A. Baselines

KVQA with Knowledge Graphs and Retrieval. We select representative state-of-the-art approaches, including direct question-only answering (Q Only) [14], BAN [11], MUTAN [3], ConceptBERT [7], KRISP [15], MAVEx [21], VLCBERT [18], HCNMN [26], and MCAN [25]. Since BAN and MUTAN are limited to learning unimodal visual features, we enhance them with ArticleNet (AN) [14], which retrieves relevant information from Wikipedia based on the given question–image pair to support external knowledge reasoning. These enhanced versions are referred to as "BAN + AN" and "MUTAN + AN" [14].

KVQA with LLMs / MLLMs. We employ PICa [23], KAT [8], and REVIVE [12]. The results of *KVQA with Knowledge Graphs and Retrieval* as well as *KVQA with Large Language Models* are from prior work [5], where the exact same experimental setup and evaluation protocols are adopted.

IK-KVQA with Multimodal Large Language Models. We employed three types of Multimodal Large Language Models (MLLMs):

- **Advanced open-source MLLMs**: Including three regular-sized models: Qwen2.5-VL-7B [2], Llama-3.2-11B-Vision [6], and Gemma-3-12B [10]; as well as three larger and more advanced models: Gemma-3-27B [10], Qwen2.5-VL-72B [2], and InternVL3-78B [27]. All of them are **instruction-tuned versions**.
- **Proprietary state-of-the-art MLLMs**: Including two of Google's most advanced models, Gemini 2.5 Flash and Gemini 2.5 Pro [4], as well as OpenAI's flagship multimodal model, GPT-4o [9]. Both Gemini 2.5 Flash and Gemini 2.5 Pro perform inference in the Dynamic Thinking mode.
- **Augmented MLLMs**:
  - **Supervised fine-tuning (SFT)** [17] is a crucial process that trains a pre-trained MLLM on a high-quality dataset of instructions and responses, making it more effective at following specific commands and performing user-facing tasks. The MLLM backbone is Qwen2.5-VL-7B.
  - **Chain of Thought (CoT)** [20] is a prompting technique that improves the reasoning abilities of large language models by guiding them to break down a complex problem into a series of intermediate steps before providing a final answer. The MLLM backbone is Qwen2.5-VL-7B.
  - **CoT + SFT** is a well-optimized CoT-prompted SFT baseline.
  - **LLaVA-CoT** [22], a new multimodal model that uses a chain-of-thought method to improve vision-language models' ability to reason step-by-step.
  - **M2-Reasoning** (7B) [1] is a multimodal large language model (MLLM) that achieves state-of-the-art (SOTA) performance in both general and spatial reasoning by using a high-quality data pipeline and a dynamic multi-task training strategy.
  - **Self-Distillation Fine-Tuning (SDFT)** [24] rewrites task responses into its own style and fine-tunes on them to reduce distribution shift and forgetting. The MLLM backbone is Qwen2.5-VL-7B.

A.1. Implementation Details.

Our approach StaR-KVQA has been implemented using PyTorch 2.7.0 as well as Python 3.10, and all experiments have been conducted on the NVIDIA L20 GPU. During training, the batch size (with accumulation) is set to 16, the learning rate is $1e{-}4$, the LoRA rank is 32, the LoRA alpha is 64, the traininig epoch is 3. In the OK-VQA dataset, K is set as 3, and in the FVQA dataset, K is set as 4.

We adhere to the established evaluation setting and fix the random seed to `42` throughout data loading, parameter initialization, and decoding. Consistent with prior work [5], we report single-run results in the main tables to maintain strict comparability with published baselines. We did not sweep over seeds or report standard deviations; we view multi-seed evaluation as complementary and leave it to future extensions or large-scale replication studies.

To ensure a level playing field across closed- and open-source models, we (i) supply only the image and the question as inputs, without chain-of-thought or auxiliary prompts; and (ii) adopt each model's *default* inference hyperparameters (decoding temperature and maximum generation length), avoiding any model-specific tuning. This protocol matches the default settings recommended by the model providers and prevents gains from hyperparameter overfitting.

B. Metric

For the open-ended task, *i.e.*, direct answer (DA) setting, we evaluate generated answers using the following accuracy definition:

$$\text{Accuracy} = \min\left(\frac{\#\text{humans that provided that answer}}{3}, 1\right) \tag{1}$$

i.e., an answer is considered fully correct (100% accuracy) if it matches the responses of at least three annotators. Before comparison, all responses are normalized by lowercase, converting numbers to digits, and removing punctuation and articles. We deliberately avoid soft similarity measures such as Word2Vec [16], which may incorrectly cluster semantically distinct words (e.g., "left" vs. "right"). Likewise, we exclude machine translation metrics such as BLEU and ROUGE, as they are mainly suited for multi-word sentence evaluation rather than short answers typically found in VQA.

C. Theoretical Notes for StaR-KVQA

This appendix offers compact analyses that formalize how (i) typed, path-grounded traces (planner + *reasoning composer*), (ii) the single-model selector, and (iii) single-model self-distillation contribute to StaR-KVQA. The statements are backbone-agnostic and match the components introduced in Sec 4.

C.1. Notation and Standing Assumptions

Let $(I, Q, a^\star) \sim \mathcal{D}$ denote image, question, and ground-truth answer. A *trace* is $T = (P_t, P_v, C)$. Our model with parameters θ induces

$$p_\theta(T, a \mid I, Q) = p_\theta(P_t, P_v \mid I, Q)\, p_\theta(C \mid I, Q, P_t, P_v)\, p_\theta(a \mid I, Q, T). \tag{2}$$

We reuse two structural predicates from Sec. 4.2:

$$\mathrm{Cover}(C; P_t, P_v) \geq \kappa, \qquad \mathrm{Vis}(C; I) \geq \rho, \tag{3}$$

encoding path–sentence coverage and visual attestability. Define the feasible set $\mathcal{T}_{\kappa,\rho} = \{T : \mathrm{Cover} \geq \kappa,\ \mathrm{Vis} \geq \rho\}$.

C.2. Generalization Benefit from Typed and Verifiable Traces

We compare an *answer-only* class with a *trace-constrained* class that must produce $T \in \mathcal{T}_{\kappa,\rho}$ alongside a.

Hypothesis classes. Let $\mathcal{H}_{\mathrm{ans}} = \{h : (I, Q) \mapsto a\}$ and

$$\mathcal{H}_{\mathrm{trace}} = \{h : (I, Q) \mapsto (T, a) \text{ s.t. } T \in \mathcal{T}_{\kappa,\rho}\}. \tag{4}$$

Both are realized by the *same* architecture but trained with different supervision.

Theorem 1 (Rademacher shrinkage via verifiable structure)**.** *Assume bounded losses $\ell(a, a^\star) \in [0, 1]$ and $\ell_{\mathrm{trace}}(T, a; a^\star) \in [0, 1]$ with $\ell_{\mathrm{trace}}(T, a; a^\star) \geq \ell(a, a^\star)$ and equality whenever $T \in \mathcal{T}_{\kappa,\rho}$. Then for any sample size N and $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$\mathcal{R}_{\mathcal{D}}(h_{\mathrm{trace}}) \leq \widehat{\mathcal{R}}_N(h_{\mathrm{trace}}) + 2\,\mathfrak{R}_N(\mathcal{H}_{\mathrm{trace}}) + \sqrt{\tfrac{\ln(1/\delta)}{2N}}, \tag{5}$$

and moreover $\mathfrak{R}_N(\mathcal{H}_{\mathrm{trace}}) \leq \mathfrak{R}_N(\mathcal{H}_{\mathrm{ans}}) \cdot \sqrt{\Pi(\mathcal{T}_{\kappa,\rho})/\Pi(\mathcal{T})}$, where $\mathfrak{R}_N(\cdot)$ is the empirical Rademacher complexity and $\Pi(\cdot)$ the growth function.

Intuition. Enforcing typed, verifiable traces prunes implausible labelings (fewer admissible traces per example), which lowers the effective complexity term and tightens the bound. *Practical takeaway.* Structure acts as an inductive bias without changing the backbone.

C.3. Selector as Maximum Likelihood under a Consistency-Noise Model

Our best-triplet selector uses the *single-model* setup to score candidates. The score can be interpreted as a log-likelihood under a simple noise model.

Model. For candidate b, define binary indicators $Y_b^{(\mathrm{ans})}, Y_b^{(\mathrm{ent})}, Y_b^{(\mathrm{align})}, Y_b^{(\mathrm{coh})} \in \{0, 1\}$ for answer correctness, explanation⇒answer entailment, path→explanation alignment, and explanation coherence. Assume conditional independence given a latent quality q_b:

$$\Pr(Y_b^{(j)} = 1 \mid q_b) = \sigma(w_j q_b), \qquad j \in \{\text{ans, ent, align, coh}\}, \tag{6}$$

with logistic σ and weights $w_j > 0$. Let $\hat{y}_b^{(j)} \in [0, 1]$ be soft proxies estimated by the model; the log-likelihood is $\log L_b(q_b) = \sum_j \hat{y}_b^{(j)} \log \sigma(w_j q_b) + (1 - \hat{y}_b^{(j)}) \log(1 - \sigma(w_j q_b))$.

Proposition 2 (Selector equals MLE/MAP ranking)**.** *The maximizer $\hat{q}_b = \arg\max_q \log L_b(q)$ is monotone in $s_\phi(b) := \sum_j w_j (2\hat{y}_b^{(j)} - 1)$. Therefore selecting $b^\star = \arg\max_b s_\phi(b)$ agrees with MLE (and with MAP under any log-concave prior).*

Intuition. The weighted consistency cues act like independent "votes." A larger weighted sum implies a larger MLE quality and thus a higher rank. *Practical takeaway.* Our LLM-as-a-judge ranking matches likelihood-based selection under a reasonable noise model.

C.4. Single-Model Self-Distillation Reduces Supervision{Generation Shift

Let P be the generator distribution over traces (from MLLM_ϕ) and Q_θ the student's distribution after fine-tuning. Let $\mathcal{L} \in [0, 1]$ be a bounded loss on completions.

Lemma 1 (Risk gap upper bounded by divergence)**.** *For any (I, Q),*

$$\left| \mathbb{E}_{T \sim P} \mathcal{L}(T) - \mathbb{E}_{T \sim Q_\theta} \mathcal{L}(T) \right| \leq \sqrt{2\,\mathrm{KL}(P \,\|\, Q_\theta)}. \tag{7}$$

Proof. By total variation (TV) and Pinsker's inequality: $|\,\mathbb{E}_P f - \mathbb{E}_Q f\,| \leq 2\,\mathrm{TV}(P, Q)$ for $f \in [0, 1]$, and $\mathrm{TV}(P, Q) \leq \sqrt{\frac{1}{2}\mathrm{KL}(P \| Q)}$. Combining gives the stated bound. □

Theorem 3 (Self-distillation alignment)**.** *If fine-tuning reduces $\mathrm{KL}(P \| Q_\theta)$ on augmented traces (i.e., the student learns from traces in the generator's style), the supervision–generation risk gap is $O(\sqrt{\mathrm{KL}(P \| Q_\theta)})$ by Lemma 1. Using a* single-model *setup (shared format/tokenization) typically attains a smaller KL than heterogeneous teachers.*

Intuition. Learning from "in-style" traces narrows the distribution gap, which directly controls the risk gap. *Practical takeaway.* Single-model self-distillation stabilizes training and mitigates forgetting.

C.5. Training Objective as a Joint-Likelihood Lower Bound

Our loss in Sec. 4 supervises (P_t, P_v), C, and a. It can be seen as maximizing a lower bound on $\log p_\theta(a^\star \mid I, Q)$ marginalized over feasible traces.

Proposition 4 (ELBO-style lower bound with feasible traces)**.** *Let $\mathcal{T}_{\kappa,\rho}$ be the feasible set. For any auxiliary distribution $q(T \mid I, Q)$ supported on $\mathcal{T}_{\kappa,\rho}$,*

$$\begin{aligned}\log p_\theta(a^\star \mid I, Q) \;\geq\; & \underbrace{\mathbb{E}_q[\log p_\theta(P_t, P_v \mid I, Q)]}_{\textit{path term}} \\ & + \underbrace{\mathbb{E}_q[\log p_\theta(C \mid I, Q, P_t, P_v)]}_{\textit{explanation term}} \\ & + \underbrace{\mathbb{E}_q[\log p_\theta(a^\star \mid I, Q, T)]}_{\textit{answer term}} \\ & - \mathrm{KL}(q(T \mid I, Q) \,\|\, p_\theta(T \mid I, Q, a^\star))\,. \end{aligned} \tag{8}$$

Proof. Write $\log p_\theta(a^\star \mid I, Q) = \log \sum_{T \in \mathcal{T}_{\kappa,\rho}} p_\theta(T, a^\star \mid I, Q)$, insert $q(T \mid I, Q)$, and apply Jensen:

$$\log \sum_T q(T) \frac{p_\theta(T, a^\star)}{q(T)} \geq \mathbb{E}_q\big[\log p_\theta(T, a^\star) - \log q(T)\big].$$

Factorize $p_\theta(T, a^\star)$ using the model and rearrange. □

Intuition. Supervising paths, explanations, and answers maximizes a tractable surrogate of the marginal likelihood; better selection of q (stronger traces) tightens the bound. *Practical takeaway.* Improving the selector/feasibility checks translates into better training signals.

C.6. Putting Pieces Together

Theorems 1–3 and Prop. 4 jointly suggest: (i) typed, verifiable traces reduce effective hypothesis space; (ii) the single-model selector is equivalent to MLE/MAP under a simple consistency–noise view; (iii) single-model self-distillation reduces supervision–generation shift; and (iv) the training objective maximizes a joint-likelihood lower bound whose tightness benefits from stronger traces and selection.

D. Use of Large Language Models

In preparing this article, Large Language Models (LLMs) were employed only for stylistic refinement. Their role was limited to editing the wording of certain sections in order to improve readability and fluency of the manuscript. The intellectual contributions—including the development of ideas, design of experiments, analysis of results, and formulation of conclusions—were carried out entirely by the authors. No part of the research process, data interpretation, or scientific claims relied on the use of LLMs. The authors assume full responsibility for the content presented and ensure its originality and accuracy.

E. Data Ethics Statement

To evaluate the efficacy of StaR-KVQA, we conducted experiments which only use publicly available datasets, namely, OK-VQA [14] and FVQA [19]. We also confirm that no personally identifiable information was utilized, and this research did not involve any human or animal subjects.

F. Prompts

Following the methodology of ROG [13], we process the reasoning paths in two stages. First, we serialize each path by separating its constituent steps with a `<SEP>` token and terminating the sequence with `</PATH>`. Second, these serialized paths are parsed and converted into a structured format where consecutive steps are linked by an arrow (→), e.g., `dog.color` → `dog.size`.

Prompt of Dual-Path Planner

```
Given the image and the question below, generate exactly two relation paths
to help answer the question using a knowledge graph reasoning approach.
Follow the instructions precisely:
1. **vision_path**: Infer a visual reasoning path based on detectable
objects, scenes, or attributes in the image.
2. **text_path**: Derive a semantic reasoning path from the meaning of the
question using background knowledge.

IMPORTANT:
- Output ONLY two lines.
- Use the exact format:
 vision_path: <PATH> relation1 <SEP> relation2 <SEP> ... </PATH>
 text_path: <PATH> relation1 <SEP> relation2 <SEP> ... </PATH>
- Do NOT include any explanations, additional text, or extra lines.
- Replace relations with appropriate knowledge graph predicates (e.g.,
object.type, sports.use).
- Use <SEP> to separate each step in the path.
- End each path with </PATH>.

Example:
Question: What sport can you use this for?
text_path: <PATH> sports.equipment <SEP> sports.name </PATH>
image_path: <PATH> vehicle.type <SEP> vehicle.brand <SEP> sports.use </PATH>

Now answer the following:
{{Image}} {{Question}}
```

Figure 1. **Prompt of Dual-Path Planner.**

Prompt of Reasoning Composer

```
Based on the vision reasoning path {{vision_path}} and the text reasoning path
{{text_path}} , analyze the image to answer the question: {{ Question}} .

Use both paths as your primary guide for reasoning. Apply the visual path to
identify relevant objects or features in the image, and interpret the text
path to understand the semantic relationship needed. Combine both to reach a
clear conclusion.

Do not say the instruction is unclear or incomplete. Assume the paths are
valid and sufficient. Avoid emojis, disclaimers, or speculative language. Be
factual, concise, and directly derive the answer from the two reasoning paths.
```

Figure 2. **Prompt of Reasoning Composer.**

Prompt of Best-Triplet Selector

```
You are a judging module that outputs exactly one lowercase letter from the
provided choices.
Think silently; do not write your reasoning.

Primary criteria:
1) Explanation→Answer entailment.
2) Path→Explanation alignment (C explicitly references vision_path / text_path).
3) Visual plausibility from Image.
4) Coherence and concision.
5) a_pred vs ground_truth correctness (normalized; allow clear synonyms).

Output constraints:
- Output exactly one character from {{choices_letters}} .
- No spaces/newlines/punctuation.

Image: {{ Image}} Question : {{ Question}}

{{ #each candidates as |cand idx|}}
{{ choices_letters[idx]}}:
text_path : {{ cand.text_path}}
vision_path : {{ cand.vision_path }}
C: {{ cand.C}}
a_pred: {{ cand.ans}}
{{ /each}}

ground_truth: {{ ground_truth}}
```

Figure 3. **Prompt of Best-Triplet Selector.**

References

[1] Inclusion AI, Fudong Wang, Jiajia Liu, Jingdong Chen, Jun Zhou, Kaixiang Ji, Lixiang Ru, Qingpei Guo, Ruobing Zheng, Tianqi Li, et al. M2-reasoning: Empowering mllms with unified general and spatial reasoning. *arXiv preprint arXiv:2507.08306*, 2025. 1

[2] 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. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025. 1

[3] Hedi Ben-younes, Rémi Cadène, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2631–2639, 2017. 1

[4] Gheorghe Comanici, Eric Bieber, and Mike Schaekermann et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *ArXiv*, abs/2507.06261, 2025. 1

[5] Junnan Dong, Qinggang Zhang, Huachi Zhou, Daochen Zha, Pai Zheng, and Xiao Huang. Modality-aware integration with large language models for knowledge-based visual question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2417–2429, Bangkok, Thailand, 2024. Association for Computational Linguistics. 1

[6] Abhimanyu Dubey, Abhinav Jauhri, and Abhinav Pandey et al. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. 1

[7] François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. Conceptbert: Concept-aware representation for visual question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 489–498, 2020. 1

[8] Liangke Gui, Borui Wang, Qiuyuan Huang, Alexander G. Hauptmann, Yonatan Bisk, and Jianfeng Gao. Kat: A knowledge augmented transformer for vision-and-language. *ArXiv*, abs/2112.08614, 2021. 1

[9] OpenAI Aaron Hurst, Adam Lerer, and Adam P. Goucher et al. Gpt-4o system card. *ArXiv*, abs/2410.21276, 2024. 1

[10] Gemma Team Aishwarya Kamath, Johan Ferret, and Shreya Pathak et al. Gemma 3 technical report. *ArXiv*, abs/2503.19786, 2025. 1

[11] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Neural Information Processing Systems*, 2018. 1

[12] Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. Revive: Regional visual representation matters in knowledge-based visual question answering. *ArXiv*, abs/2206.01201, 2022. 1

[13] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. *ArXiv*, abs/2310.01061, 2023. 3

[14] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3190–3199, 2019. 1, 3

[15] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14111–14121, 2021. 1

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*, 2013. 1

[17] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1

[18] Sahithya Ravi, Aditya Chinchure, Leonid Sigal, Renjie Liao, and Vered Shwartz. Vlc-bert: Visual question answering with contextualized commonsense knowledge. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1155–1165, 2022. 1

[19] Peng Wang, Qi Wu, Chunhua Shen, Anthony R. Dick, and Anton van den Hengel. Fvqa: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2413–2427, 2016. 3

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 1

[21] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-modal answer validation for knowledge-based vqa. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2712–2721, 2022. 1

[22] Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024. 1

[23] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. *ArXiv*, abs/2109.05014, 2021. 1

[24] Zhaorui Yang, Qian Liu, Tianyu Pang, Han Wang, H. Feng, Minfeng Zhu, and Wei Chen. Self-distillation bridges distribution gap in language model fine-tuning. In *Annual Meeting of the Association for Computational Linguistics*, 2024. 1

[25] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6274–6283, 2019. 1

[26] Yifeng Zhang, Shi Chen, and Qi Zhao. Toward multi-granularity decision-making: Explicit visual reasoning with hierarchical knowledge. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2573–2583, 2023. 1

[27] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Yue Cao, Yangzhou Liu, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Han Lv, Dengnian Chen, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Cong He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Ying Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Lijun Wu, Kai Zhang, Hui Deng, Jiaye Ge, Kaiming Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *ArXiv*, abs/2504.10479, 2025. 1