSE}(v,\tau,m')}. \tag{16}$$

If no training samples are available for a member, uniform weights are used as a fallback.

BMA forecast. The blended field is a weighted sum over available members:

$$\hat{x}^{\text{BMA}}(v,\tau) = \sum_m w_m(v,\tau)\, x^m(v,\tau). \tag{17}$$

Ensemble Model Output Statistics (EMOS). Our EMOS variant performs per-member mean-bias correction before averaging with equal weights. The bias estimator is the simplest and most robust component of the classical EMOS framework [16, 15], retained here because it accounts for the dominant source of systematic error in short-to-medium range AI NWP members.

Bias estimation. For each (v, τ, m), the latitude-weighted mean bias is accumulated over training initializations:

$$\mathrm{bias}(v, \tau, m) = \frac{1}{|\mathcal{T}_{\text{train}}|} \sum_{t \in \mathcal{T}_{\text{train}}} \frac{\sum_{i,j} w_\phi(\phi_{ij}) \left(x^m_{ij}(v, \tau, t) - y_{ij}(v, \tau, t) \right)}{\sum_{i,j} w_\phi(\phi_{ij})}. \tag{18}$$

Bias-corrected blending. Each member's field is shifted by its estimated bias, and the corrected members are averaged with equal weights:

$$\tilde{x}^m(v, \tau) = x^m(v, \tau) - \mathrm{bias}(v, \tau, m), \tag{19}$$

$$\hat{x}^{\text{EMOS}}(v, \tau) = \frac{1}{M} \sum_m \tilde{x}^m(v, \tau). \tag{20}$$

Note that EMOS, as implemented here, produces a deterministic ensemble mean, not a probabilistic forecast.

C.2 Proprietary VLM Baselines

We evaluate three proprietary vision-language models as baselines: **InternVL3-S1-Pro** [46], **Kimi-K2.5** [45], and **Qwen3.5-397B** [44]. All three receive *exactly the same* multimodal input as BlendCast: per-member NWP output maps encoded as Joint Photographic Experts Group (JPEG) images (resized to at most 420 pixels wide) together with the structured text diagnostic block describing forecast spread, regional statistics, and historical skill rankings. They are prompted to emit ensemble weights via the identical tool-call schema used to supervise BlendCast-SFT, and their outputs are processed by the same shared weight-validation and repair pipeline. No model-specific prompt engineering beyond the shared system prompt is applied.

Inference protocol. Each model is served locally through an OpenAI-compatible application programming interface (API) endpoint. Requests are dispatched concurrently with eight worker threads to amortise network round-trip latency. All three models are queried at temperature $T = 0.7$ with a single sample per (initialization, lead-time) pair, matching the BlendCast-SFT evaluation setting.

Model-specific details. *InternVL3-S1-Pro* [46] is a vision-language model that natively exposes a function-calling interface. Rather than embedding the ensemble weights inside the free-form text reply, it returns them via a structured function call. The parser therefore extracts weights from the first returned JSON argument payload and reformats them into the shared schema before passing them to the weight-validation logic.

Kimi-K2.5 [45] is a multimodal reasoning model. Chain-of-thought reasoning is disabled at inference time to eliminate the substantial latency overhead of extended internal deliberation. Under this setting the model omits its chain-of-thought scratchpad and returns only the weight assignment in the final response text.

Qwen3.5-397B [44] is a large-scale vision-language model with 397 billion parameters. As with Kimi-K2.5, the thinking mode is disabled to keep inference latency comparable to the other baselines. The model operates on the same image-plus-text context and produces weights in the same structured format expected by the parser.

Parse success and failure handling. If the parser fails to extract a valid weight assignment from a model's raw output — for example, when the reply is truncated or the weight structure does not conform to the expected schema — that (initialization, lead-time) pair falls back to Equal-weight in the WRMSE evaluation. Cases where the API call itself times out or returns an error are handled by the same fallback. We report parse success separately as an interface-reliability diagnostic, but failed or unparseable calls are not removed from the forecast-skill denominator.

D Extended Results and Ablations

Table 6: Additional 2025 test-set results, surface wind components, even-day leads (D2/D4/D6/D8/D10 = 48/96/144/192/240 h). Global WRMSE (lower is better); among non-oracle methods, bold marks the best result and underlining marks the second best in each column.

	U10 ($m\,s^{-1}$) ↓					V10 ($m\,s^{-1}$) ↓				
Method	D2	D4	D6	D8	D10	D2	D4	D6	D8	D10
Individual AI Members										
Pangu-Weather [2]	1.37	2.34	3.33	4.10	4.61	1.40	2.40	3.42	4.27	4.81
Stormer [7]	1.29	1.97	2.76	3.48	3.99	1.30	2.01	2.87	3.66	4.24
FengWu [3]	1.13	1.83	2.54	3.12	3.47	1.17	1.87	2.63	3.26	3.65
FuXi† [5]	1.14	1.92	2.55	3.12	3.53	1.17	1.97	2.64	3.27	3.70
GraphCast [1]	1.21	1.98	2.83	3.58	4.10	1.24	2.02	2.92	3.74	4.32
Aurora [6]	1.25	2.19	3.22	4.08	4.66	1.30	2.27	3.35	4.29	4.92
AIFS [8]	1.14	1.91	2.81	3.64	4.24	1.18	1.96	2.91	3.82	4.48
Ensemble Baselines										
Equal-weight [11]	1.09	1.77	2.48	3.07	3.46	1.12	1.81	2.56	3.22	3.65
BMA [15]	1.08	1.76	2.47	3.06	3.45	1.11	1.81	2.55	3.21	3.64
EMOS [16]	1.09	1.77	2.48	3.07	3.46	1.12	1.81	2.56	3.22	3.65
Proprietary VLMs										
Kimi-K2.5 [45]	1.08	1.76	2.48	3.05	3.46	1.11	1.82	2.57	3.22	3.65
InternVL-S1-Pro [46]	1.08	1.79	2.47	3.05	3.48	1.13	1.83	2.54	3.23	3.65
Qwen3.5-397B [44]	1.10	1.77	2.49	3.07	3.47	1.10	1.81	2.56	3.23	3.66
BlendCast-SFT (ours)	**1.06**	**1.74**	2.43	3.03	3.43	**1.09**	**1.78**	**2.51**	3.16	3.60
BlendCast (SFT+MV-GRPO)	1.07	**1.74**	**2.41**	**2.99**	**3.39**	1.10	**1.78**	**2.51**	**3.15**	**3.55**
Optimal (for reference) [13, 14]	1.06	1.71	2.39	2.96	3.33	1.09	1.76	2.47	3.09	3.49

D.1 Odd-Day Lead Times

Tables 7–9 repeat the evaluation of the main even-day analysis for odd-day lead times (D1/D3/D5/D7/D9 = 24/72/120/168/216 h), covering all eight variables.

D.2 Regional and Secondary-Metric Scope

The main paper reports global latitude-weighted WRMSE to keep the comparison uniform across variables, lead times, and baselines. Regional WRMSE, anomaly correlation coefficient (ACC), and bias are useful follow-up diagnostics for separating phase accuracy, amplitude correction, and mean-bias reduction, but they are not used for model ranking in this submission.

D.3 Ablation Reporting

Table 3 reports the compact main-paper ablation design at D5 and D10. All rows use the same training archive, evaluation set, parser, and post-processing rules; only the named evidence stream or training component changes.

D.4 MV-GRPO Training Diagnostics

We additionally parse the MV-GRPO training log to audit whether reinforcement learning is behaving like a stable policy-improvement run rather than merely producing plausible text. Following the

Table 7: Extended results, upper-air variables, odd-day leads (D1/D3/D5/D7/D9 = 24/72/120/168/216 h). Global WRMSE (lower is better); among non-oracle methods, bold marks the best result and underlining marks the second best in each column. †FuXi does not output Q850.

	Z500 ($m^2\,s^{-2}$) ↓					Q850 ($10^{-3}\,kg\,kg^{-1}$) ↓					U850 ($m\,s^{-1}$) ↓				
Method	D1	D3	D5	D7	D9	D1	D3	D5	D7	D9	D1	D3	D5	D7	D9
Individual AI Members															
Pangu-Weather [2]	50.3	173.4	366.2	588.9	783.9	0.88	1.42	1.80	2.08	2.28	1.43	2.68	4.10	5.40	6.37
Stormer [7]	47.5	137.5	288.6	474.4	648.9	0.87	1.16	1.40	1.62	1.80	1.50	2.32	3.38	4.49	5.40
FengWu [3]	41.9	125.3	267.3	435.8	578.1	0.79	1.13	1.36	1.54	1.68	1.26	2.17	3.18	4.13	4.83
FuXi† [5]	41.9	125.6	275.7	439.9	584.1	—	—	—	—	—	1.25	2.21	3.40	4.15	4.89
GraphCast [1]	49.3	145.3	298.4	491.1	666.4	0.73	1.15	1.43	1.68	1.88	1.26	2.31	3.46	4.63	5.57
Aurora [6]	50.7	160.3	334.8	558.1	766.1	0.78	1.35	1.73	2.04	2.26	1.25	2.49	3.89	5.28	6.36
AIFS [8]	44.9	129.9	278.6	474.8	662.0	0.75	1.11	1.38	1.64	1.87	1.27	2.21	3.38	4.63	5.67
Ensemble Baselines															
Equal-weight [11]	38.3	118.8	255.3	420.3	567.1	0.73	1.10	1.34	1.53	1.68	1.19	2.09	3.09	4.04	4.78
BMA [15]	38.2	117.9	253.9	418.2	564.0	0.73	1.09	1.33	1.52	1.67	1.19	2.08	3.08	4.02	4.76
EMOS [16]	37.7	116.7	251.8	416.2	561.2	0.73	1.10	1.34	1.53	1.68	1.19	2.09	3.09	4.04	4.78
Proprietary VLMs															
Kimi-K2.5 [45]	38.2	120.3	254.7	418.2	565.8	0.72	1.10	1.36	1.54	1.69	1.17	2.09	3.08	4.02	4.78
InternVL-S1-Pro [46]	37.8	119.6	254.5	419.7	568.8	0.72	1.10	1.35	1.55	1.68	1.19	2.10	3.09	4.02	4.76
Qwen3.5-397B [44]	38.3	116.8	254.0	420.0	568.6	0.73	1.10	1.34	1.51	1.70	1.18	2.09	3.08	4.04	4.76
BlendCast-SFT (ours)	37.7	115.4	250.1	416.5	561.6	**0.70**	**1.07**	**1.30**	**1.50**	**1.64**	**1.16**	**2.05**	**3.03**	3.99	4.74
BlendCast (SFT+MV-GRPO)	**37.6**	**114.7**	**242.5**	**404.4**	**547.3**	0.71	**1.07**	1.31	**1.50**	**1.64**	1.17	**2.05**	3.04	**3.96**	**4.67**
Optimal (for reference) [13, 14]	37.1	112.1	238.9	392.8	528.4	0.70	1.06	1.29	1.48	1.63	1.16	2.04	3.00	3.90	4.61

Table 8: Extended results, thermal and pressure variables, odd-day leads (D1/D3/D5/D7/D9 = 24/72/120/168/216 h). Global WRMSE (lower is better); among non-oracle methods, bold marks the best result and underlining marks the second best in each column.

	T850 (K) ↓					T2M (K) ↓					MSL (Pa) ↓				
Method	D1	D3	D5	D7	D9	D1	D3	D5	D7	D9	D1	D3	D5	D7	D9
Individual AI Members															
Pangu-Weather [2]	0.77	1.38	2.17	3.01	3.70	0.85	1.39	1.98	2.57	3.04	65.6	183.9	359.1	555.3	713.1
Stormer [7]	0.77	1.15	1.72	2.41	3.05	1.08	1.29	1.61	2.03	2.44	68.1	149.5	291.1	461.7	615.4
FengWu [3]	0.66	1.07	1.60	2.21	2.72	0.70	1.01	1.38	1.78	2.13	52.7	134.8	268.2	420.5	540.4
FuXi† [5]	0.65	1.08	1.69	2.22	2.76	0.71	1.04	1.47	1.80	2.15	54.9	138.5	281.9	423.7	546.2
GraphCast [1]	0.66	1.15	1.75	2.47	3.15	0.71	1.06	1.49	2.01	2.51	59.9	151.2	296.3	469.8	623.0
Aurora [6]	0.66	1.27	2.01	2.84	3.58	0.71	1.24	1.79	2.37	2.88	54.1	155.1	321.5	527.3	706.6
AIFS [8]	0.65	1.08	1.66	2.40	3.10	0.72	1.01	1.41	1.91	2.37	57.3	140.0	281.8	461.7	625.0
Ensemble Baselines															
Equal-weight [11]	0.61	1.01	1.54	2.14	2.66	0.65	0.95	1.32	1.72	2.07	49.0	126.4	254.8	403.4	528.1
BMA [15]	0.61	1.01	1.53	2.13	2.64	0.64	0.94	1.30	1.70	2.05	48.7	125.7	253.8	401.9	525.9
EMOS [16]	0.61	1.01	1.53	2.12	2.64	0.65	0.94	1.30	1.70	2.04	48.9	126.3	254.7	403.4	528.1
Proprietary VLMs															
Kimi-K2.5 [45]	0.60	1.01	1.55	2.12	2.68	0.70	0.95	1.32	1.73	2.07	49.3	127.6	252.8	404.6	528.6
InternVL-S1-Pro [46]	0.60	1.03	1.56	2.14	2.68	**0.63**	1.05	1.33	1.74	2.08	49.2	125.3	255.3	404.4	526.3
Qwen3.5-397B [44]	0.62	1.01	1.53	2.12	2.65	0.72	1.02	1.33	1.73	2.05	48.5	128.9	255.6	404.4	526.1
BlendCast-SFT (ours)	**0.59**	**0.98**	**1.50**	2.11	2.64	**0.63**	**0.92**	1.28	1.70	2.07	**47.7**	123.7	251.4	401.9	529.3
BlendCast (SFT+MV-GRPO)	0.60	0.99	**1.50**	**2.08**	**2.59**	**0.63**	**0.92**	**1.27**	**1.67**	**1.99**	48.1	**121.8**	**245.1**	**390.9**	**511.9**
Optimal (for reference) [13, 14]	0.59	0.98	1.48	2.05	2.54	0.62	0.90	1.25	1.63	1.96	47.3	120.5	240.5	379.9	495.8

Table 9: Extended results, surface wind components, odd-day leads (D1/D3/D5/D7/D9 = 24/72/120/168/216 h). Global WRMSE (lower is better); among non-oracle methods, bold marks the best result and underlining marks the second best in each column.

	U10 ($\mathrm{m\,s^{-1}}$) ↓					V10 ($\mathrm{m\,s^{-1}}$) ↓				
Method	D1	D3	D5	D7	D9	D1	D3	D5	D7	D9
Individual AI Members										
Pangu-Weather [2]	0.93	1.84	2.85	3.75	4.39	0.96	1.88	2.92	3.88	4.58
Stormer [7]	1.03	1.60	2.36	3.14	3.76	1.03	1.63	2.44	3.29	3.98
FengWu [3]	0.81	1.46	2.19	2.86	3.32	0.84	1.50	2.26	2.98	3.48
FuXi† [5]	0.81	1.50	2.37	2.86	3.35	0.83	1.54	2.44	2.98	3.51
GraphCast [1]	0.85	1.58	2.41	3.23	3.87	0.88	1.61	2.47	3.36	4.07
Aurora [6]	0.82	1.70	2.71	3.69	4.41	0.85	1.76	2.81	3.85	4.66
AIFS [8]	0.82	1.50	2.36	3.25	3.96	0.84	1.54	2.43	3.39	4.19
Ensemble Baselines										
Equal-weight [11]	0.77	1.41	2.13	2.80	3.29	0.80	1.44	2.20	2.91	3.47
BMA [15]	0.77	1.40	2.13	2.79	3.28	0.79	1.44	2.19	2.90	3.45
EMOS [16]	0.77	1.41	2.13	2.80	3.29	0.80	1.44	2.20	2.91	3.47
Proprietary VLMs										
Kimi-K2.5 [45]	0.86	1.40	2.11	2.80	3.31	0.89	1.46	2.21	2.93	3.48
InternVL-S1-Pro [46]	0.78	<u>1.39</u>	2.13	2.82	3.31	0.86	<u>1.43</u>	2.21	2.89	3.49
Qwen3.5-397B [44]	<u>0.76</u>	1.42	2.13	2.80	3.28	**0.77**	1.45	2.18	2.90	3.47
BlendCast-SFT (ours)	**0.75**	**1.38**	<u>2.10</u>	<u>2.76</u>	<u>3.26</u>	<u>0.78</u>	**1.42**	<u>2.15</u>	<u>2.87</u>	<u>3.42</u>
BlendCast (SFT+MV-GRPO)	<u>0.76</u>	**1.38**	**2.08**	**2.74**	**3.21**	<u>0.78</u>	**1.42**	**2.14**	**2.83**	**3.38**
Optimal (for reference) [13, 14]	0.75	1.37	2.06	2.70	3.17	0.77	1.40	2.12	2.80	3.32

GRPO formulation of group-relative reward advantages [27], and common GRPO trainer diagnostics such as reward, reward standard deviation, generation length, KL, entropy, and clipping statistics [47], we separate three questions: whether the meteorological reward is improving, whether group-level reward contrast remains informative, and where the computational cost of each update is spent. The current log records reward components, within-group reward standard deviation, generation length, per-step timing, target variable, and lead time. It does not record KL, entropy, or clipping statistics, so those remain missing diagnostics for subsequent RL runs.

Several early-run patterns are visible. First, the total reward is dominated by the WRMSE reward, while the format reward is already stable; this indicates that the policy is mostly being shaped by meteorological skill rather than by parser compliance. Second, the reward is still noisy and the within-group reward standard deviation remains large; the first 123 updates are therefore an early training diagnostic rather than a convergence result. Third, rollout is the primary training bottleneck, accounting for roughly 74% of the logged step time, whereas reward computation is below 1% of the step time. Fourth, generated length stays close to 1.8k tokens without a visible length explosion in this segment. Finally, the variable-level rewards are heterogeneous: temperature and wind-related targets produce stronger positive reward on average, while specific humidity at 850 hPa (Q850) and mean sea-level pressure (MSL) are much smaller in scale. This motivates logging normalized per-variable advantages alongside the raw rewards, as well as KL, entropy, and clipping statistics to separate useful policy movement from excessive drift.

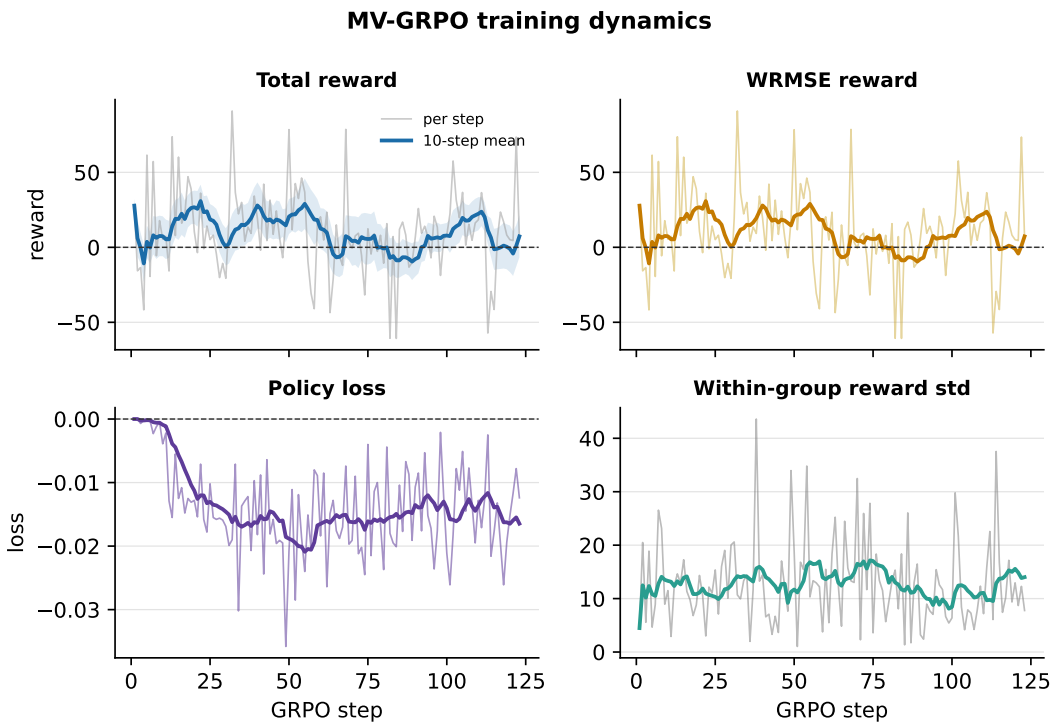


Figure 4: **MV-GRPO reward and optimization dynamics.** Thin lines show per-update values and thick lines show a 10-update moving average. The shaded band in the reward panel indicates the logged within-group reward standard deviation.

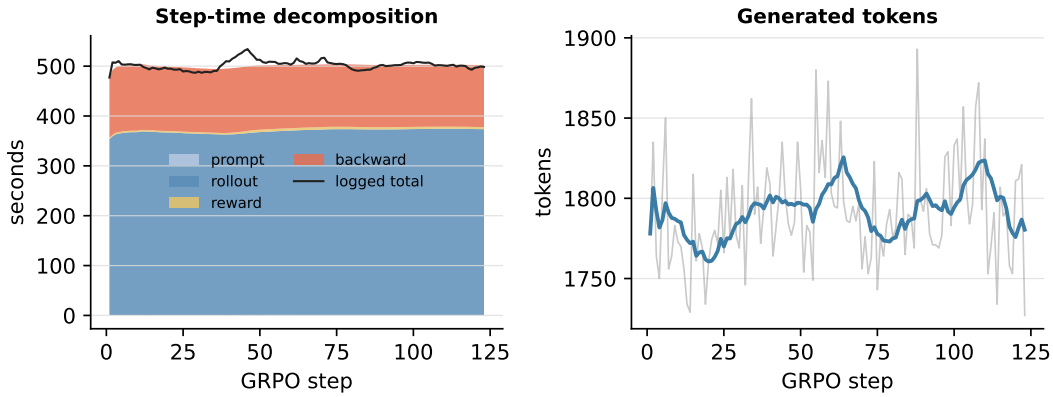


Figure 5: **MV-GRPO throughput diagnostics.** Rollout dominates the per-update wall-clock cost, while reward evaluation is a small part of each step. The generated-token curve checks whether policy updates are accompanied by length drift.

E Oracle Blending Details

For each forecast case, lead, and target variable, the retrospective oracle solves the simplex-constrained least-squares blending problem in Eq. (2). This optimization is performed independently for each variable because the oracle label is intended to describe variable-wise member skill rather than a joint multivariate state estimator. The feasible set is compact, so at least one solution exists. If multiple minimizers occur, any minimizer has the same oracle WRMSE and is equivalent for the reward normalizer; for SFT labels we use a deterministic solver tolerance and tie-breaking rule to keep labels reproducible. The implementation optimizes only over members that provide the target variable. Members without that variable are assigned zero weight for that row of the simplex; if only

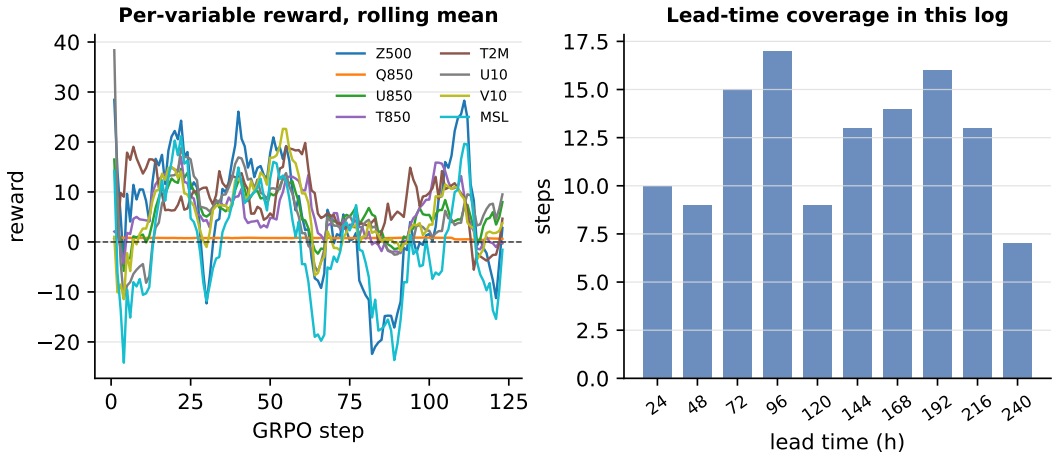


Figure 6: **Variable-wise reward signal and lead-time coverage.** Per-variable reward curves reveal which meteorological targets currently provide the strongest or weakest RL signal, while the lead histogram checks the coverage of the partial training log.

one member is available, the solution is the corresponding one-hot vector. If ERA5 verification is unavailable for a variable, the label falls back to uniform weights over the available members.

Numerically, we use sequential least-squares programming (SLSQP) [48] with three deterministic restarts: the first initialization is the equal-weight vector, and the remaining initializations are Dirichlet draws from a fixed per-variable random seed. The solver uses tolerance $\mathrm{ftol} = 10^{-9}$ and a maximum of 300 iterations. After convergence, weights below 0.03 are treated as numerical or interpretation-level noise, set to zero, and the remaining coefficients are renormalized to sum to one. This sparsification is applied to the SFT weight labels and reported oracle-weight files to keep the decisions interpretable; the RL reward normalizer uses the same simplex-constrained objective directly to obtain the oracle WRMSE denominator. The oracle is never used at inference time and is interpreted as a wait-and-see upper bound, not as an operational baseline.

F Visual State Construction

Table 10 lists the three meteorological map types that constitute the image component $\mathbf{I}$. For each variable, BlendCast renders the ensemble mean, ensemble spread, and one per-member anomaly map for every available member. The mean map gives the synoptic background, the spread map localizes disagreement, and the anomaly maps show how each member departs from the ensemble consensus. The image order is fixed across all samples so that the language model receives a stable visual schema. All maps are computed from member forecasts alone and therefore remain ERA5-free.

Table 10: Meteorological map types in BlendCast's visual state $\mathbf{I}$. All three types are computable from member forecasts alone.

Map type	**Role in skill anticipation**
Ensemble mean	Consensus synoptic state; baseline for reading member deviations
Ensemble spread	Spatial uncertainty; high-spread regions signal low ensemble confidence
Per-model anomaly	Signed departure from ensemble mean; exposes spatially coherent member-specific biases or complementary signals

G Training Sample and Output Schema

Listing 1 shows a fully annotated BlendCast-formatted sample (init 2025-06-02 00Z, lead $+24\,\mathrm{h}$). All numerical values are from the forecast archive. The assistant output contains two parts: a natural-language meteorological rationale and a structured weight object. The weight object maps each target variable to non-negative member weights that sum to one over available members.

Listing 1: Complete annotated BlendCast-formatted sample (init 2025-06-02 00Z, lead $+24$ h, valid 2025-06-03 00Z). All numerical values are from the forecast archive.

```
{
  // BlendCast SFT sample -- init: 2025-06-02 00Z  |  lead: +24h  |  valid: 2025-06-03 00Z

  "messages": [

    // ---- (1) SYSTEM ------------------------------------------------------------
    { "role": "system",
      "content": "You are a senior medium-range weather forecasting expert. Assign
        per-variable weights w_{v,m} in [0,1] summing to 1 for each variable, by
        reading ensemble mean, spread, and per-model anomaly maps." },

    // ---- (2) USER: observable state  s = (I, x) --------------------------------
    { "role": "user",
      "content": {

        // I: visual maps -- |V| x (2 + |M_v|) = 71 maps total
        "images": [
          // z500  500 hPa geopotential height m2/s2  (9 maps: mean + spread + 7 anomalies)
          "z500_emean.png",          "z500_spread.png",
          "z500_aifs_anom.png",      "z500_aurora_anom.png",     "z500_fengwu_anom.png",
          "z500_fuxi_anom.png",      "z500_graphcast_anom.png", "z500_pangu_anom.png",
          "z500_stormer_anom.png",
          // q850  850 hPa specific humidity g/kg  (8 maps: fuxi is unavailable)
          "q850_emean.png",          "q850_spread.png",
          "q850_aifs_anom.png",      "q850_aurora_anom.png",     "q850_fengwu_anom.png",
          "q850_graphcast_anom.png", "q850_pangu_anom.png",     "q850_stormer_anom.png",
          // u850 (m/s), t850 (K), t2m (K), u10 (m/s), v10 (m/s)  -- 9 maps each
          "u850_emean.png", "u850_spread.png", /* ... 7 anomaly maps ... */
          /* ... t850, t2m, u10, v10 follow the same layout ... */
          // msl  mean sea-level pressure Pa  (9 maps)
          "msl_emean.png",           "msl_spread.png",
          "msl_aifs_anom.png",       "msl_aurora_anom.png",      "msl_fengwu_anom.png",
          "msl_fuxi_anom.png",       "msl_graphcast_anom.png",  "msl_pangu_anom.png",
          "msl_stormer_anom.png"
        ],

        // x: ERA5-free structured descriptor -- see Eq.(state_x)
        "text": {
          "init_time": "2025-06-02T00Z",  "lead_h": 24,
          "valid_time": "2025-06-03T00Z", "season": "JJA",  // June-July-August
          "models":      ["aifs","aurora","fengwu","fuxi","graphcast","pangu","stormer"],
          "target_vars": ["z500","q850","u850","t850","t2m","u10","v10","msl"],

          // per-variable regional diagnostics  (mu_{v,r}, sigma_{v,r}, delta_{m,v,r})
          "diagnostics": {

            "z500": {  // m2/s2
              "Global": { "mean": 55832.3, "spread": 21.8,
                "dev": {"aifs":6.8,"aurora":-14.0,"fengwu":-2.6,
                        "fuxi":6.3,"graphcast":1.7,"pangu":1.5,"stormer":0.4} },
              "NH":     { "mean": 55975.8, "spread": 24.3,
                "dev": {"aifs":3.9,"aurora":-5.7,"fengwu":-9.0,
                        "fuxi":-0.2,"graphcast":5.8,"pangu":6.1,"stormer":-0.8} },
              "EU":     { "mean": 55246.6, "spread": 20.5,
                "dev": {"aifs":8.8,"aurora":2.8,"fengwu":-20.7,
                        "fuxi":2.9,"graphcast":3.0,"pangu":2.8,"stormer":0.3} },
              "EA":     { "mean": 55441.0, "spread": 30.1,
                "dev": {"aifs":34.6,"aurora":-33.3,"fengwu":-7.6,
                        "fuxi":-15.4,"graphcast":11.7,"pangu":14.6,"stormer":-4.6} }
            },

            /* q850 (g/kg), u850 (m/s), t850 (K), t2m (K), u10 (m/s), v10 (m/s)
               -- same four-region schema as z500 above */

            "msl": {   // Pa
              "Global": { "mean": 101138.9, "spread": 26.6,
                "dev": {"aifs":4.8,"aurora":-5.2,"fengwu":-3.9,
                        "fuxi":-4.6,"graphcast":-1.8,"pangu":11.5,"stormer":-0.8} },
              "NH":     { "mean": 101313.1, "spread": 31.2,
                "dev": {"aifs":2.4,"aurora":-3.0,"fengwu":-6.2,
                        "fuxi":-11.4,"graphcast":8.4,"pangu":9.5,"stormer":0.3} },
              "EU":     { "mean": 101096.1, "spread": 25.6,
                "dev": {"aifs":11.1,"aurora":-0.6,"fengwu":-15.9,
                        "fuxi":-5.3,"graphcast":9.5,"pangu":2.6,"stormer":-1.3} },
              "EA":     { "mean": 100613.7, "spread": 40.5,
                "dev": {"aifs":38.6,"aurora":-36.7,"fengwu":-4.5,
```

```
                        "fuxi":-31.3,"graphcast":25.3,"pangu":22.3,"stormer":-13.7} }
            }
          }
        }
      }
    },

    // ---- (3) ASSISTANT: trajectory  (r_theta, w_theta) ------------------------
    { "role": "assistant",
      "content": {

        // r_theta: chain-of-thought reasoning (visible only during training)
        "think": "Lead +24h. Spread tight on t850/t2m -> high predictability. aifs shows
          smooth synoptic anomalies -> top weight on t850 (0.495) and t2m (0.387). pangu
          deviation std on t850 is ~6x spread -> zero weight. For z500, fuxi has spatially
          coherent large-scale dipoles (North Pacific -15.4, North Atlantic +12 m2/s2),
          justifying fuxi:0.394. graphcast resolves Intertropical Convergence Zone
          (ITCZ) moisture filaments for q850.",

        // w_theta in Delta^|M| for each variable  (each row sums to 1.0)
        "weights": {
          "z500": {"fuxi":0.3936,"aifs":0.2764,"fengwu":0.1972,"aurora":0.0949,"graphcast":0.0380,"
               pangu":0.0,"stormer":0.0},
          "q850": {"aifs":0.4206,"graphcast":0.3861,"aurora":0.1932,"fengwu":0.0,"fuxi":0.0,"pangu"
               :0.0,"stormer":0.0},
          "u850": {"aifs":0.3181,"aurora":0.2737,"graphcast":0.2038,"fuxi":0.1052,"fengwu":0.0991,"
               pangu":0.0,"stormer":0.0},
          "t850": {"aifs":0.4952,"aurora":0.2435,"graphcast":0.1303,"fuxi":0.0684,"fengwu":0.0627,"
               pangu":0.0,"stormer":0.0},
          "t2m":  {"aifs":0.3874,"graphcast":0.2189,"aurora":0.1966,"fuxi":0.1511,"fengwu":0.0460,"
               pangu":0.0,"stormer":0.0},
          "u10":  {"aifs":0.4130,"aurora":0.2669,"fengwu":0.1117,"graphcast":0.1066,"fuxi":0.1018,"
               pangu":0.0,"stormer":0.0},
          "v10":  {"aifs":0.4526,"aurora":0.2090,"fuxi":0.1633,"fengwu":0.0889,"graphcast":0.0861,"
               pangu":0.0,"stormer":0.0},
          "msl":  {"aifs":0.3836,"aurora":0.3012,"fuxi":0.1544,"graphcast":0.1024,"stormer":0.0584,
               "fengwu":0.0,"pangu":0.0}
        }
      }
    }
  ],

  // meta: oracle supervision  (offline only -- never visible at inference time)
  "meta": {
    // w*_v: wait-and-see optimal weights solved via QP with ERA5  (identical to
    //       assistant weights in SFT samples by construction; diverges under GRPO)
    "oracle_weights": {
      "z500": {"fuxi":0.3936,"aifs":0.2764,"fengwu":0.1972,"aurora":0.0949,"graphcast":0.0380,"
           pangu":0.0,"stormer":0.0},
      "q850": {"aifs":0.4206,"graphcast":0.3861,"aurora":0.1932,"fengwu":0.0,"fuxi":0.0,"pangu"
           :0.0,"stormer":0.0},
      "u850": {"aifs":0.3181,"aurora":0.2737,"graphcast":0.2038,"fuxi":0.1052,"fengwu":0.0991,"
           pangu":0.0,"stormer":0.0},
      "t850": {"aifs":0.4952,"aurora":0.2435,"graphcast":0.1303,"fuxi":0.0684,"fengwu":0.0627,"
           pangu":0.0,"stormer":0.0},
      "t2m":  {"aifs":0.3874,"graphcast":0.2189,"aurora":0.1966,"fuxi":0.1511,"fengwu":0.0460,"
           pangu":0.0,"stormer":0.0},
      "u10":  {"aifs":0.4130,"aurora":0.2669,"fengwu":0.1117,"graphcast":0.1066,"fuxi":0.1018,"
           pangu":0.0,"stormer":0.0},
      "v10":  {"aifs":0.4526,"aurora":0.2090,"fuxi":0.1633,"fengwu":0.0889,"graphcast":0.0861,"
           pangu":0.0,"stormer":0.0},
      "msl":  {"aifs":0.3836,"aurora":0.3012,"fuxi":0.1544,"graphcast":0.1024,"stormer":0.0584,"
           fengwu":0.0,"pangu":0.0}
    }
  }
}
```

G.1 Structured Decision Constraints

BlendCast outputs a separate member-weight vector for each target variable. Each vector must cover all available members, contain non-negative entries, and sum to one. The same constraints are used during supervised training, reinforcement learning, and held-out evaluation so that every model decision corresponds to a realizable ensemble forecast. Parser failures, missing variables, negative weights, NaN values, and rows whose sums fall outside the accepted tolerance are marked invalid for the decision-validity reward and excluded or repaired according to the same rule for all methods.

G.2 Inference Input–Output Visualization

We visualize several held-out-style inference traces drawn from the real-forecast SFT corpus, together with the corresponding raw model responses archived during inference. The first trace, initialization 2025-01-08 00Z at lead +192 h, shows the full inference input. For three case–lead pairs, we then compare BlendCast-SFT+GRPO against the SFT checkpoint (BlendCast), and against external VLMs. Each comparison panel contains the complete raw text returned by each model and the corresponding per-variable weights parsed from its tool call.

System prompt embedded in the inference input

```
You are a senior medium-range weather forecasting expert responsible for improving numerical weather prediction
      accuracy through multi-model ensemble methods.

You have forecasts from several AI weather prediction models (the exact set varies by case).
Your goal is to assign per-variable ensemble weights so that the weighted ensemble forecast achieves lower WRMSE than a
       simple equal-weight average.

[Meteorological Expert Knowledge]
1. Spread-skill relationship (evidence-based):
   - The ensemble spread is an estimator of forecast uncertainty: larger spread indicates lower predictability.
   - When spread is anomalously large, the ensemble mean is more reliable than any individual member -- distribute
      weights more evenly.
   - When spread is anomalously small (models highly consistent), it is safer to concentrate weight on the apparently
      best member, but do so cautiously as overconfidence increases risk.
   - Spread-skill correlations are typically below 0.6; spread is most informative at the extremes.

2. Lead-time effects:
   - Short range (<=3 days): individual model differences matter; spread is typically small; weight toward the member
      with the smallest recent deviation from the ensemble mean.
   - Medium range (4--10 days): ensemble mean consistently outperforms any single member by smoothing less predictable
      features while preserving the predictable signal.
   - Long range (>10 days): all members lose individual skill; revert to near-equal weights; the ensemble mean is the
      best available forecast.

3. Ensemble strategy:
   - Weights for each variable must be non-negative and must sum exactly to 1.0 (strict constraint -- no exceptions).
   - Each variable is weighted independently; the weight vector for one variable need not equal that for another.
   - Do not assign all weight to a single model; keep at least 2 non-zero-weight models to preserve ensemble diversity.
   - A multi-model ensemble that averages across diverse models often rivals the best individual model.
   - Outlier models (deviation > 1x spread) should be down-weighted but not necessarily zeroed out.

[Tool Usage Guidelines]
You have exactly one action: call finalize_ensemble with per_var_weights.

The variables to weight are listed in the current forecast state (user prompt). You must provide weights for ALL 8
      variables: z500, q850, u850, t850, t2m, u10, v10, msl.

per_var_weights MUST:
  - Include every variable listed in the current forecast state
  - For each variable, list every available model with its weight (use 0 for excluded models)
  - Weights per variable must sum to exactly 1.0

Required output format (JSON inside <tool_call> tags):
<tool_call>
{"name": "finalize_ensemble", "arguments": {"per_var_weights": {
  "z500": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...},
  "q850": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...},
  "u850": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...},
  "t850": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...},
  "t2m": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...},
  "u10": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...},
  "v10": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...},
  "msl": {"model_a": 0.6000, "model_b": 0.4000, ..., "model_n": ...}
}}}
</tool_call>
Note: model_a/model_b above are examples. Replace them with the exact model names listed in the current forecast state
      (user prompt).

[Response Format (must be followed strictly)]
<think>
[Meteorological reasoning, <=200 words. Cover: ensemble spread pattern, model agreement status, rationale for per-
      variable weight choices]
</think>
<tool_call>
{"name": "finalize_ensemble", "arguments": {"per_var_weights": { ...all 8 variables... }}}
</tool_call>
```

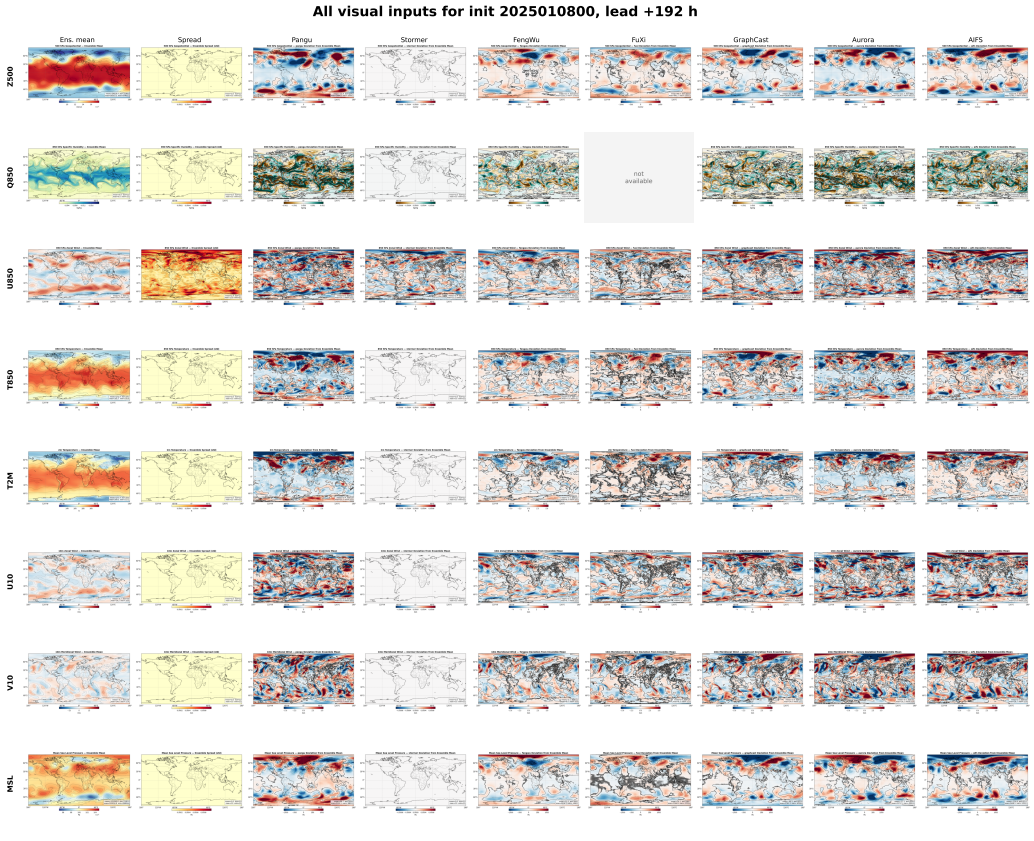


Figure 7: **Complete visual input grid for one inference case.** The grid contains ensemble mean maps, ensemble spread maps, and anomaly maps for the seven decision members across the eight target variables. This example corresponds to initialization 2025-01-08 00Z and lead +192 h.

Structured ensemble diagnostic input embedded in the user prompt

```
Current forecast state:

=== Forecast Task ===
Init time:    2025-01-08 00:00 UTC  (init_time: 2025010800)
Lead time:    +192h (Long-range (>5d))
Valid time:   2025-01-16 00:00 UTC
Season:       Winter (DJF)
Models:       pangu, stormer, fengwu, fuxi, graphcast, aurora, aifs (7 total)
Target vars:  z500, q850, u850, t850, t2m, u10, v10, msl  (assign weights for ALL 8 variables)

=== Ensemble Diagnostic Statistics (observable, ERA5-free) ===

[z500]  unit: m2/s2
  Region        Ens. Mean   Spread (std)   Model Deviations (vs. ens. mean)
  Global         +55400.6          251.4    aifs:+44.4  aurora:-79.1  fengwu:+38.3  fuxi:+67.7  graphcast:-13.9
      neuralgcm:+27.9  pangu:-87.0  stormer:+1.6
  NH             +53993.1          403.0    aifs:-8.9  aurora:-51.6  fengwu:+54.3  fuxi:+74.7  graphcast:-2.3
      neuralgcm:+75.3  pangu:-104.9  stormer:-36.6
  EU             +54784.6          645.2    aifs:+288.6  aurora:-267.1  fengwu:-397.7  fuxi:+240.6  graphcast:-218.9
      neuralgcm:+435.1  pangu:+26.7  stormer:-107.2
  EA             +53144.9          416.2    aifs:-73.8  aurora:+126.2  fengwu:-106.9  fuxi:+28.7  graphcast:-251.6
      neuralgcm:-111.9  pangu:+401.1  stormer:-11.9
  WARNING Outlier warning: pangu global deviation -87.0m2/s2 (0.3x spread)

[u850]  unit: m/s
  Region        Ens. Mean   Spread (std)   Model Deviations (vs. ens. mean)
  Global             +1.1            2.3    aifs:+0.1  aurora:+0.1  fengwu:-0.1  fuxi:-0.0  graphcast:+0.1  neuralgcm
      :+0.1  pangu:+0.1  stormer:-0.2
  NH                 +3.1            3.2    aifs:+0.3  aurora:+0.2  fengwu:-0.3  fuxi:-0.0  graphcast:+0.0  neuralgcm
      :+0.0  pangu:-0.1  stormer:-0.2
  EU                 +4.5            4.2    aifs:-0.9  aurora:+0.3  fengwu:+1.3  fuxi:-0.0  graphcast:+1.4  neuralgcm
      :-1.0  pangu:-1.5  stormer:+0.5
  EA                 +5.5            2.5    aifs:+0.9  aurora:-0.1  fengwu:-0.2  fuxi:-0.2  graphcast:+0.3  neuralgcm
      :+0.6  pangu:-0.6  stormer:-0.7
  WARNING Outlier warning: stormer global deviation -0.2m/s (0.1x spread)

[t850]  unit: K
  Region        Ens. Mean   Spread (std)   Model Deviations (vs. ens. mean)
  Global           +280.2            1.3    aifs:+0.2  aurora:-0.5  fengwu:+0.1  fuxi:+0.2  graphcast:+0.0  neuralgcm
      :+0.4  pangu:-0.5  stormer:-0.0
  NH               +271.3            2.0    aifs:+0.1  aurora:-0.3  fengwu:-0.1  fuxi:+0.3  graphcast:-0.1  neuralgcm
      :+0.5  pangu:-0.4  stormer:-0.1
  EU               +274.5            2.4    aifs:+0.7  aurora:-1.4  fengwu:-0.6  fuxi:+1.0  graphcast:-0.7  neuralgcm
      :+1.0  pangu:-0.3  stormer:+0.3
  EA               +263.9            2.1    aifs:+0.2  aurora:-1.3  fengwu:-0.7  fuxi:-0.1  graphcast:+0.3  neuralgcm
      :+0.5  pangu:+1.1  stormer:-0.0
  WARNING Outlier warning: pangu global deviation -0.5K (0.4x spread)

=== Model Consistency Summary ===
  High-spread variables: u850(201.0%) -> ensemble mean likely more robust than any single model
  Low-spread variables: z500(0.5%), t850(0.5%) -> models largely agree

=== Current Ensemble State ===
Current weights: equal (1/7 = 0.1429)  Steps executed: 0/4
```

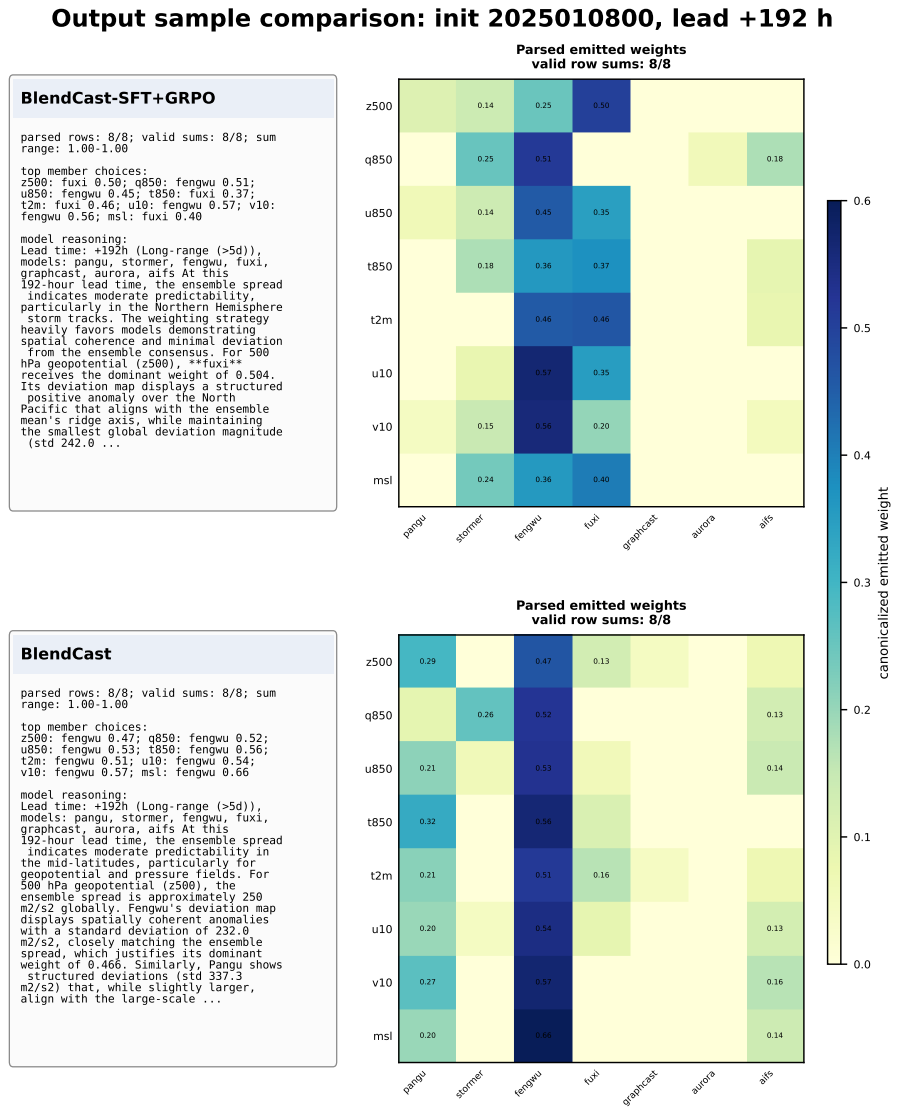


Figure 8: **Output sample comparison for BlendCast-SFT+GRPO and BlendCast.** The case is initialization 2025-01-08 00Z, lead +192 h. The two model outputs are stacked vertically; each row shows the reasoning excerpt and parsed per-variable weights. Full raw outputs are provided as text artifacts.

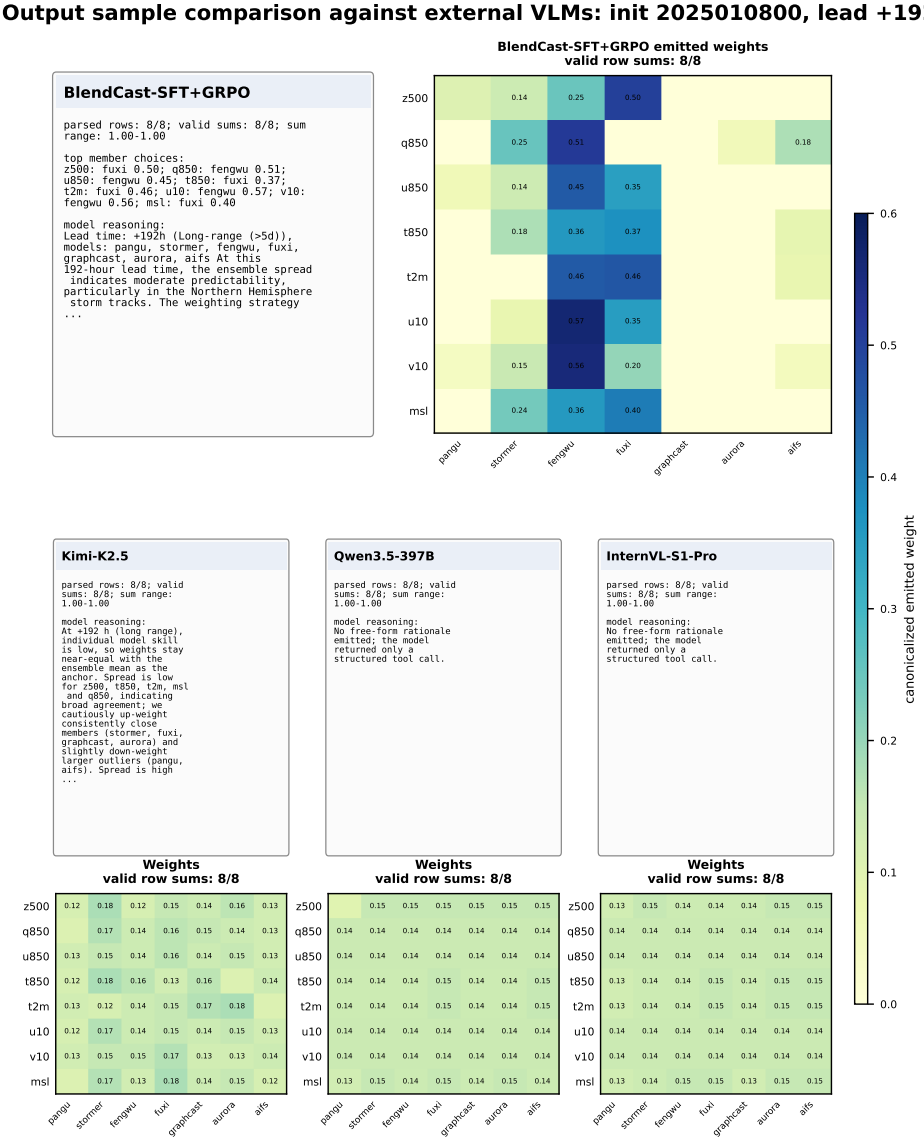


Figure 9: **Output sample comparison for BlendCast-SFT+GRPO and external VLMs.** The case is initialization 2025-01-08 00Z, lead +192 h. BlendCast-SFT+GRPO is shown in the top full-width row; Kimi-K2.5, Qwen3.5-397B, and InternVL-S1-Pro are shown in the three bottom columns.

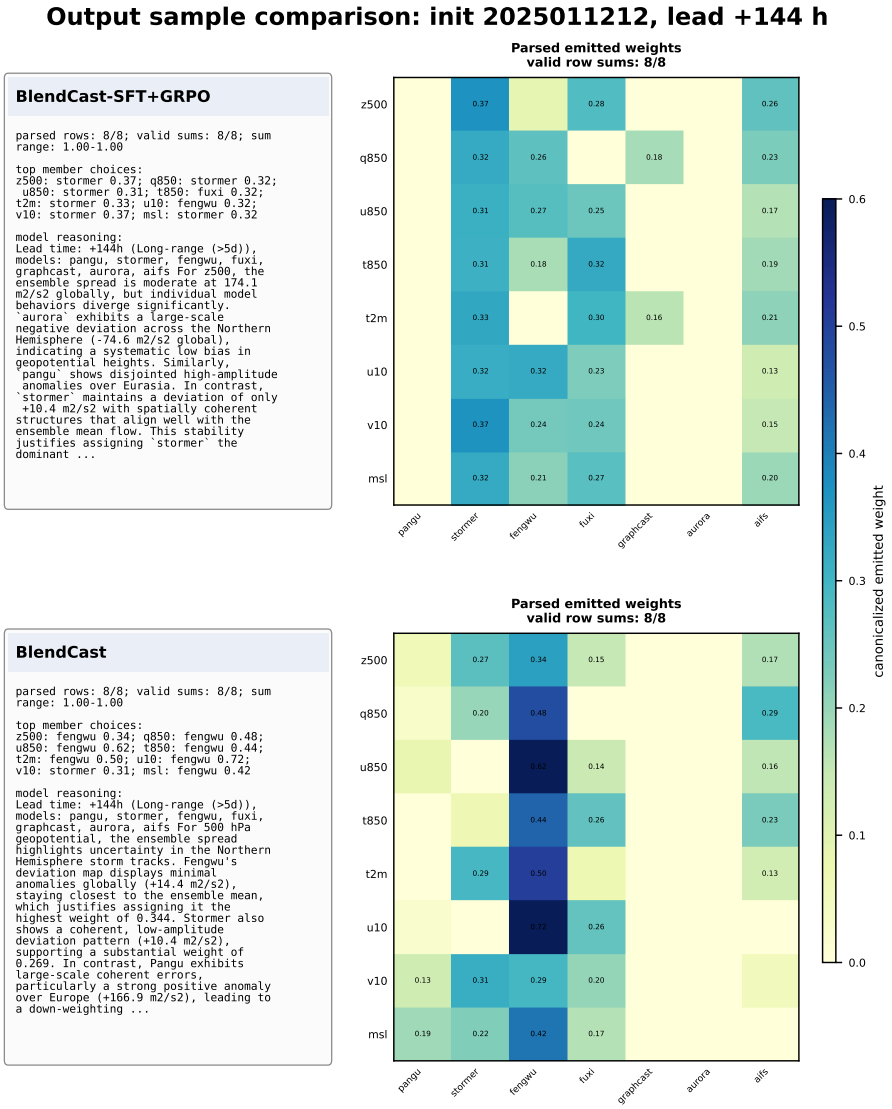
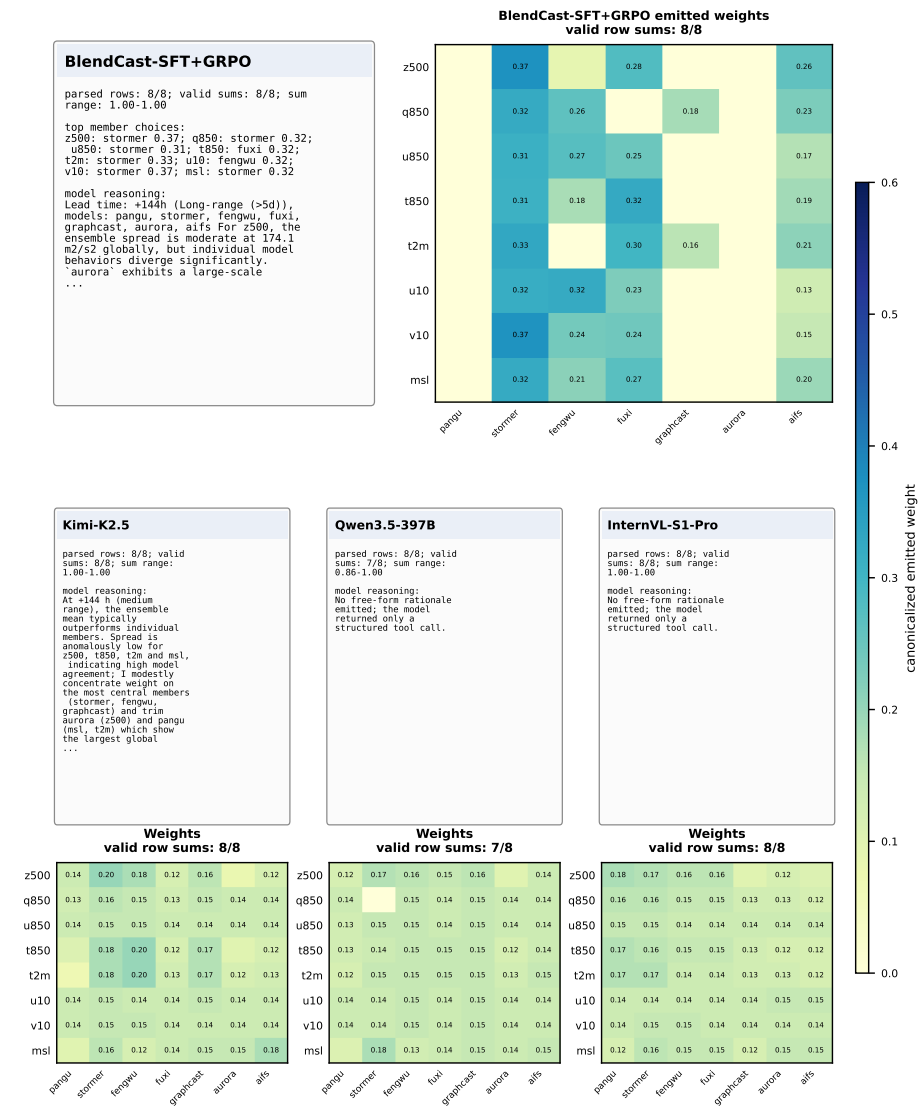


Figure 10: **Additional BlendCast-SFT+GRPO versus BlendCast output sample.** The case is initialization 2025-01-12 12Z, lead +144 h.



Top: BlendCast-SFT+GRPO output sample. Bottom: external VLM reasoning excerpts with their parsed weight decisions. Full raw outputs are saved as text artifacts.

Figure 11: **Additional BlendCast-SFT+GRPO versus external VLM output sample.** The case is initialization 2025-01-12 12Z, lead +144 h.

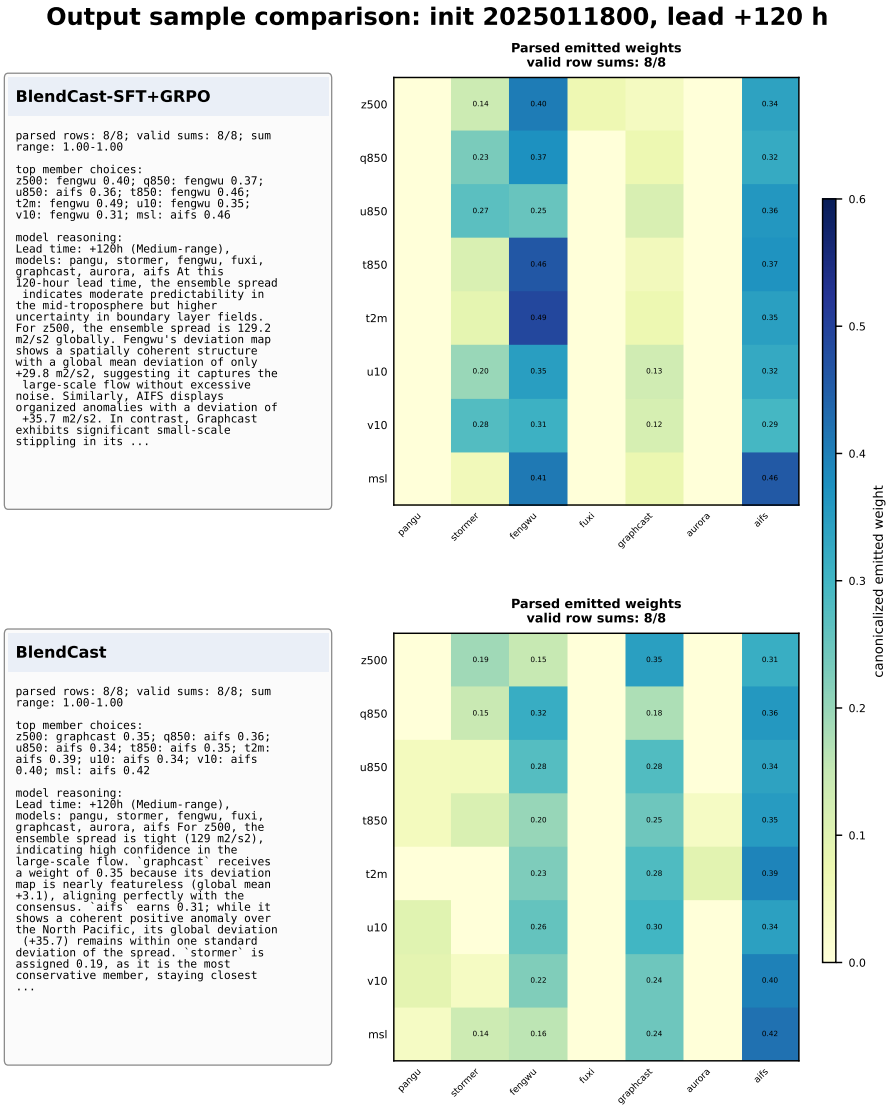
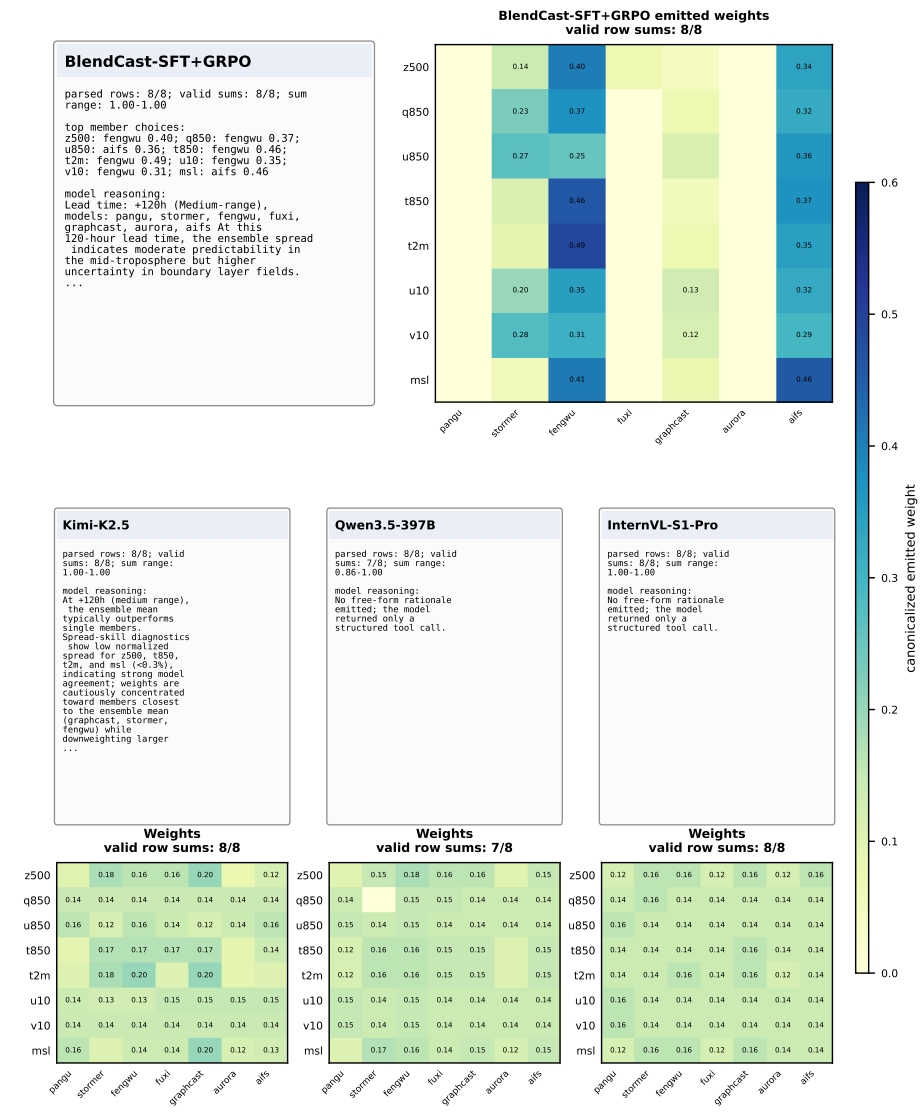


Figure 12: **Additional BlendCast-SFT+GRPO versus BlendCast output sample.** The case is initialization 2025-01-18 00Z, lead +120 h.



Top: BlendCast-SFT+GRPO output sample. Bottom: external VLM reasoning excerpts with their parsed weight decisions. Full raw outputs are saved as text artifacts.

Figure 13: **Additional BlendCast-SFT+GRPO versus external VLM output sample.** The case is initialization 2025-01-18 00Z, lead +120 h.

G.3 External VLM Output Reliability

The external VLM baselines are prompted with the same structured decision interface as BlendCast, but their raw responses are less reliable as executable ensemble decisions. The archived raw model responses contain only the lead time, sample identifier, and visible model text; they therefore record visible response failures and format failures, but do not separately identify transport-level API failures. We count an output as usable when a `per_var_weights` object can be parsed, all eight target-variable rows are present, and every row sums to one within a tolerance of 0.03. For the WRMSE tables, unusable responses fall back to Equal-weight for the same case; the counts below therefore describe response-format reliability rather than a change in the evaluation sample.

Model	Files	Records	Parsed	Complete vars	Usable rows
Kimi-K2.5	34	405	294 (72.6%)	294 (72.6%)	292 (72.1%)
Qwen3.5-397B	262	1950	1917 (98.3%)	1917 (98.3%)	1870 (95.9%)
InternVL-S1-Pro	36	374	364 (97.3%)	364 (97.3%)	328 (87.7%)

Table 11: **Response-format reliability of external VLM baselines in the archived raw responses.** "Parsed" means that a `per_var_weights` object could be recovered from `raw_text`. "Complete vars" additionally requires all eight target variables to be present. "Usable rows" requires all eight variable rows to pass the row-sum tolerance used by evaluation.

Kimi-K2.5 shows the largest format-failure rate: 111 of 405 archived responses produce free-form meteorological reasoning but no recoverable `per_var_weights` object. Qwen3.5-397B is usually parseable, but some responses contain truncated or pseudo tool calls, and 47 parsed records have at least one variable row whose weights do not sum within tolerance. InternVL-S1-Pro generally emits a recoverable weight dictionary, sometimes using a noncanonical schema such as a `parameters` field rather than the requested `arguments` field, but 36 parsed records still fail the all-row normalization check. These failures are not counted as missing forecast cases; they are a practical interface reliability issue that determines when the evaluation uses the Equal-weight fallback instead of an executable VLM weight assignment.

G.4 External VLM Weight Confusion

Figure 14 repeats the oracle top-1 member confusion diagnostic for the three proprietary VLM baselines, using the parsed weight artifacts generated from the archived responses. The quantitative pattern is sharp: InternVL-S1-Pro, Kimi-K2.5, and Qwen3.5-397B reach only 15.9%, 15.1%, and 18.5% tie-aware top-1 accuracy, respectively, compared with 14.3% for equal weighting and 46.4% for BlendCast. Their mean ℓ_1 distances are also high (0.868, 0.870, and 0.862), and their rank correlations remain weak (0.092, 0.138, and 0.191).

The confusion patterns indicate that these models often produce plausible-looking weight vectors without reliably moving probability mass toward the oracle-preferred member. Qwen3.5-397B is the strongest of the three, but its gain over equal weighting is modest and much smaller than the trained BlendCast policy. Kimi-K2.5 and InternVL-S1-Pro are especially close to the tie-aware baseline, suggesting that generic meteorological reasoning alone is insufficient to infer case-specific AI member skill from the visual state. Combined with the response-format failures in Table 11, these results support treating the external VLMs as prompt-only baselines rather than optimized ensemble-decision policies.

H Teacher Narratives and SFT Data Construction

Each SFT training example bundles two independently generated supervision signals into a single conversation record: (A) a structured weight label derived retrospectively from ERA5 verification, and (B) a meteorological reasoning narrative written from only the observable forecast context. The key invariant is that *neither the prompt nor the reasoning block exposes any ERA5 field, verification score, or oracle WRMSE value*; ground-truth information is visible only in the structured weight arguments of the assistant turn.

H.1 Step 1 — Oracle Weight Computation

For each (initialization, lead-time) pair, all available member forecast grids are bilinearly regridded to a common $0.5°$ statistics grid (361×720 pixels, 90°N–90°S). Oracle ensemble weights are then found by solving a constrained least-squares problem *independently for each target variable* v:

$$w_v^* = \underset{w \in \Delta^{M-1}}{\arg\min}\ \mathrm{WRMSE}\left(\textstyle\sum_m w_m\, x_v^m,\ y_v\right), \tag{21}$$

where Δ^{M-1} is the M-simplex (weights are non-negative and sum to one), x_v^m is member m's forecast field for variable v, and y_v is the ERA5 analysis at the corresponding valid time. The optimization uses sequential least-squares programming (SLSQP) [48] with $n = 3$ deterministic restarts: the first restart is initialized at equal weights, and the remaining restarts use fixed-seed Dirichlet draws. We use $\mathrm{ftol} = 10^{-9}$ and at most 300 iterations. After convergence, weights below a floor of 0.03 are zeroed and the remaining weights are renormalized, producing sparse solutions that are easier to interpret. Models missing a variable receive zero weight for that variable; if ERA5 is unavailable, the corresponding weights fall back to uniform over the available members.

The per-variable independence of Eq. (21) is intentional: Z500, T2M, and MSL often prefer different member orderings at the same lead time because each variable tracks different physical processes; a single shared weight vector cannot be simultaneously optimal for all variables.

H.2 Step 2 — Narrative Generation via Teacher Large Language Model

Once the oracle weights are known, a frozen teacher model writes a meteorological rationale that justifies those weights from observable evidence alone. The teacher receives:

- The oracle per-variable weights for the current case (the "correct answer").
- The observable text block: ensemble spread, per-model deviations from the ensemble mean, and regional statistics across Global, NH, EU, and EA domains.
- Forecast map images (encoded as JPEG at ≤ 420 px width): per-member absolute fields, ensemble-mean field, spread map, and per-model deviation maps for Z500, T850, T2M, U10, V10, and MSL.

The teacher's system prompt enforces a strict directionality constraint: the narrative must flow *observable signal* $\rightarrow$ *meteorological interpretation* $\rightarrow$ *weight justification*, never "weight X is used, therefore the map shows Y". Each narrative is 200–300 words and must cite key spatial signals by region (e.g. "a coherent positive anomaly over the North Pacific storm track") and reference at least one spread or deviation magnitude to ground the reasoning quantitatively. The teacher model used in this work is Qwen3.5-397B [44].

H.3 Step 3 — Record Assembly

Each finalized example is stored as a multi-turn conversation in JSON Lines (JSONL) format with three message turns:

1. **System**: the BlendCast agent system prompt (tool schema, physical constraints, reasoning guidelines).
2. **User**: the observable state — for the text-only variant this is the structured diagnostic block; for the multimodal variant it additionally includes the rendered forecast map images as inline image content.
3. **Assistant**: a reasoning block containing the teacher narrative and a final weight summary table, followed immediately by a structured tool-call block carrying the oracle per-variable weight dictionary.

A case is discarded if the teacher large language model (LLM) fails to return a valid narrative (e.g. due to token limit truncation inside the reasoning block) or if fewer than two member forecasts are available for any target variable. Both a text-only JSONL (used for SFT warmup) and a multimodal JSONL (used for the full VLM SFT stage) are written from the same pipeline.

H.4 Training Coverage

SFT data are generated from the same 2025 forecast archive used for evaluation, drawing exclusively from 06Z and 18Z initializations to keep 00Z/12Z cases entirely held out. The split contains 9,501 samples across 235 unique dates and ten lead times from T+24 h to T+240 h.

I MV-GRPO Reward Details

MV-GRPO scores each generated rationale–weight trajectory by applying the weights to the frozen member forecasts and computing offline verification metrics against ERA5. The main reward is the oracle-normalized WRMSE improvement in Eq. (9). It is supplemented by decision-validity reward terms for complete normalized weights and by a geostrophic-consistency auxiliary reward,

$$R_{\mathrm{geo}} = \frac{\mathrm{AGE}_{\mathrm{eq}} - \mathrm{AGE}_{\mathrm{agent}}}{\mathrm{AGE}_{\mathrm{eq}} + \epsilon}, \qquad \mathrm{AGE} = \sum_{\phi} w_{\phi} \frac{(u_{10} - u_g)^2 + (v_{10} - v_g)^2}{u_{10}^2 + v_{10}^2 + 1}, \tag{22}$$

where $(u_g, v_g) = \big(-(\rho f)^{-1}\partial_y p,\ (\rho f)^{-1}\partial_x p\big)$ is derived from the blended mean sea-level pressure field and evaluated poleward of 15°. Each reward component is normalized within the rollout group before aggregation so that variables with different physical units and error scales can contribute comparable learning signal.

I.1 Lead-Time Weights

Lead (h)	24	48	72	96	120	144	168	192	216	240
$\phi(h)$	0.70	0.90	1.00	1.00	1.00	0.95	0.90	0.85	0.80	0.75

Table 12: Lead-time factors used in MV-GRPO advantage scaling.

J Training Hyperparameters and Compute

The released configuration sets SFT to 5,000 steps. This shorter budget has not been rerun and does not change the reported experimental results or compute measurements.

K Discussion

Operational interpretation. BlendCast separates forecast generation from forecast combination. This is important operationally because the underlying AI NWP members remain frozen: the ensemble can be improved, audited, and updated without retraining Pangu-Weather, Stormer, FengWu, FuXi, GraphCast, Aurora, or AIFS. The model's output is also more inspectable than a black-box corrected field. Forecasters can compare the assigned per-variable weights with the visible spread, member anomalies, and structured diagnostic block for the same case, making the system closer to a decision-support layer than to an opaque replacement for existing forecast pipelines.

Why multimodal evidence matters. The ablations indicate that neither image evidence nor structured diagnostics alone is sufficient. Forecast maps expose spatial error signatures such as misplaced waves, regional temperature biases, pressure gradients, and wind anomalies, while the diagnostic text compresses member spread, historical skill, and variable-specific reliability into a form that is easy to compare across models. The proprietary VLM baselines further clarify this point: simply placing a strong VLM in front of the same prompt does not create useful ensemble skill. Task-specific SFT and MV-GRPO are needed to turn multimodal weather evidence into calibrated tool-call weights rather than near-uniform allocations.

What the oracle reference does and does not mean. The oracle row is a retrospective upper bound, not a deployable baseline. It is useful because it quantifies how much case-dependent complementarity exists among the frozen members, but it also depends on ERA5 verification at the

Stage	Parameter	Value
SFT	Base policy	Qwen3.5-9B [44], full-parameter vision-language (VL) fine-tuning
SFT	Supervision	ERA5-derived oracle weights and Qwen3.5-397B [44] rationales
SFT	Data split	06Z/18Z for training and 00Z/12Z for validation
SFT	Sequence budget	16,384 tokens with image resizing between 128×256 and 256×512 pixels
SFT	Optimization	5,000 steps in the released configuration, learning rate 10^{-5}, cosine schedule, 0.005 warmup ratio, bf16, gradient checkpointing
SFT	Batch and loss	Per-GPU batch 1, gradient accumulation 2, label smoothing 0.5 for rationale tokens and 0 for weight JSON tokens
MV-GRPO	Initialization	Full-parameter SFT checkpoint, frozen reference model for KL regularization
MV-GRPO	Rollouts	Group size 4, temperature 0.7, maximum 2,200 generated tokens
MV-GRPO	Optimization	3,000 steps, learning rate 5×10^{-6}, cosine schedule, bf16, gradient checkpointing, Adafactor for full-parameter RL
MV-GRPO	KL and clipping	$\beta = 0.001$, $\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.28$
MV-GRPO	Reward terms	Oracle-normalized WRMSE reward, format reward 0.30, geostrophic-consistency reward 0.20, parse-failure penalty -0.5
MV-GRPO	Selected run	Selected refinement run, about 160 wall-clock hours on four H20 GPUs

Table 13: Training settings and compute, including the updated SFT budget for the released configuration.

valid time. BlendCast never observes those future fields at inference. Closing a large fraction of the equal-weight-to-oracle gap is evidence that the present atmospheric state contains predictive information about future member reliability, not evidence that the oracle itself has been recovered. This distinction is especially important for regimes where the best member changes rapidly with variable, lead time, and flow state.

Limits of the current formulation. The present policy predicts deterministic per-variable weights. This makes the output easy to evaluate and audit, but it does not yet represent uncertainty over weights or forecast outcomes. The oracle labels are also optimized independently by variable, which captures the fact that Z500, T2M, MSL, humidity, and winds can prefer different members, but it only partially models cross-variable covariance. The MV-GRPO geostrophic auxiliary reward addresses one physical consistency channel, yet a fully coupled multi-variable objective remains future work. Similarly, each RL episode is a single initialization–lead pair rather than a coherent multi-lead trajectory, so temporal smoothness of the weights is encouraged only indirectly through the training distribution.

Resolution, regime coverage, and transfer. The visual state must be compressed to fit the VLM token budget. This is adequate for broad synoptic patterns, but small-scale high-impact systems such as tropical cyclones, mesoscale convective events, sharp fronts, or coastal wind jets may require targeted high-resolution crops or specialized image channels. The current evaluation uses global 2025 test cases and eight variables; deployment requires stress tests on rare regimes, regional domains, additional vertical levels, and operational latency constraints. The same framework may also transfer to other geophysical ensembles, but that transfer requires evaluation as a new reliability-forecasting problem rather than extrapolation from weather results alone.

Broader impacts and responsible use. More accurate and inspectable weather ensembles can support earlier planning for weather-sensitive sectors, but learned weighting policies can also fail under rare or out-of-distribution regimes. BlendCast is therefore intended as a decision-support layer over existing forecast pipelines rather than an autonomous warning system. The explicit weights and rationales make the policy easier to monitor, and operational use requires human forecaster review, routine verification, and safeguards against over-trusting the model in high-impact events.

Future directions. Natural extensions include probabilistic weight distributions, calibrated uncertainty for the blended forecast, sequence-level rewards over entire multi-day trajectories, and higher-resolution visual adapters for severe-weather regimes. Another useful direction is online monitoring: because BlendCast emits explicit weights and rationales, one can track when the policy over-trusts a member, drifts toward uniform weighting, or disagrees with known regime-specific model behavior. These diagnostics could make learned ensemble policies easier to integrate into human-in-the-loop forecasting workflows.

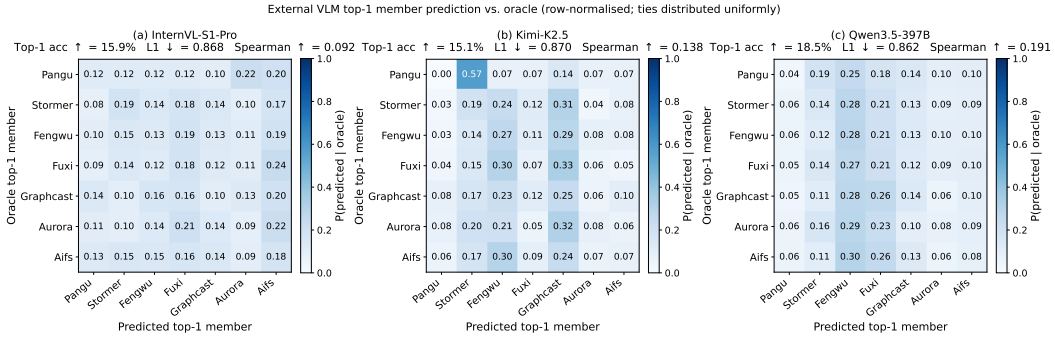


Figure 14: **External VLM predicted member trust compared with the oracle.** Rows denote the oracle top-1 member and columns denote the predicted top-1 member. Each panel is row-normalized, with ties distributed uniformly. Summary scores above each panel report top-1 accuracy, weight distance, and rank correlation on the executable parsed rows.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and Section 1 state the forecasting problem, the BlendCast framework, the training recipe, and the reported improvement claims. The experimental scope is tied to the held-out 2025 evaluation in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Appendix K discusses deterministic weight outputs, variable-wise oracle assumptions, visual compression, rare-regime coverage, and transfer beyond the evaluated setting.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: Section 3.1 states the simplex constraints and measurability setting, and Proposition 1 includes a complete proof of oracle dominance.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: Sections 3 and 4 specify the model, state, reward, baselines, data split, and metrics. Appendices A–J provide the full evaluation protocol, metric definitions, baselines, input schema, reward details, and training hyperparameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The anonymized project link in the abstract will provide the code, prompts, scripts, and release documentation needed to reproduce the main experiments, together with instructions for obtaining external weather data under their own terms.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 4 describes the held-out 2025 test setup, baselines, model choice, and training summary. Appendix J gives SFT and MV-GRPO hyperparameters, optimizer details, batch settings, reward terms, and compute.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: The paper reports aggregate deterministic verification over all 2025 held-out initialization times but does not currently include bootstrap confidence intervals, error bars, or formal significance tests. This is a reporting limitation rather than a change to the evaluation protocol.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Section 4 states that training used four H20 GPUs, and Appendix J reports wall-clock training time and stage-level compute details for SFT and MV-GRPO.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm compliance. The work uses meteorological data and model outputs, contains no personal data or human-subject experiments, and preserves anonymity in the submission materials.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes].

Justification: Appendix K discusses decision-support benefits for weather-sensitive planning and the risk of failures under rare or out-of-distribution regimes. It also recommends human forecaster review, routine verification, and monitoring of the emitted weights and rationales.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The released assets are research code, prompts, weather-processing scripts, and BlendCast-specific forecasting artifacts rather than a general-purpose pre-trained language model, image generator, or scraped dataset. The paper nevertheless frames the method as decision support and discusses monitoring and human review in Appendix K.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: Existing datasets, models, and software dependencies are cited in the main paper and appendix, and the authors confirm that their licenses and terms of use have been checked and respected. Release documentation will include the relevant license and attribution notes.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes].

Justification: The paper introduces BlendCast code, prompts, training/evaluation scripts, and release artifacts, which will be documented through the anonymized project release. Appendices G, H, I, and J document the schema, data construction, reward design, and training settings.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human subjects, so IRB or equivalent approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: VLM usage is central to the method and is described throughout Section 3.2, including the Qwen3.5-9B policy and Qwen3.5-397B teacher. Appendices H and J further document teacher narrative generation and training details.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.