HyperCLOVA X 32B Think

NAVER Cloud
HyperCLOVA X Team*
Huggingface Model Card

Abstract

In this report, we present HyperCLOVA X 32B Think, a vision–language model designed with particular emphasis on reasoning within the Korean linguistic and cultural context, as well as agentic ability. HyperCLOVA X 32B Think is pre-trained with a strong focus on reasoning capabilities and subsequently post-trained to support multimodal understanding, enhanced reasoning, agentic behaviors, and alignment with human preferences. Experimental evaluations against comparably sized models demonstrate that our model achieves strong performance on Korean text-to-text and vision-to-text benchmarks, as well as on agent-oriented evaluation tasks. By open-sourcing HyperCLOVA X 32B Think, we aim to support broader adoption and facilitate further research and innovation across both academic and industrial communities.

1 Introduction

Human reasoning is not solely grounded in textual knowledge but is often informed by visual context. Although large language models (LLMs) have demonstrated solid progress in reasoning ability, their application remains largely confined to scenarios centered on textual information. Expanding training beyond text is indispensable for achieving a deeper and more comprehensive understanding of the real world (Huh et al., 2024; Chen et al., 2025). Moreover, as models are more closely integrated into real-world environments, the ability to interact with external tools is becoming essential. Therefore, dedicated training phases aimed at instilling agentic ability are required. In parallel, there is a growing demand for models with a deep understanding of the linguistic and cultural context of Korea, while maintaining the ability to leverage the extensive knowledge available in English. Addressing this requires a training curriculum that incorporates both English and Korean data.

Motivated by these needs, we introduce HyperCLOVA X 32B Think (THINK). THINK builds on its predecessor (HyperCLOVA X Team, 2025) to achieve the capability of understanding both text and vision within a unified framework, while maintaining an emphasis on reasoning in the context of the Korean language and culture, with an added focus on agentic ability. THINK is a decoder-only Transformer with both text tokens and vision patches projected onto the same continuous embedding space and processed together through standard self-attention. The pre-training curriculum consists of multiple stages with particular attention to honing the ability to reason in Korean. Post-training first instills multimodal capabilities into the model through supervised fine-tuning. Then, the model is further refined in the areas of multimodal reasoning, agentic behavior, and human preference alignment via several reinforcement learning methodologies. This process is conducted in accordance with the NAVER AI Ethics guidelines for enhanced safety.

We evaluate THINK on various text-to-text (*e.g.*, KMMLU, HAERAE-1.0, MMLU, and PIQA), vision-to-text (*e.g.*, KoNET, K-DTCBench, SEED-IMG, and DocVQA), and agent (*e.g.*, Tau^2 and

*The complete list of contributors appears in the Contributions and Acknowledgments section. For correspondence, please contact dl_hcx_technical_report@navercorp.com.

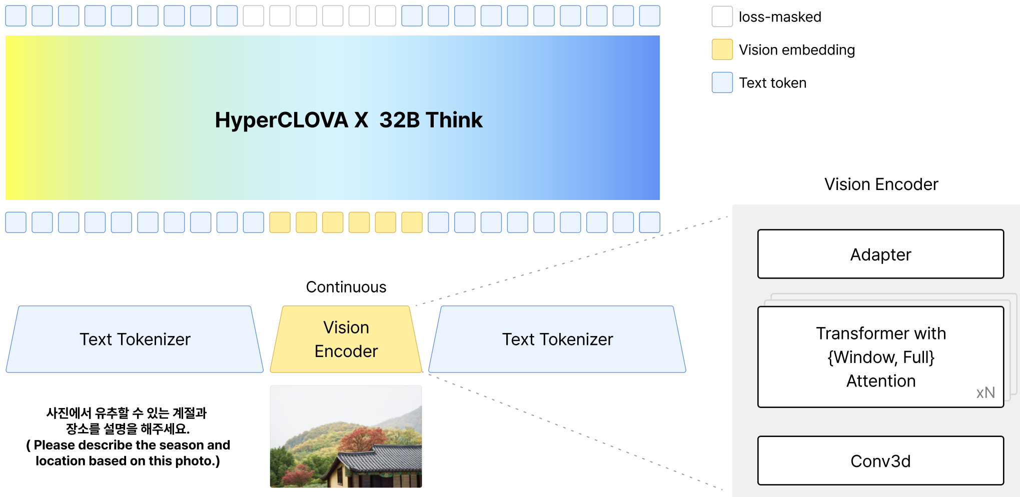


Figure 1: Overview of HyperCLOVA X 32B Think. Text is tokenized into discrete tokens and mapped to token embeddings, while images are encoded into continuous visual embeddings by the vision encoder. Visual embeddings are projected to the Transformer embedding space and interleaved with text embeddings, enabling joint multimodal processing with a single decoder-only Transformer.

Terminal Bench) benchmarks in Korean and English. We find that THINK exhibits competitive performance against comparably sized models on text-to-text and vision-to-text benchmarks in Korean, as well as on agent benchmarks.

THINK is released as an open-weight model under a custom license that permits commercial use subject to specified conditions. With its competitive performance in Korean and agentic abilities, we present THINK as a valuable resource for academic and industry partners in both the Korean and global research community.

2 HyperCLOVA X 32B Think (THINK)

As shown in Figure 1, THINK combines a decoder-only Transformer backbone with a text tokenizer and a vision encoder to support unified text-image understanding. In the remainder of this section, we describe the architecture, text tokenizer, and vision encoder in more detail.

Architecture. We adopt a standard decoder-only Transformer (Vaswani et al., 2017) and follow design choices commonly used in recent LLMs to improve its training stability and inference efficiency. Specifically, the model applies RMSNorm (Zhang and Sennrich, 2019) as a pre-normalization and adopts SwiGLU (Shazeer, 2020) as the activation function in the feed-forward networks. We use Rotary Positional Embeddings (RoPE) (Su et al., 2023) for positional encoding with the rotary base parameter set to 500,000 to enhance long-context modeling ability. To improve KV-cache memory efficiency, we employ grouped-query attention (Ainslie et al., 2023) with eight query groups. In addition, we remove bias terms from linear layers and untie the input and output embedding weights. We instantiate the backbone with 72 layers and a hidden size of 5,120, yielding a deep and wide model with sufficient capacity for modeling large-scale knowledge.

For multimodal integration, we employ a streamlined fusion strategy (Liu et al., 2023a; Deitke et al., 2025; Beyer et al., 2024) that minimizes architectural complexity while improving efficiency. Specifically, visual representations are projected into the Transformer embedding dimension and interleaved with text token embeddings. To ensure stability and facilitate seamless collaboration, we maintain the original RoPE configuration of the LLM backbone without introducing intrusive multimodal extensions to the positional encoding. This architecture enables the joint processing of visual and textual representations within a shared representational space while preserving the core efficiency of the pre-trained decoder-only Transformer.

Tokenizer	English			Korean	
	General	Code	STEM	General	STEM
Tiktoken (OpenAI, 2022)	4.82	3.24	3.50	1.56	1.90
HyperCLOVA X Think (HyperCLOVA X Team, 2025)	4.74	3.14	3.48	1.82	2.11
HyperCLOVA X 32B Think (THINK)	4.55	2.72	3.35	2.15	2.18

Table 1: Tokenizer compression rate is measured as the average number of characters per subword token across domains, where the general column represents all datasets except for code and math domain datasets. Higher values indicate fewer tokens for the same domain datasets.

Text-Tokenization. The tokenization process sequentially employs a pre-tokenizer step and a subword tokenizer step to convert text into a discrete token sequence. The pre-tokenizer step is specifically designed to prevent merging different language scripts, as implemented in DeepSeek-V3 (Liu et al., 2024). Such pre-tokenization rules play a crucial role in determining tokenizer compression efficiency and downstream performance (Dagan et al., 2024; Wegmann et al., 2025; Liu et al., 2025a; Schmidt et al., 2025). An empirical study further indicates that tokenizers yielding a lower upper bound on the Kolmogorov complexity of the tokenized text tend to achieve lower language modeling loss and improved downstream task performance, and that this complexity upper bound is highly sensitive to pre-tokenization rules (Chung and Kim, 2025).

Following the approach of LLaMA (Touvron et al., 2023) and Qwen (Bai et al., 2023), we implement single-digit tokenization for numbers to prevent performance degradation in code and mathematics tasks (Garreth Lee and Wolf, 2024). To mitigate *token boundary bias* that may arise from this pre-tokenization (Kudo, 2018), we apply a regularization technique known as StoChasTok (Sims et al., 2025). We deliberately set the StoChasTok probability to a low value, as high probabilities tend to frequently split words into individual characters, which inflates the token count and consequently reduces training efficiency. We balance this by applying an appropriate probability value. For subword tokenization, we modify a representative open-source tokenizer, *e.g.*, , tiktoken (OpenAI, 2022), originally optimized for English, Code, and Math. As shown in Table 1, these English-centric tokenizers achieve lower compression rates for Korean stems and general text than English. To address this limitation while maintaining performance on the original target domains, we introduce a vocabulary adaptation method comprising two stages: 1) pruning and 2) substitution.

First, the pruning stage removes existing merge rules associated with Korean letters, which are identified via an automated ISO15924 script detection. Moreover, we remove low-utility rules from non-target languages to maximize the compression rate, resulting in degradation below 1%. The scope of this trimming is carefully calibrated to ensure that the degradation of compression rate in non-Korean text. Second, the substitution stage inserts new merge rules derived from a trained morpheme-aware subword tokenizer trained on our internal corpus (HyperCLOVA X Team, 2025). Unlike traditional expansion methods (Kim et al., 2024b) that append new rules at the lowest priority, our substitution stage inserts these rules directly into the priority slots vacated during the pruning stage. As shown in Table 1, our tokenizer achieves a substantial increase in Korean compression efficiency (up to 2.15) compared to tiktoken and the tokenizer of the previous version, while maintaining comparable performance across English and STEM domains with minimal trade-offs.

Vision-Encoder. For the visual understanding component, THINK adopts the Vision Transformer (ViT) architecture from Qwen2.5-VL (Bai et al., 2025). The encoder employs a 3D-convolutional layer for patch embedding with a $2 \times 14 \times 14$ kernel, allowing for the unified representation of both static images and video frame sequences (Bai et al., 2025). To maintain computational efficiency when processing high-resolution or long-duration inputs, the encoder incorporates a local-window attention mechanism that constrains interactions to local neighborhoods.

We employ a simplified Vision-Language junction to ensure architectural stability and production efficiency, leveraging the vision encoder that natively supports a multimodal extension of RoPE (Su et al., 2024). Specifically, we utilize a linear adapter after the vision encoder module to project the encoded dense representations, aligning them with the hidden dimension of the LLM backbone (Liu et al., 2023a; Deitke et al., 2025; Beyer et al., 2024). This streamlined integration enables joint multimodal processing without requiring intrusive modifications to the core attention mechanism of the decoder-only Transformer.

A primary design objective of THINK is to improve the efficiency of visual tokens. During training, images are constrained within a 1920×1080 pixel envelope ($\approx$2.1 MP), which reduces overall training cost by approximately 53% in terms of GPU-hours compared to the original Qwen2.5-VL setting ($\approx$12.8 MP), while maintaining comparable downstream performance.

While the vision encoder supports a larger pixel budget, we default to the training resolution at inference time to reduce visual token count and improve serving efficiency. For video sequences, we apply a uniform sampling strategy with a maximum of 120 frames, where each frame is constrained to a 378×378 pixel resolution ($\approx$0.14 MP). Under this configuration, each video is compressed to a maximum budget of approximately 11 K tokens, while static images are represented by at most 3 K tokens. This approach offers significant computational advantages over contemporary models; for instance, Qwen2.5-VL (Bai et al., 2025) and SEED-1.5-VL (Guo et al., 2025) often allocate over 16K and 82K tokens per input, respectively.

Notably, the vision encoder remains unfrozen throughout training to acquire Korean-centric multimodal capabilities. This full-parameter tuning is essential for internalizing Korean-specific visual contexts, including cultural entities, local landmarks, and high-density Korean-script OCR. By optimizing token allocation and specializing the visual representation for the Korean domain, THINK achieves a superior balance between multimodal reasoning performance and inference throughput for large-scale deployment.

3 Pre-Training

A primary focus of the pre-training phase is to endow the model with the capacity to reason in Korean. For this, we first build a Korean-centered training dataset with a scalable pipeline for cleaning, filtering, masking personal identifiable information (PII), and adding synthetic examples (Section 3.1). We then train with a four-stage curriculum that uses longer contexts and gradually increases the proportion of high-quality reasoning data (Section 3.2).

3.1 Data Preparation

Consistent with the predecessor (HyperCLOVA X Team, 2025), we structured the THINK data pipeline to prioritize scalability, reusability, and quick refresh. To achieve this, we implement the pipeline with a hybrid processing framework that integrates Datatrove (Penedo et al., 2024b) and NeMo-Curator[2]. This approach minimizes the operational burden of updating datasets while maintaining data integrity.

Our preprocessing pipeline comprises four stages: (1) collection and normalization stage: we apply lightweight cleaning, render documents into a consistent representation, and map heterogeneous inputs to a unified schema; (2) quality scoring stage: we compute document-level structural and linguistic signals, store them as metadata, and detect and mask PII; (3) filtering stage: we construct stage-specific corpora by combining threshold-based heuristics with learned quality scores; and (4) serialization stage: we serialize the selected documents into sharded files designed for efficient, streaming-based training.

Data Filtering. Data filtering plays an important role in determining the final training corpus for THINK in the filtering stage of the data pipeline. We adapt a two stage pipeline that follows the same structure as the predecessor (HyperCLOVA X Team, 2025). The first stage applies lightweight, rule-based heuristics (Weber et al., 2024; Penedo et al., 2024a) to remove structurally malformed or clearly noisy documents, using document statistics such as symbol ratios, sentence statistics, and boilerplate indicators. This first stage is identical to the previous version.

The second stage performs model-based quality scoring and threshold-based filtering to remove low-quality content that is difficult to identify with heuristic filtering in the previous stage alone. For English data, we retain lightweight scorers based on FastText (Joulin et al., 2016) and a transformer encoder Merrick et al. (2024); these models are trained to regress 0–5 quality ratings for efficient

[2] `https://github.com/NVIDIA-NeMo/Curator`

Classifier	FineWeb2 Dataset		In-house Dataset		Average	
	Acc.	Recall	Acc.	Recall	Acc.	Recall
HyperCLOVA X Think (HyperCLOVA X Team, 2025)	69.0	86.8	73.5	83.2	71.3	85.0
HyperCLOVA X 32B Think (THINK)	**86.9**	**92.5**	**90.9**	**92.7**	**88.9**	**92.6**

Table 2: Comparison of Korean document quality classifiers measured in accuracy (Acc) and recall (Rec). Labels are annotated 0–5 quality scores and binarized into high-quality (≥ 3) and low-quality (< 3). Recall measures recall on the low-quality class.

large-scale filtering (Li et al., 2024b; Penedo et al., 2024a). This approach has been shown to work well for scoring English educational and STEM documents.

In contrast, our Korean corpus is broad and includes substantial non-STEM, user-facing content such as game guides, consumer reviews, and travel guides. When we applied the English educational scorer to Korean without adaptation, it struggled to distinguish low-quality documents and frequently filtered out useful documents that are not related to the curriculum. To address this mismatch, we train a Korean-specific 0–5 quality regressor on (document, score) pairs obtained from (1) high-quality external corpora (*e.g.*, , FineWeb-Edu2 (Penedo et al., 2025)), (2) internal corpora that cover our target domains, and (3) an additional pool of clearly low-quality documents curated via LLM-based annotation. We fine-tune a Qwen3-Embedding-0.6B with a lightweight regression head (Zhang et al., 2025) using our training dataset. Table 2 shows that our Korean scorer consistently improves both accuracy and low-quality recall over a Korean reproduction of FineWeb-Edu-Classifier. Motivated by recent work (Saada et al., 2025) that the primary gains from data curation come from excluding clearly low-quality documents rather than selecting clear high quality dataset, we choose thresholds for Korean dataset and focus on low-quality removal. Finally, we apply MinHash-based near-duplicate removal after filtering to improve dataset efficiency.

Synthetic Data Generation. To complement the data filtering strategy and to mitigate the limited availability of high-quality Korean data, we generate synthetic data using two complementary approaches.

First, existing documents are improved and rewritten to enhance their quality. High-quality seed data collected from diverse STEM sources are reformulated to improve coherence while preserving information content. For example, we take high-quality problem–answer pairs and synthesize intermediate reasoning steps, producing self-contained examples that include complete, high-fidelity reasoning traces. For mathematics, we further curate these traces to favor concise reasoning steps, avoiding unnecessarily long solutions.

Second, new text is generated from carefully selected high-quality seed data. Many high-quality Korean documents contain figures, tables, or diagrams that include essential information. Since removing these elements in the normalization stage makes the remaining text incomplete for understanding the document, we rewrite such documents into self-contained text by using LLMs to generate explicit descriptions from captions and surrounding context. Furthermore, for personally identifiable information, naive masking often degrades fluency and can miss context-dependent PII. We apply a two-step LLM pipeline: (1) detect PII-bearing sentences or spans, then (2) rewrite them with safe, generic substitutes while preserving meaning. This reduces residual PII in the corpus from 89.28% to 0.13% without loss of the quality.

Table 3 illustrates the data composition across training stages, which are detailed in the next subsection. It illustrates how the proportions of general, code, mathematics, and instruction-oriented data are progressively adjusted throughout the training curriculum.

3.2 Pre-Training Curriculum

We adopt a multi-stage curriculum learning strategy (Olmo et al., 2025; HyperCLOVA X Team, 2025) in which the training signal becomes progressively more demanding across stages. Each stage is designed to emphasize a distinct capability, starting from broad linguistic and factual coverage and then moving toward long-context modeling and reasoning, while gradually increasing both the context window and the proportion of curated high-quality data. Notably, we incorporate reasoning-

Domain	Stage 1	Stage 2	Stage 3	Stage 4
General	79.4%	61.6%	59.1%	17.0%
Code	12.0%	20.1%	20.0%	25.2%
Math	8.6%	18.2%	20.0%	25.3%
Instruction tuning	0.0%	0.1%	1.0%	32.5%

Table 3: Pre-training data distribution across stages. The General domain dataset comprises all documents excluding code, math, and instruction tuning datasets. The instruction tuning dataset consists of instruction-response pairs.

oriented synthetic data (*e.g.*, , coda and math) during pre-training, rather than post-training, which has been observed to improve downstream reasoning performance (Wang et al., 2025a). The detailed data compositions of each training stage are provided in Table 3.

Throughout all stages, we apply Fill in the Middle (Bavarian et al., 2022) to 10% of tokens, which strengthens infilling behavior and improves code generation and long-context modeling. Because longer contexts require higher memory and computation costs and lead to different effective batch sizes across stages, we set a different learning rate for each stage to maintain stable and efficient optimization.

Stage 1: Foundation Knowledge Construction. The first stage builds broad linguistic and factual coverage from a multilingual corpus, primarily Korean and English. Training uses sequences up to 4K tokens and consumes 6 trillion tokens. The learning rate is linearly warmed up from $1.5e^{-5}$ to $3.1e^{-5}$ over the first 2,000 steps, and then kept constant until the end of the first stage.

Stage 2: Context Extension and Quality Up-sampling. The context length is expanded from 4K to 8K, accompanied by a substantial increase in long-form documents such as academic papers and source code. From this stage onward, high-quality supervised fine-tuning data and chain-of-thought data are mixed into pre-training to establish instruction-following capability. To ensure stable convergence, we apply a cosine decay learning rate schedule and reduce the peak learning rate to approximately 10% of that used in the previous stage.

Stage 3: Advanced Reasoning and Long-Context Adaptation. In the third stage, we extend the context length to 32K to further enhance long-context modeling and reasoning abilities. We apply stricter filtering to general Korean and English documents and augment the mixture with reasoning-oriented synthetic data from math and code. To better support very long contexts, we increase the RoPE base frequency from 500K to 5M, and apply cosine decay down to $1.0e^{-5}$.

Stage 4: High-Quality Annealing. The last stage boosts reasoning ability with a carefully curated high-quality reasoning dataset. We use a carefully selected reasoning-oriented dataset, predominantly composed of math, code, and instruction data. The learning rate starts at $1.0e^{-5}$ and is annealed to one-third of its initial value to maximize reasoning and complex task performance.

4 Post-Training

As illustrated in Figure 2, the post-training of THINK is conducted in two phases—supervised fine-tuning (SFT) and reinforcement learning (RL)—with the goal of strengthening reasoning, while also instilling multimodal and agent abilities. More specifically, we first build a strong foundation for text instruction following and extend it to multimodal capabilities via SFT (Section 4.1). We then further refine the model through RL with objectives targeting reasoning, agentic behavior, and alignment with human preferences (Section 4.2). This section details the process with particular attention to elements that differ from the predecessor (HyperCLOVA X Team, 2025).

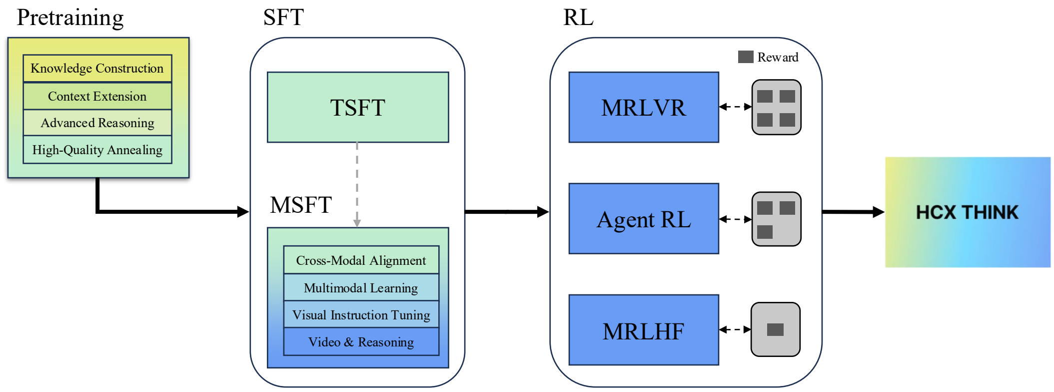


Figure 2: Training pipeline of HyperCLOVA X 32B Think.

4.1 Supervised Fine-Tuning (SFT)

THINK undergoes a two-stage SFT process for enhancing text-based and multimodal instruction following. Since a well-defined chat template is essential for controlling the generation process, we introduce a unified chat template to be used across various scenarios. See Appendix A for examples.

Text Supervised Fine-Tuning (TSFT). TSFT leverages three types of training data: *non-reasoning* for general instruction-following; *reasoning* for multi-step reasoning; and *agent* for sequential decision making and tool-augmented interactions. These data types are not strictly disjoint and may overlap in practice. Each data type is processed using a distinct filtering pipeline tailored to its specific characteristics and training objectives. To ensure high-quality supervision signals, we strictly retain only those trajectories directly sampled using our system that pass all associated test cases. As a result, each selected trajectory captures the complete interaction lifecycle, including code generation, execution, feedback observation, debugging, and verified solution production, while failed or partial runs are excluded.

Multimodal Supervised Fine-Tuning (MSFT). MSFT consists of four stages: *cross-modal alignment* for aligning visual features to the text embedding space; *multimodal knowledge learning* for broadening visual knowledge and representation capacity; *task-oriented instruction tuning* for enhancing task-oriented multimodal interaction; and *advanced reasoning and video understanding* for establishing long-context multimodal reasoning and temporal dynamics. This sequential approach is designed to first establish cross-modal alignment and subsequently instill complex reasoning and temporal understanding capabilities. To mitigate the degradation of previously acquired text-based reasoning capabilities, TSFT data are continuously incorporated into all stages except Stage 1, with their proportion progressively decreasing from 15.3% in Stage 2 to 6.9% in Stage 4.

4.2 Reinforcement Learning (RL)

THINK is trained using a combination of reinforcement learning approaches, including multimodal reinforcement learning with verifiable rewards (MRLVR), agent reinforcement learning (Agent-RL), and multimodal reinforcement learning from human feedback (MRLHF).

Multimodal Reinforcement Learning with Verifiable Rewards (MRLVR). THINK develops its core reasoning capabilities through MRLVR across a broad range of tasks and domains. Training is structured into multiple MRLVR stages, each sharing a common verifiable reward framework while targeting a distinct aspect of reasoning behavior: *integrated domain RLVR* for general reasoning; *length-control RLVR* for controlling reasoning length; *multi-turn RLVR* for checking objective completion; and *instruction-following RLVR* for instruction-following.

Throughout training, verifiable rewards are complemented by auxiliary reward terms that regulate output format, language consistency, and repetition. To control task difficulty, offline data sampling is applied prior to training to filter out instances that are either trivially easy or excessively challeng-

	General Agent	SWE Agent
Context window size (tokens)	44K	128K
Max prompt length (tokens)	12K	8K
Max response length (tokens)	32K	120K
GRPO group size	8	16

Table 4: Context window size, maximum prompt length, maximum response length, and selected training configurations for agent reinforcement learning. The context window size equals the sum of the prompt and response lengths. Response length includes both model-generated outputs and environment observations.

ing, using precomputed pass rates from the initial model. Training is carried out using variants of Group Relative Policy Optimization (GRPO) (Shao et al., 2024b), augmented with techniques such as dynamic sampling and adaptive clipping (Yu et al., 2025; Xi et al., 2025) for sample diversity and training stability.

Reinforcement Learning for Agent (Agent-RL). We distinguish general agents from SWE agents and train them in different phases. The agents exhibit markedly different rollout characteristics, particularly in rollout duration, interaction depth, and response length. On average, SWE agent rollouts are longer (37k vs 4k tokens) with more turns (40 vs 11), resulting in higher computational cost and longer interaction traces. Since environment characteristics differ across phases, we design phase-specific environments and reward. Here, reward is a weighted sum of the following rewards, where the weights are optimized for each domain and task difficulty level: *environment* for primary evaluation based on rule-based verifiable reward and LLM-as-a-judge-based natural language reward; *format* for the output format; and *language consistency* for input-output language matching.

Training proceeds in two stages: we first stabilize the core behaviors of a general agent, and then progressively specialize it into an SWE agent. We perform training using GRPO (Shao et al., 2024a). To account for the substantial differences between general and SWE agents in interaction length and learning dynamics, we decouple the token length configurations by phase. For general agent training, we use a 44K context window, with the maximum prompt length capped at 12K tokens and the maximum response length capped at 32K tokens. For SWE agent training, we use a 128K context window, with the maximum prompt length set to 8K tokens and the maximum response length set to 120K tokens. Consequently, the total token budget per agent rollout—defined as the sum of prompt tokens and response tokens (including not only model-generated outputs but also environment observation tokens)—differs substantially across phases, as summarized in the Table 4.

We also adjust GRPO's group size by phase: we use a group size of 8 for general agent training and 16 for SWE agent training. This choice is motivated by the observation that SWE agent tasks exhibit markedly lower success rates and tend to have sparse, high-variance rewards; increasing the group size helps obtain more reliable learning signals and stabilizes policy optimization under these conditions. From an algorithmic perspective, following DAPO (Yu et al., 2025), we remove the KL-penalty term that can suppress generation diversity, and we encourage exploration by setting the upper bound of PPO clipping larger than the lower bound.

Building on these settings, Agent-RL adopts the following training strategies, which are designed to reflect the characteristics of agent tasks and to improve training stability, robustness, and ease of deployment in real-world services: *offline reward filtering* for curating problems of intermediate difficulty for optimal learning; *thinking mode fusion* for switching between reasoning and non-reasoning modes; *adaptive rollout sampling parameter* for improving sample efficiency and training stability; *self-correction via error feedback* for learning recovery behaviors from tool-call exceptions; and *service caching* for avoiding repeated queries.

Multimodal Reinforcement Learning from Human Feedback (MRLHF). We use MRLHF to align the behavior of THINK with human preferences for improved harmlessness, creativity, and other qualities. We begin by training a reward model (RM) on a diverse collection of open-source and in-house ranking datasets. In these datasets, response preferences are annotated either by human evaluators or by judge models. Once the reward model is trained, we optimize the policy model to maximize the rewards assigned by the RM using the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm. Through empirical experiments, we find that PPO, requiring only a single rollout per prompt, enables substantially more optimization steps within a fixed training time compared to variants of GRPO, which rely on multiple rollouts per prompt. Based on this efficiency advantage, we adopt PPO as our primary policy optimization method.

We also introduce an auxiliary reward term to explicitly control both the output format and language of the generated responses. The format-related component enforces adherence to the chat template. The language-related component leverages per-prompt language statistics derived from the chosen responses. If the model generates content in a language that is absent or has a low prior probability under these statistics, a penalty is applied. Both reward model and policy model training are applied exclusively to the final generated responses, while intermediate reasoning steps or think paths are not used for training.

5 Evaluation

Recent studies have shown that evaluation of LLMs with reasoning ability are highly sensitive to answer extraction and evaluation protocols (He et al., 2024; Jo et al., 2025). Therefore, we test all models in this report using Omni-Evaluator [3], a unified evaluation framework designed to support reproducible benchmarking across various inference engines and modalities.

Omni-Evaluator decouples inference engines from evaluation logic, enabling consistent evaluation across on-device inference (*e.g.*, HuggingFace), API-based engines (*e.g.*, vLLM, SGLang), and in-house serving systems. In addition, it integrates modality-specific evaluation toolkits into a single framework, which allows unified evaluation across understanding and generation tasks in text, vision, and audio, while ensuring consistency across implementations.

All evaluations are conducted under fixed, version-pinned configurations, including inference engines and evaluation toolkits, ensuring fair and reproducible comparison across models, inference engines, and modalities.

5.1 Baselines

We evaluate our model across three benchmark categories: Text-to-Text, Vision-to-Text, and Agent. For both the Text-to-Text and Vision-to-Text benchmarks, we compare our model against widely adopted open-source vision-language models of comparable scale. Specifically, we use Qwen3-VL-32B-Thinking (Yang et al., 2025) and InternVL3_5 38B-Thinking (Wang et al., 2025b) as baselines, as they are well-established VLMs known for strong multimodal reasoning performance. This comparison allows us to assess our model against representative open-source VLMs under both purely textual and vision-conditioned reasoning settings. To assess our model's fundamental reasoning capabilities beyond the scope of vision language models, we also include EXAONE 4.0 32B (LG AI Research, 2025) as a text-only baseline. Although EXAONE is trained as a text-only model without multimodal inputs, it serves as a reference point for comparison between multimodal model and text-only model performance.

Agent benchmarks focus on capabilities beyond standard question answering, including multi-step planning, tool use, and interaction with environments. We compare our model not only with open-source baselines but also with substantially larger and more capable commercial models. In particular, we include GPT-5.1 (OpenAI, 2025) and Qwen3 235B-A22B (Yang et al., 2025), which are widely regarded as strong frontier models for agentic reasoning. These models provide a realistic upper-bound on agent performance in practical deployment scenarios.

5.2 Text-to-Text Benchmarks

We aim to develop a model that goes beyond achieving numerical superiority on global benchmarks and instead intrinsically reflects a Korean-centric perspective, incorporating linguistic and cultural inclusiveness. We parse the answer from the generated output using regular expressions. If the answer cannot be found, the sample is considered incorrect.

Korean Benchmarks. On Korean Text-to-Text benchmarks, THINK generally outperforms the baselines across most benchmarks. To systematically evaluate Korean text understanding and gener-

[3] The repository will be publicly released.

Language	Dataset	HyperCLOVA X 32B Think	EXAONE 4.0 32B	Qwen3-VL 32B-Thinking	InternVL3_5 38B-Thinking
Text-to-Text					
Korean	KMMLU	71.3	**75.2**	47.4	61.2
	KoBALT	**50.6**	48.3	21.1	31.9
	CLIcK	**75.2**	73.1	62.4	62.7
	HAERAE Bench 1.0	**87.4**	64.3	51.5	43.0
	Flores+ (En→Ko)	**31.8**	30.9	24.8	26.9
English	MMLU	87.7	**89.9**	71.5	76.7
	HellaSwag	57.2	**65.7**	62.1	59.0
	PIQA	76.7	75.7	**88.0**	81.1
	Flores+ (Ko→En)	30.9	30.2	28.2	**31.8**
Vision-to-Text					
Korean	KoNET	**75.1**	–	74.3	33.9
	K-MMBench	88.1	–	**91.1**	81.1
	K-DTCBench	93.3	–	**95.4**	92.1
English	SEED-IMG	77.9	–	79.3	**88.0**
	LLAVA-W	106.4	–	**111.9**	100.7
	TextVQA	**85.4**	–	82.5	82.2
	DocVQA	95.5	–	**96.5**	93.3
Agent					
English	Tau^2 – Airline	**58.0**	45.6	54.4	–
	Tau^2 – Retail	**71.6**	59.5	57.0	–
	Tau^2 – Telecom	**65.1**	26.1	33.3	–
	Terminal Bench 1.0	**21.8**	10.0	11.3	–
	Terminal Bench Hard	**9.9**	3.6	6.9	–

Table 5: Performance comparison across Text-to-Text, Vision-to-Text, and Agent benchmarks (User-simulator: GPT-4.1) in English and Korean.

Dataset	HyperCLOVA X 32B Think	GPT-5.1 (Medium)	Qwen3 235B-A22B
User-simulator: GPT-4.1			
Tau^2 – Airline	58.0	**64**	36.8
Tau^2 – Retail	71.6	**78.1**	51.4
Tau^2 – Telecom	65.1	**80.9**	20.5
User-simulator: GPT-5.1			
Tau^2 – Airline	58.0	**64**	41.6
Tau^2 – Retail	67.2	**73.6**	58.4
Tau^2 – Telecom	**92.3**	81.2	24.2

Table 6: Tau^2 results under different user simulators, comparing HyperCLOVA X 32B Think with large-scale frontier models.

ation capabilities, we categorize our evaluation suites into three complementary ability dimensions: (1) Korean STEM and professional knowledge, (2) Korean cultural and commonsense knowledge, and (3) Korean linguistic and grammatical competence.

For linguistic and grammatical competence, we use KoBALT (Shin et al., 2025), which probes fine-grained Korean linguistic knowledge (*e.g.*, syntax, semantics, pragmatics, phonology, and morphology) and therefore tests structural understanding beyond surface-level recall. For Korean cultural and commonsense knowledge, we adopt CLIcK (Kim et al., 2024a) and HAE-RAE Bench (HAERAE-1.0) (Son et al., 2024), both of which evaluate culturally grounded knowledge, vocabulary, and commonsense reasoning. Finally, to assess Korean STEM and professional knowledge, we report results on KMMLU (Son et al., 2025), an expert-level multi-subject benchmark collected from original Korean examinations that measures broad-domain reasoning and professional knowledge expressed in Korean.

Across all Korean benchmarks, THINK shows a clear advantage over comparable sized open-source vision-language LLMs, such as Qwen3-VL and InternVL. This margin stems from our Korean-aware training recipe, which integrates carefully curated Korean data and training algorithms to enhance performance across all three evaluation dimensions. These results suggest that THINK has internalized Korean linguistic characteristics as well as the social and cultural context required for faithful expression and reasoning.

THINK outperforms EXAONE on most Korean text-to-text benchmarks, with the exception of KoBALT. This achievement is particularly notable given that EXAONE is a pure text-only model, which typically benefits from intrinsic advantages in text understanding without the complexity of multimodal objectives. Despite the added complexity of a multimodal architecture, THINK maintains competitive Korean text reasoning and language understanding.

English Benchmarks. In contrast, on English Text-to-Text benchmarks, THINK shows relatively lower performance than some baseline models. MMLU (Hendrycks et al., 2021) and HellaSwag (Zellers et al., 2019) evaluate broad multi-task reasoning and commonsense-based contextual completion in English. On MMLU, THINK performs competitively with VLM-based models, while showing lower performance than the text-only EXAONE. On HellaSwag, the THINK generally shows lower performance than the comparison models, and it also underperforms on physical commonsense reasoning benchmarks such as PIQA (Bisk et al., 2020). Nonetheless, the performance gap between the THINK and state-of-the-art models of comparable scale remains moderate, and the model maintains practical levels of reasoning stability and response consistency for real-world use.

Translation Benchmarks. We evaluate 1-shot translation performance on the Flores+ benchmark (NLLB Team et al., 2024) using the BLEU metric for the English-Korean pair. For the English-to-Korean direction, we apply Ko-Mecab pre-tokenization to the generated text to compute the scores. As shown in Table 5, THINK achieves the best performance in English-to-Korean translation and the second-best performance in Korean-to-English translation. These results demonstrate that THINK has strong cross-lingual capabilities between Korean and English compared to other baseline models.

5.3 Vision-to-Text Benchmarks

We evaluate Vision-to-Text performance using image-based visual question answering (VQA) benchmarks in both Korean and English. Table 5 summarizes the comparison between THINK and open-source multimodal LLMs of comparable scale. As with Text-to-Text benchmarks, we also use the same parsing rule.

Image Benchmarks. On Korean Vision-to-Text benchmarks, THINK demonstrates consistent overall performance across a majority of tasks. To systematically assess Korean vision-to-text comprehension, we conducted a comparative analysis using the following benchmarks: (1) the Korean national educational examination (Park and Kim, 2025), (2) general vision-language understanding tasks featuring Korean QA derived from English QA (Ju et al., 2024), and (3) Korean document-based VQA (Ju et al., 2024). Specifically, KoNET (Park and Kim, 2025) is a comprehensive multimodal benchmark based on Korean national educational standards, encompassing both the Korean General Educational Development (KGED) tests—covering elementary, middle, and high school levels—and the Korean College Scholastic Ability Test (KCSAT). K-MMBench (Ju et al., 2024) evaluates general vision-language understanding capabilities, including object recognition, attribute reasoning, and commonsense inference, while K-DTCBench (Ju et al., 2024) focuses on document-, table-, and chart-based visual reasoning in Korean. THINK achieves the highest scores among the comparison models on KoNET, indicating strong capabilities in Korean visual understanding and high-difficulty reasoning involving mathematics. On K-MMBench and K-DTCBench, THINK ranks second, with a marginal performance gap compared to the top-performing model. Overall, these results demonstrate that THINK maintains consistent competitiveness on Korean-centric Vision-to-Text tasks across both education-oriented and general-purpose benchmarks.

Subject (Mathematics)	HyperCLOVA X 32B Think	1st-Grade Cut-off (Top 4%)
Probability & Statistics	92	87
Calculus	89	85
Geometry	92	85

Table 7: Performance of HyperCLOVA X 32B Think on the KCSAT 2026 Mathematics exam across elective subjects. The model achieved scores exceeding the estimated 1st-grade (top 4%) cut-offs in all categories, demonstrating top-level visual reasoning capabilities.

We also conduct a comparative analysis on English benchmarks to evaluate diverse visual-language understanding capabilities, including open-ended visual question answering, document understanding, and scene text recognition. We employ four widely adopted benchmarks: SEED-IMG (Li et al., 2024a), LLaVA-W (Liu et al., 2023b), TextVQA (Singh et al., 2019), and DocVQA (Mathew et al., 2021). THINK demonstrates robust text recognition capabilities in both scene-text and dense-document contexts. Notably, THINK achieves the best performance on TextVQA by a significant margin. However, on the remaining English Vision-to-Text benchmarks, THINK exhibits relatively lower performance compared to the baseline models.

KCSAT 2026. To validate the model's real-world reasoning capabilities on the most current academic standards, we evaluate THINK on the recently conducted KCSAT 2026. We specifically focus on Mathematics, as it presents high-difficulty problems that necessitate the seamless integration of textual instructions with complex visual diagrams. To ensure a robust and stable estimation of the model's performance, we employ consensus@64, deriving the final answer through majority voting across 64 independent generation paths. Furthermore, our evaluation follows a hybrid input strategy: image-based inputs are provided for problems where visual context is vital, while textual representation is also used for others to leverage the full capacity of the backbone. As summarized in Table 7, THINK achieves an exceptional score, placing it within the top 4% of human examinees—a performance level equivalent to the highest tier (1st grade) in the Korean grading system. In particular, THINK shows strong performance on geometry, where solving problems often requires precise visual–spatial reasoning over 2D and 3D diagrams. These results demonstrate that the model can robustly perform sophisticated, image-based reasoning on novel and challenging academic materials that demand human-level cognitive synthesis.

Video Benchmarks. While the primary design philosophy of THINK focuses on achieving high-fidelity reasoning within the static visual domain, we extend the architecture to support video input processing to ensure service versatility and meet diverse user needs. Video capability is integrated through a uniform sampling strategy of up to 120 frames, capturing temporal dynamics while maintaining computational efficiency. In evaluations on the Video-MME benchmark (Fu et al., 2025) (conducted without subtitles), THINK achieves a score of 63.4, surpassing established models such as GPT-4V (OpenAI, 2023) (59.9) and our previous iteration, HCX-Video (NAVER Cloud, 2025) (61.4). Although a performance gap exists relative to models specifically optimized for massive video-reasoning benchmarks (e.g., Qwen3-VL-32B-Thinking at 77.3), THINK demonstrates superior robustness in localized, real-world content understanding. Notably, in an internal benchmark assessing NAVER TV Content[4] comprehension, THINK scores 67.1, significantly exceeding GPT-4V's 50.0. These results highlight the model's practical effectiveness in internalizing the temporal and cultural nuances required for Korean-centric AI services.

5.4 Agent Benchmarks

We evaluate agentic capabilities using a collection of agent benchmarks that assess an agent's ability to perform multi-step reasoning, interact with tools, and maintain task coherence over extended interactions. Specifically, we employ the Tau^2 benchmark suite, which consists of *tau*^{2}*-telecom*, *tau*^{2}*-retail*, and *tau*^{2}*-airline*, to evaluate goal completion in interactive multi-turn environments Barres et al. (2025). These benchmarks are designed to test long-horizon planning, state tracking, and consistent decision-making across sequential interactions.

[4] `https://tv.naver.com/`

For SWE-agent–style evaluations that emphasize tool-oriented reasoning in command-line environments, we use TerminalBench Contributors (2025). In this work, TerminalBench 1.0 refers to evaluations conducted using the terminal-bench core implementation version 0.1.1. We additionally report results on TerminalBench Hard, which corresponds to the 47 challenging tasks from the artificial analysis split and is designed to assess robustness to execution errors and recovery from failures.

As shown in Table 5, THINK consistently outperforms open-source baseline models across all evaluated agent benchmarks. On the Tau^2 tasks, our model achieves higher success rates than both EXAONE 4.0 32B and Qwen3-VL 32B-Thinking, indicating stronger agentic reasoning and more reliable execution of multi-step action sequences. The performance gap is particularly pronounced in scenarios that require longer interaction horizons, suggesting improved state tracking and decision consistency.

Similarly, on TerminalBench and TerminalBench Hard, THINK demonstrates clear advantages over the baseline models. These results indicate that our model is more effective at integrating reasoning with tool-driven actions even under challenging evaluation settings.

Table 6 further contextualizes these results by comparing our model with substantially larger commercial and frontier models. While THINK underperforms relative to GPT-5.1, it consistently outperforms Qwen3 235B-A22B across most agent benchmarks. Given the significant difference in model scale, these results highlight a favorable efficiency–performance trade-off and demonstrate strong agentic capabilities relative to model size.

Detailed evaluation protocols, prompting strategies, and success criteria for each benchmark are provided in Appendix C.

5.5 Discussion

Overall, the evaluation highlights the main strengths of THINK and its design characteristics. First, the most prominent strength of THINK is its strong Korean language understanding and reasoning capability. Across Korean Text-to-Text benchmarks, THINK performs on par with or better than the text-only model EXAONE 4.0, despite being a multimodal model. In particular, THINK shows clear advantages on CLIcK and HAERAE Bench, which require understanding of cultural and social context. These results indicate that THINK goes beyond a translation-based approach and effectively internalizes linguistic and cultural characteristics of the Korean language.

Second, THINK also demonstrates competitive performance on Vision-to-Text tasks. On Korean Vision-to-Text benchmarks, THINK consistently outperforms the comparison models. This suggests that aligning visual information on top of a text-centric pre-training foundation can be effectively extended to the Korean language setting. Overall, these results support the practical applicability of THINK as a Korean vision–language reasoning model.

Third, THINK exhibits strong performance efficiency on agent tasks relative to its model scale. Across agent benchmarks, THINK consistently achieves better results than other models of similar size, and in some settings, it shows competitive performance even against large commercial models. This indicates that THINK has effectively learned agent behaviors that are critical in real-world scenarios, such as long-term state tracking, appropriate tool usage, and recovery from errors.

At the same time, the evaluation also reveals clear limitations of THINK. On English-centric Text-to-Text and Vision-to-Text benchmarks, THINK underperforms compared to some baseline models. This can be understood as a natural trade-off that arises when building a global model while first ensuring strong and stable performance in Korean.

6 Conclusion

In this work, we presented HyperCLOVA X 32B Think, a vision-language model trained with an emphasis on reasoning within the context of the Korean language and culture, as well as agentic ability. Experiments against comparably sized models demonstrate that HyperCLOVA X 32B Think is competitive on text-to-text (e.g. KoBALT, CLIcK, HAERAE-1.0) and vision-to-text (KoNET and K-MMBench) benchmarks in Korean, as well as agent (Tau^2 and Terminal Bench) benchmarks.

Given such performance, we expect the public release of HyperCLOVA X 32B Think to benefit both the academic and industrial partners.

THINK is built on a strong text-only backbone that is subsequently extended to vision understanding. This sequential training paradigm enables rapid acquisition of multimodal capabilities by reusing the extensive knowledge learned during text-based pre-training. As a result, competitive vision understanding performance can be achieved with relatively little additional training. However, we observe that as multimodal training progresses, performance on text-based benchmarks gradually degrades, consistent with previous reports on catastrophic forgetting in multimodal large language models (Zhai et al., 2023; Driess et al., 2023; Liu et al., 2025b). More broadly, this highlights a structural limitation of strictly sequential modality expansion, in which modalities are introduced one after another rather than learned jointly. It becomes increasingly challenging to preserve strong text-based reasoning performance while scaling multimodal capabilities.

In light of this, we do not rely strictly on sequential training in our subsequent model HyperCLOVA X 8B Omni. Instead, we adopt a joint training strategy in which text, vision, and audio modalities are learned together from early stages of training. This design mitigates catastrophic forgetting caused by modality extension. Please see the HyperCLOVA X 8B Omni report for more details on the omnimodal training strategy.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-VL Technical Report.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint*.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. Efficient training of language models to fill in the middle. *arXiv preprint arXiv:2207.14255*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*.

Woojin Chung and Jeonghoon Kim. 2025. Exploiting vocabulary frequency imbalance in language model pre-training. *CoRR*, abs/2508.15390.

Terminal-Bench Contributors. 2025. Terminal-bench: An evaluation framework for terminal-based agent tasks. `https://www.tbench.ai/`. Accessed: 2025-12.

Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2025. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. 2025. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. In *CVPR*.

Leandro von Werra Garreth Lee, Guilherme Penedo and Thomas Wolf. 2024. From digits to decisions: How tokenization impacts arithmetic in llms.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, Jingji Chen, Jingjia Huang, Kang Lei, Liping Yuan, Lishu Luo, Pengfei Liu, Qinghao Ye, Rui Qian, Shen Yan, Shixiong Zhao, Shuai Peng, Shuangye Li, Sihang Yuan, Sijin Wu, Tianheng Cheng, Weiwei Liu, Wenqian Wang, Xianhan Zeng, Xiao Liu, Xiaobo Qin, Xiaohan Ding, Xiaojun Xiao, Xiaoying Zhang, Xuanwei Zhang, Xuehan Xiong, Yanghua Peng, Yangrui Chen, Yanwei Li, Yanxu Hu, Yi Lin, Yiyuan Hu, Yiyuan Zhang, Youbin Wu, Yu Li, Yudong Liu, Yue Ling, Yujia Qin, Zanbo Wang, Zhiwu He, Aoxue Zhang, Bairen Yi, Bencheng Liao, Can Huang, Can Zhang, Chaorui Deng, Chaoyi Deng, Cheng Lin, Cheng Yuan, Chenggang Li, Chenhui Gou, Chenwei Lou, Chengzhi Wei, Chundian Liu, Chunyuan Li, Deyao Zhu, Donghong Zhong, Feng Li, Feng Zhang, Gang Wu, Guodong Li, Guohong Xiao, Haibin Lin, Haihua Yang, Haoming Wang, Heng Ji, Hongxiang Hao, Hui Shen, Huixia Li, Jiahao Li, Jialong Wu, Jianhua Zhu, Jianpeng Jiao, Jiashi Feng, Jiaze Chen, Jianhui Duan, Jihao Liu, Jin Zeng, Jingqun Tang, Jingyu Sun, Joya Chen, Jun Long, Junda Feng, Junfeng Zhan, Junjie Fang, Junting Lu, Kai Hua, Kai Liu, Kai Shen, Kaiyuan Zhang, Ke Shen, Ke Wang, Keyu Pan, Kun Zhang, Kunchang Li, Lanxin Li, Lei Li, Lei Shi, Li Han, Liang Xiang, Liangqiang Chen, Lin Chen, Lin Li, Lin Yan, Liying Chi, Longxiang Liu, Mengfei Du, Mingxuan Wang, Ningxin Pan, Peibin Chen, Pengfei Chen, Pengfei Wu, Qingqing Yuan, Qingyao Shuai, Qiuyan Tao, Renjie Zheng, Renrui Zhang, Ru Zhang, Rui Wang, Rui Yang, Rui Zhao, Shaoqiang Xu, Shihao Liang, Shipeng Yan, Shu Zhong, Shuaishuai Cao, Shuangzhi Wu, Shufan Liu, Shuhan Chang, Songhua Cai, Tenglong Ao, Tianhao Yang, Tingting Zhang, Wanjun Zhong, Wei Jia, Wei Weng, Weihao Yu, Wenhao Huang, Wenjia Zhu, Wenli Yang, Wenzhi Wang, Xiang Long, XiangRui Yin, Xiao Li, Xiaolei Zhu, Xiaoying Jia, Xijin Zhang, Xin Liu, Xinchen Zhang, Xinyu Yang, Xiongcai Luo, Xiuli Chen, Xuantong Zhong, Xuefeng Xiao, Xujing Li, Yan Wu, Yawei Wen, Yifan Du, Yihao Zhang, Yining Ye, Yonghui Wu, Yu Liu, Yu Yue, Yufeng Zhou, Yufeng Yuan, Yuhang Xu, Yuhong Yang, Yun Zhang, Yunhao Fang, Yuntao Li, Yurui Ren, Yuwen Xiong, Zehua Hong, Zehua Wang, Zewei Sun, Zeyu Wang, Zhao Cai, Zhaoyue Zha, Zhecheng An, Zhehui Zhao, Zhengzhuo Xu, Zhipeng Chen, Zhiyong Wu, Zhuofan Zheng, Zihao Wang, Zilong Huang, Ziyu Zhu, and Zuquan Song. 2025. Seed1.5-VL Technical Report.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance?

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

NAVER Cloud HyperCLOVA X Team. 2025. Hyperclova x think technical report.

Hwiyeol Jo, Joosung Lee, Jaehone Lee, Sang-Woo Lee, Joonsuk Park, and Kang Min Yoo. 2025. Finding answers in thought matters: Revisiting evaluation on large language models with reasoning.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

Jeongho Ju, Daeyoung Kim, SunYoung Park, and Youngjune Kim. 2024. Varco-vision: Expanding frontiers in korean vision-language models.

Eunsu Kim, Juyoung Suk, Philhoon Oh, Haneul Yoo, James Thorne, and Alice Oh. 2024a. CLIcK: A benchmark dataset of cultural and linguistic intelligence in Korean. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3335–3346, Torino, Italia. ELRA and ICCL.

Seungduk Kim, Seungtaek Choi, and Myeongho Jeong. 2024b. Efficient and effective vocabulary expansion towards multilingual large language models. *arXiv preprint arXiv:2402.14714*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

LG AI Research. 2025. Exaone 4.0: Unified large language models integrating non-reasoning and reasoning modes. *arXiv preprint arXiv:2507.11407*.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024a. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13299–13308.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee F. Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah M. Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Raghavi Chandu, Thao Nguyen, Igor Vasiljevic, Sham M. Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alex Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. 2024b. Datacomp-lm: In search of the next generation of training sets for language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. 2025a. Superbpe: Space travel for language models. *CoRR*, abs/2503.13423.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *NeurIPS*.

Wenzhuo Liu, Fei Zhu, Haiyang Guo, Longhui Wei, and Cheng-Lin Liu. 2025b. Llava-c: Continual improved visual instruction tuning.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models. *arXiv preprint arXiv:2405.05374*.

NAVER Cloud. 2025. HyperCLOVA X Video: Seeing through motion. `https://clova.ai/en/tech-blog/hyperclova-x-video-seeing-through-motion`. Accessed: 2025-12-31.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Jeffrey Olmo, Jared Wilson, Max Forsey, Bryce Hepner, Thomas Vincent Howe, and David Wingate. 2025. Features that make a difference: Leveraging gradients for improved dictionary learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7609–7619, Albuquerque, New Mexico. Association for Computational Linguistics.

OpenAI. 2022. tiktoken: A fast BPE tokeniser for use with OpenAI's models. `https://github.com/openai/tiktoken`. Accessed: 2025-12-31.

OpenAI. 2023. GPT-4V(ision) System Card. System card, OpenAI.

OpenAI. 2025. Gpt-5 system card. `https://cdn.openai.com/gpt-5-system-card.pdf`.

Sanghee Park and Geewook Kim. 2025. Evaluating multimodal generative AI with Korean educational standards. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 671–688, Albuquerque, New Mexico. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. 2024a. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Guilherme Penedo, Hynek Kydlíček, Alessandro Cappelli, Mario Sasko, and Thomas Wolf. 2024b. Datatrove: large scale data processing.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language.

Thiziri Nait Saada, Louis Bethune, Michal Klein, David Grangier, Marco Cuturi, and Pierre Ablin. 2025. The data-quality illusion: Rethinking classifier-based quality filtering for llm pretraining. *arXiv preprint arXiv:2510.00866*.

Craig W. Schmidt, Varshini Reddy, Chris Tanner, and Yuval Pinter. 2025. Boundless byte pair encoding: Breaking the pre-tokenization barrier. *CoRR*, abs/2504.00178.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024a. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Noam Shazeer. 2020. Glu variants improve transformer.

Hyopil Shin, Sangah Lee, Dongjun Jang, Wooseok Song, Jaeyoon Kim, Chaeyoung Oh, Hyemi Jo, Youngchae Ahn, Sihyun Oh, Hyohyeong Chang, Sunkyoung Kim, and Jinsik Lee. 2025. Kobalt: Korean benchmark for advanced linguistic tasks.

Anya Sims, Cong Lu, Klara Kaleb, Jakob Nicolaus Foerster, and Yee Whye Teh. 2025. Stochastok: Improving fine-grained subword understanding in llms. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. KMMLU: Measuring massive multitask language understanding in Korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104, Albuquerque, New Mexico. Association for Computational Linguistics.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jae cheol Lee, Je Won Yeom, Jihyu Jung, Jung woo Kim, and Songseong Kim. 2024. HAE-RAE bench: Evaluation of Korean knowledge in language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7993–8007, Torino, Italia. ELRA and ICCL.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Liang Wang, Nan Yang, Shaohan Huang, Li Dong, and Furu Wei. 2025a. Thinking augmented pre-training.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. 2025b. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Anna Wegmann, Dong Nguyen, and David Jurgens. 2025. Tokenization is sensitive to language variation. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 10958–10983. Association for Computational Linguistics.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, et al. 2025. Bapo: Stabilizing off-policy reinforcement learning for llms via balanced policy optimization with adaptive clipping. *arXiv preprint arXiv:2510.18927*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. 2025. Dapo: An open-source llm reinforcement learning system at scale.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In *Conference on Parsimony and Learning (Proceedings Track)*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in neural information processing systems*, 32.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Contributions and Acknowledgments

*Within each role, **names are listed in alphabetical order** by first name, followed by the last name.*

Technical Writing

Cheonbok Park
Dongyoon Han
Geewook Kim
Hwiyeol Jo
Jaehong Lee
Jeonghoon Kim
Jin-Hwa Kim
Joonsuk Park
Joosung Lee
Sangdoo Yun
Sanghyuk Choi
Sungwook Jeon
Taeho Kil
Yoonsik Kim

Model Research and Training

Bado Lee
Byeongwook Kim
Byungwook Lee
Cheonbok Park
Chiheon Ham
Daehee Kim
Donghyeon Ko
DongJu Lee
Geewook Kim
Gichang Lee
Hanbyul Kim
Hangyeol Yu
Hodong Lee
Hyunsoo Ha
Jaehong Lee
Jaemin Han
Jeonghoon Kim
Jinbae Im
Jingu Kang
Jinhyeon Kim
Jisung Wang
Juncheol Kim
Jungwhan Kim
Ka Yeon Song
Kyeongseok Jeong
Min Young Lee
Moonbin Yim
Nako Sung
Ohsung Kwon
Sang Hee Park
Sanghyuk Choi
Seongjin Shin
Shinyoung Joo
Soobee Lee
Sookyo In
Soyoon Kim
Suk Min Seo
Sungju Kim
Taeho Kil
Taehong Min
Taehwan Yoo
Yeontaek Oh
Yoonsik Kim
You Jin Kim
Youngki Hong

Model Evaluation and Analysis

Chansong Jo
Chiheon Ham
Daehee Kim
Dagyeong Lee
DongJu Lee
Gayoung Lee
Hagyeong Lee
Hangyeol Yu
Hwiyeol Jo
Hyunhoon Jung
Injae Lee
Jaemin Han
Jeonghyun Lee
Jieun Lee
Jieun Shin
Jingu Kang
Jonghak Kim
Joosung Lee
Jungwhan Kim
Ka Yeon Song
Kiyoon Moon
Min Young Lee
Minchul Song
Minkyoung Kim
Minseong Choi
MinYoung Kim
Munhyong Kim
MyungIn You
Saerim Cho
Shinyoung Joo
Soobee Lee
Suk Min Seo
Taeyong Kim
Yeguk Jin
Yehbin Lee
Yelim Jeong
Yeontaek Oh
Yesol Lee
You Jin Kim
Youngjin Kwon
Youngki Hong

Data

Chansong Jo
Donghyeon Ko
Hanbyul Kim
Hyunsoo Ha
Injae Lee
Jieun Lee
Jinbae Im
Jisung Wang
Juncheol Kim
Kiyoon Moon
Kyeongseok Jeong
Minkyoung Kim
Minseong Choi
Moonbin Yim
Munhyong Kim
MyungIn You
Ohsung Kwon
Seongjin Shin
Sookyo In
Soyoon Kim
Sung Ae Lee
Sungju Kim
Taehong Min
Taehwan Yoo
Taemin Lim
Taeyong Kim
Woobin Choi
Yeguk Jin
Yehbin Lee
Youngjun Kim
Zoo Hyun Lee

Model Serving and Inference

Bong-Jin Lee
Chankyu Lee
Han-Gyu Kim
Hanbae Seo
Hodong Lee
Hyunjoon Jeong
Jaeeun Kil
Jaegwang Lee
Jeongtae Lee
Jinhyeon Kim
Joonghoon Kim
Junga Choi
Junhee Yoo
Lukas Lee
Minjung Jo
Minsub Kim
Myungwoo Oh
Ohhyeok Kwon
Sang Hee Park
Seungyeol Lee
Sungjae Lee
Youngki Kwon

Model Planning

Dagyeong Lee
Eunchul Kim
Gayoung Lee
Hagyeong Lee
Hyunhoon Jung
Jeonghyun Lee
Jieun Shin
Jonghak Kim
JongHyun Lee
Matt Yeo
Minchul Song
MinYoung Kim
Saerim Cho
Yelim Jeong
Yesol Lee
Youngjin Kwon

Business and Brand Strategy

Dukmin Jung
Kyungmin Lee
Hyojin Park
Sujin Roh
Misuk Park

Residency Program

Bumkyu Park
Byung Hyun Lee
Doohyuk Jang
Geeho Kim
Hyewon Jeon
Hyunbin Jin
Hyungwook Choi
Ijun Jang
Inju Ha
Jewon Yeom
Jihwan Kim
Jihwan Kwak
Joonki Min
Juan Yeo
Junbeom Kim
Junyeob Kim
Kunhee Kim
Kyubyung Chae
Kyudan Jung
Minha Jhang
Sangyoon Lee
Sehyun Lee
Seunghee Kim
Song-ha Jo
Suho Ryu
Yokyung Lee

Internships

Dong-Jae Lee
Jihwan Moon
Jinho Heo
Jisu Jeon
Junseo Jang
Minsik Choi
Seulbi Lee
Singon Kim
Sumin Cho
Woojin Chung
Yunjae Won

Appendix

A Chat Templates

A.1 Non-Reasoning

```
<|im_start|>user
{query}<|im_end|>
<|im_start|>assistant
<think>

</think>

{response}
<|im_end|>
```

A.2 Reasoning

```
<|im_start|>user
{query}<|im_end|>
<|im_start|>assistant
<think>
{reasoning_content}
</think>

{response}
<|im_end|>
```

A.3 Agent

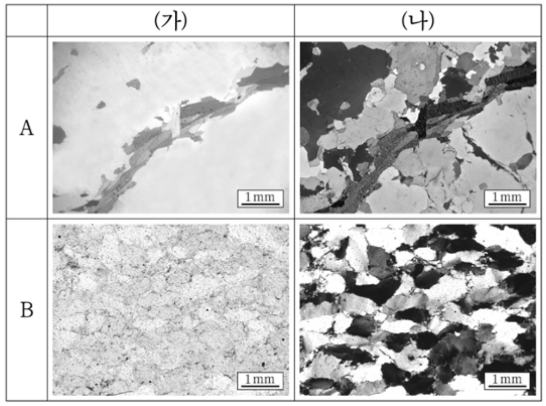
```
<|im_start|>system
# Tools

You may call one or more functions to assist with the user
query.

You are provided with function signatures within <tools>
</tools> XML tags:
<tools>
{"type": "function", "function": {"name": "FUNCTION_NAME_1",
"description": "DESCRIPTION_1", "parameters": { ... } } }
{"type": "function", "function": {"name": "FUNCTION_NAME_2",
"description": "DESCRIPTION_2", "parameters": { ... } } }
...
</tools>

For each function call, output the function name and arguments
within the following XML format:
<tool_call>{function-name}
<arg_key>{arg-key-1}</arg_key>
<arg_value>{arg-value-1}</arg_value>
<arg_key>{arg-key-2}</arg_key>
<arg_value>{arg-value-2}</arg_value>
...
</tool_call><|im_end|>
<|im_start|>user
test<|im_end|>
<|im_start|>assistant
<think>
{reasoning_content}
```

```
</think>

{response}
<|im_end|>
```

A.4 Multimodal

```
<|im_start|>user
{multimodal embedding}\n{query}<|im_end|>
<|im_start|>assistant
<think>
{reasoning_content}
</think>

{response}
<|im_end|>
```

15. 표는 편광 현미경으로 관찰한 암석 A와 B의 박편 사진을 나타낸 것이다. A와 B는 각각 규암과 편마암 중 하나이고, (가)와 (나)는 각각 개방 니콜과 직교 니콜 중 하나이다.



이에 대한 설명으로 옳은 것만을 <보기>에서 있는 대로 고른 것은?

<보 기>

ㄱ. (가)는 개방 니콜이다.
ㄴ. A는 편마암이다.
ㄷ. B에서 재결정 작용을 받은 광물이 관찰된다.

① ㄱ ② ㄷ ③ ㄱ, ㄴ ④ ㄴ, ㄷ ⑤ ㄱ, ㄴ, ㄷ

Figure 3: Multimodal Example.

B Examples

B.1 Vision–Language QA Example

Model response

HyperCLOVA X THINK (Translated Model Output – English)

Okay, let's solve this problem. First, we need to carefully read the question. Thin-section photographs of rocks A and B observed under a polarizing microscope are given, and A and B are each either quartzite or gneiss. In addition, (a) and (b) correspond to either plane-polarized light or cross-polarized light. The task is to choose the correct statements from the options.
First, let's recall the difference between plane-polarized light and cross-polarized light in a polarizing microscope. Plane-polarized light refers to the state where only the lower polarizer is present, whereas cross-polarized light refers to the state where the upper and lower polarizers are oriented perpendicular to each other. Under cross-polarized light, anisotropic minerals display interference colors, while under plane-polarized light, extinction and general optical properties of minerals can be observed.
Now, looking at the photographs, we need to determine which of (a) and (b) corresponds to plane-polarized light and which to cross-polarized light. Examining the images of A and B, the image of A in (a) appears somewhat dull, with mineral shapes looking indistinct. Under cross-polarized light, minerals usually appear brighter due to interference colors, but (a) looks relatively dark and unclear. In contrast, (b) shows minerals more distinctly, with visible banding and patterns. Since anisotropic minerals exhibit interference colors under cross-polarized light, (b) is more likely to be cross-polarized light. Accordingly, (a) would be plane-polarized light. Therefore, statement ㄱ seems to be correct.
Next, we consider which of A and B corresponds to quartzite or gneiss. Gneiss is a metamorphic rock characterized by gneissic banding, where minerals are arranged in distinct layers or

stripes. In the (b) image of A, minerals are clearly arranged in bands, which is characteristic of gneiss. Thus, A is most likely gneiss. On the other hand, B shows a more uniform distribution of minerals in both (a) and (b), without clear banding. Quartzite is composed mainly of quartz and forms through metamorphism accompanied by recrystallization, resulting in uniformly distributed mineral grains. Therefore, B is likely quartzite. Since statement ㄴ claims that A is gneiss, it also appears to be correct.
Finally, statement ㄷ states that minerals that have undergone recrystallization are observed in B. Quartzite forms through recrystallization during metamorphism, so its mineral grains are typically well-developed and uniformly distributed. In the images of B, the minerals appear evenly distributed with clearly developed grains, indicating recrystallization. Therefore, statement ㄷ is also correct.
In summary, statements ㄱ, ㄴ, and ㄷ all appear to be correct. Among the answer choices, option 5 includes all three statements, so the correct answer seems to be 5. Still, it is worth double-checking.
If A is gneiss, the gneissic banding should be visible in the (b) image, and indeed the minerals are arranged in stripes. B is quartzite, showing uniform mineral distribution and evidence of recrystallization, so ㄷ is correct. Since (a) corresponds to plane-polarized light and (b) to cross-polarized light, ㄱ is also correct. Therefore, all statements are correct, and the answer is 5. </think>
$\boxed{5}$

C Agent Evaluation

This appendix describes the evaluation protocols, benchmark configurations, and execution settings used for agentic capability assessment. We evaluate models using interactive and tool-oriented agent benchmarks, following standardized protocols to ensure reproducibility and fair comparison.

C.1 Tau^2 Benchmark Evaluation

We evaluate interactive agentic reasoning using the Tau^2 benchmark suite Barres et al. (2025), which is designed to assess long-horizon decision-making, state tracking, and goal completion in multi-turn environments. The suite consists of three domains: *tau2-telecom*, *tau2-retail*, and *tau2-airline*, each containing multiple user scenarios with predefined success criteria.

Evaluation Protocol. For each task instance, the agent interacts with the environment in a turn-based manner until one of the following conditions is met: the task success condition is satisfied, an explicit failure condition is triggered, or a maximum turn limit is reached. The final outcome is recorded as a binary success or failure signal.

Configurable Evaluation Settings. Table 8 summarizes the primary evaluation parameters that can be configured for the Tau^2 benchmark. For sampling parameters not listed in Table 8, the default settings of each model were used. All models were evaluated in reasoning mode.

Table 8: Evaluation settings for the Tau^2 benchmark suite Barres et al. (2025).

Parameter	Value	Description
Max turns	200	Maximum number of interaction turns allowed per episode
Temperature	0.6	Sampling temperature used for action generation

C.2 TerminalBench Evaluation

We evaluate tool-oriented agent performance using TerminalBench Contributors (2025), which measures an agent's ability to solve command-line tasks through iterative tool use and execution feedback. TerminalBench focuses on precise command generation, correct tool usage, and recovery from execution errors.

Benchmarks. We report results on TerminalBench 1.0 and TerminalBench Hard. TerminalBench 1.0 corresponds to evaluations conducted using the terminal-bench core implementation version 0.1.1. TerminalBench Hard consists of a subset of 47 challenging tasks selected according to the difficulty definition used by Artificial Analysis, and is intended to stress-test agent robustness in terminal-based execution scenarios[5].

Evaluation Protocol. Each task provides an initial terminal state and a natural language goal description. The agent generates shell commands iteratively, which are executed in a sandboxed environment. After each execution, the agent observes command outputs and error messages. A task is considered successful only if the final system state satisfies all predefined assertions. To ensure fair comparison, all evaluated models utilized the same Terminus-2 agent implementation with reasoning enabled throughout the evaluation.

C.3 Metrics and Reporting

For both Tau^2 and TerminalBench, the primary evaluation metric is task success rate, defined as the percentage of tasks completed successfully. All results are averaged across task instances within each benchmark unless otherwise specified. Additional diagnostic metrics, such as average interaction length and failure frequency, may be reported to provide further insight into agent behavior.

[5] `https://artificialanalysis.ai/methodology/intelligence-benchmarking#terminal-bench-hard`