GenRecal: Generation after Recalibration from Large to Small Vision-Language Models

Byung-Kwan Lee[†]
NVIDIA
byungkwanl@nvidia.com

Ryo Hachiuma
NVIDIA
rhachiuma@nvidia.com

Yong Man Ro
KAIST
ymro@kaist.ac.kr

Yu-Chiang Frank Wang
NVIDIA
frankwang@nvidia.com

Yueh-Hua Wu
NVIDIA
krisw@nvidia.com

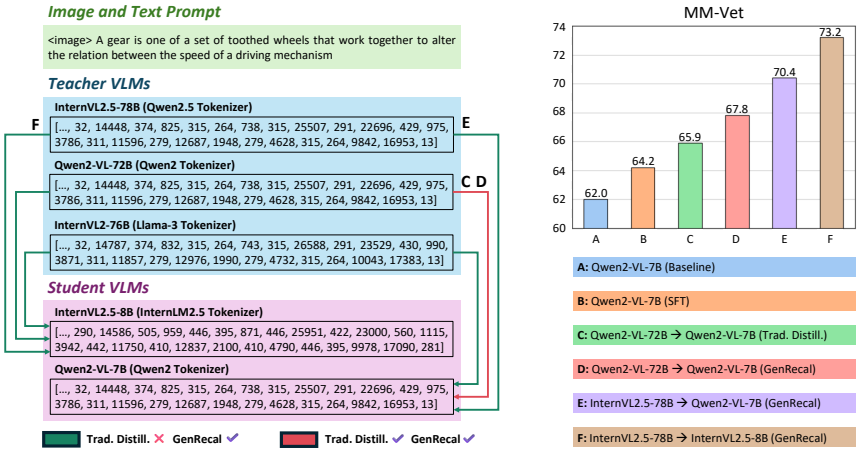


Fig. 1: (Left) Visualizing the token indices of a given image and text prompt and representing the possibility of distillation among various VLM pair combinations, comparing traditional distillation with our proposed distillation framework, GenRecal. Note that the parentheses mean each VLM's LLM tokenizer, '...' indicates the placement of image features, and the number of these features varies depending on the image embedding strategy. (Right) Comparing the performance of a challenging evaluation benchmark, MM-Vet [63], with [A] baseline, [B] SFT on the baseline, [C] traditional distillation and [D] GenRecal from same token types of large VLMs, and GenRecal with more powerful [E] large and [F] small VLMs.

Abstract. Recent advancements in vision-language models (VLMs) have leveraged large language models (LLMs) to achieve performance on par with closed-source systems like GPT-4V. However, deploying these models in real-world scenarios, particularly on resource-constrained devices, remains challenging due to their substantial computational demands. This has spurred interest in distilling knowledge from large VLMs into smaller, more efficient counterparts. A key challenge arises here from the diversity of VLM architectures, which are built on different LLMs and employ varying token types—differing in vocabulary size, token splits,

[†] Project Lead

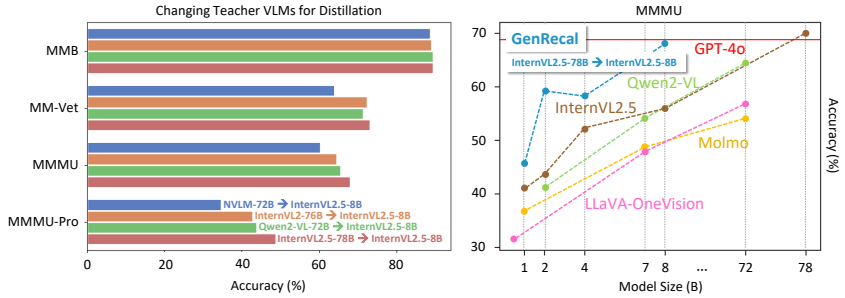


Fig. 2: (Left) Comparison of the challenging benchmark performances, MMB [36], MM-Vet [63], MMMU [65], and MMMU-Pro [66] by changing large VLMs. The more powerful large VLMs we select, the greater the performance improvement we can achieve. (Right) Comparing the performance of the challenging benchmark: MMMU [65], with GenRecal and various VLMs across model sizes. Note that all the experiments in Fig. 1 and Fig. 2 are conducted on the equal training dataset.

and token index ordering. To address this challenge of limitation to a specific VLM type, we present **Gen**eration after **Recal**ibration (GenRecal), a general-purpose distillation framework for VLMs. GenRecal incorporates a Recalibrator that aligns and adapts feature representations between heterogeneous VLMs, enabling effective knowledge transfer across different types of VLMs. Through extensive experiments on multiple challenging benchmarks, we demonstrate that GenRecal significantly improves baseline performances, eventually outperforming large-scale open- and closed-source VLMs.

Keywords: Cross-Tokenizer · Model Distillation · Efficient AI

1 Introduction

Vision-language models (VLMs) have emerged as powerful tools for understanding and processing multimodal information, enabling tasks such as image captioning and visual question answering [32,55]. Recent advancements in VLMs [2, 31,35] have leveraged large-scale language models (LLMs) to enhance reasoning and generation capabilities by incorporating extensive textual knowledge [44,51]. In pursuit of superior performance, state-of-the-art VLMs [12,17,18,29,58] now integrate scaled-up LLMs with up to 72B parameters, achieving results comparable to proprietary models such as GPT-4V [43] and Claude-3.5 Sonnet [3].

However, the increasing scale of recent VLMs introduces substantial computational overhead, limiting their practicality in real-world scenarios—particularly for on-device deployment. To mitigate this challenge, recent work has explored distillation techniques [7, 20, 49] that transfer knowledge from large (teacher) VLMs to smaller (student) ones. Yet, existing distillation methods suffer from a fundamental limitation: they typically assume that teacher and student produce sequences of equal token length, enabling token-level distance metrics such as KL divergence. This assumption breaks down when the two models use different tokenizers or adopt different image-splitting mechanisms (see Appendix A), mak-

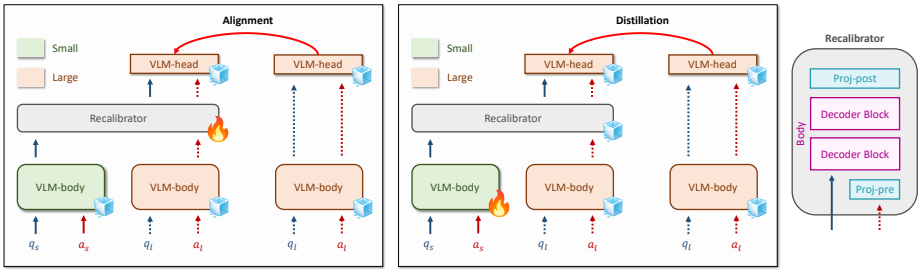


Fig. 3: Overview of the GenRecal architecture and its training stages. We denote the question and answer tokens from the small VLM as q_s and a_s, and those from the large VLM as q_l and a_l. For simplicity, the vision encoder and image-token projector are omitted from the illustration. Note that Recalibrator is used only during training as a bridge between heterogeneous VLMs. During inference, Recalibrator is removed, leaving the small VLM's architecture unchanged and no extra computational cost.

ing the distillation process infeasible. Consequently, the range of viable teacher-student VLM pairs becomes severely restricted (see Appendix B). In other words, current distillation approaches only operate when the teacher and student share identical vocabulary sizes, token splits, and token-index-ordering schemes (collectively referred to as token types), as illustrated on the left side of Fig. 1. These differences naturally lead to mismatched input or output token lengths across VLMs, preventing effective distillation. Note that even VLMs within the same family may employ different token types, further hindering compatibility.

To overcome this limitation, we propose **Gen**eration after **Recal**ibration (GenRecal), a general-purpose distillation framework for token types-agnostic VLM distillation. At its core, GenRecal introduces a *Recalibrator* that aligns and adapts the feature representations of small VLMs with those of large VLMs. This design is inspired by prior studies demonstrating that word embeddings from different models can be linearly mapped into a shared embedding space [15, 42, 50]. Unlike approaches that align only word embeddings, Recalibrator performs joint visual–linguistic alignment on the hidden representations before the language head. By projecting small VLM features into large VLMs' feature space as shared representation, Recalibrator effectively bridges the gap between the large and small VLMs, enabling general-purpose knowledge transfer.

To demonstrate the necessity of GenRecal, we first compare its performance with that of traditional distillation, as shown in experiment types C and D of Fig. 1. Although both experiments employ the same token types of VLMs—Qwen2-VL-72B as the teacher and Qwen2-VL-7B as the student—GenRecal consistently outperforms the traditional distillation method implemented by LLaVA-KD [7]. This result highlights that aligning feature representations between large and small VLMs is crucial for minimizing information loss during distillation, even when the models are homogeneous. Moreover, when we replace the large VLM from Qwen2-VL-72B with the more powerful InternVL2.5-78B, GenRecal achieves enhanced performance. Subsequently, replacing the small VLM with InternVL2.5-8B leads to an even stronger model, as summarized on the right

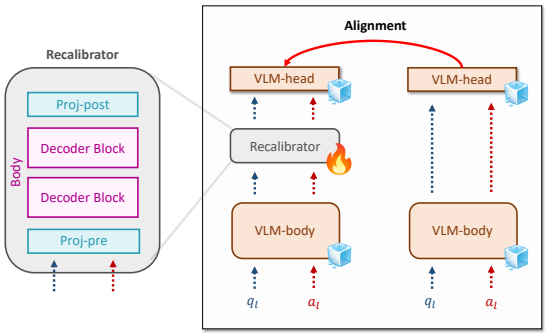


Fig. 4: Overview of regularization simultaneously done with first training stage-alignment. Note that its propagation rule is different with Fig. 3(c) where Recalibrator takes small VLM's tokens for the question and large VLM's tokens for the answer. In contrast, in this figure, only large VLM's tokens (question and answer) are fed into Recalibrator. Basically, *Rec-body* has the hidden dimension of the small VLM, so we should change this hidden dimension of large VLM's tokens to that of small VLM's ones via *Proj-pre* when the large VLM's tokens are fed into Recalibrator.

side of Fig. 1. These experiments highlight the significance of GenRecal because traditional methods cannot handle VLM pairs with different token types.

Our contributions can be summarized as follows:

- **Token Types-agnostic Recalibration:** GenRecal employs the Recalibrator to align and adapt the feature representations of large and small VLMs, enabling general-purpose distillation across token types that differ in vocabulary size, token splits, and token-index ordering.
- **Broad Applicability:** GenRecal is compatible with a wide range of VLM architectures across different model sizes, overcoming the challenge of selecting large VLMs that have different token types and demonstrating its practicality for real-world deployment in resource-constrained settings.

2 Related Work

Large-scale VLMs such as NVLM-72B [17], Qwen2-VL-72B [58], and InternVL2.5-78B [12] have recently approached the performance of GPT-4V [43] and Claude-3.5 Sonnet [3] (see Appendix C for the evolution of VLMs). However, they impose a significant computational burden in real-world applications, such as on-device processing. Hence, it is necessary to develop small VLMs for deployment on lightweight devices, and this demand has led to two types of distillation approaches. The first constructs visual instruction tuning datasets [11, 22, 25, 29, 54, 59, 67, 68] from large-scale VLMs and trains small VLMs on these curated datasets. The second performs feature distillation: LLaVA-MoD [49] conducts logit-based distillation on a mixture-of-experts (MoE) [46,47] architecture. LLaVA-KD [7] also uses logit-based distillation but adopts a three-stage training process in which the trainable parameters differ at each stage to effectively warm up and distill the knowledge. Align-KD [20] distills both vision-encoder

Algorithm 1 First stage Loss Functions (Alignment)

Input: $(q_l,\ a_l)$, $(q_s,\ a_s)$, gt_l (label index)
$[z_{q_l}, z_{a_l}] \leftarrow \textit{VLM-body}_l([q_l, a_l])$
$[z_{q_s}, z_{a_s}] \leftarrow \textit{VLM-body}_s([q_s, a_s])$
$[r_{q_s}, r_{a_l}] \leftarrow \textit{Recalibrator}([z_{q_s}, z_{a_l}])$
$\mathcal{L}_{ar} \leftarrow \mathrm{CE}(\textit{VLM-head}_l(r_{a_l}), gt_l)$
$\mathcal{L}_{kl} \leftarrow \mathcal{D}_{\mathrm{KL}}(\textit{VLM-head}_l(z_{a_l}) \mid \textit{VLM-head}_l(r_{a_l}))$
Return: $\mathcal{L}_{ar} + \mathcal{L}_{kl}$

features and decoder logits, while encouraging the student VLM's first decoder layer's attention map to resemble that of the large VLM. MoVE-KD [9] employs multiple vision encoders for the large VLM and applies a mixture of LoRA [26] to the small VLM, distilling visual information through visual attention maps. Besides, there are earlier distillation studies (see Appendix C), but they consider neither token-derived nor different-token-type distillation.

However, existing works are largely constrained to VLMs sharing identical token types, including vocabulary size, token segmentation, and token index ordering. When the large and small VLMs employ different tokenizers or image embedding strategies, distillation becomes problematic—KL divergence cannot be computed due to mismatched output token counts, and token indices may no longer correspond to semantically equivalent meaning. Overcoming this limitation would represent a significant advancement, broadening the applicability and robustness of distillation techniques for diverse VLM architectures.

3 GenRecal: Generation after Recalibration

3.1 Model Architecture and Components

We briefly illustrate the overall model architecture and training procedure in Fig. 3. GenRecal consists of three main components: a large (teacher) VLM, a small (student) VLM, and a Recalibrator module. For the large VLMs, we select models with over 72B parameters to ensure competitive performance: NVLM-72B (Qwen2-72B) [17], Qwen2-VL-72B (Qwen2-72B) [58], InternVL2-76B (Llama3-70B) [13], and InternVL2.5-78B (Qwen2.5-72B) [12]. For the small VLMs, we adopt various model sizes, including Qwen2-VL-2B/7B (Qwen2-2B/7B) and InternVL2.5-1B/2B/4B/8B (InternLM2.5-1.8B/7B and Qwen2.5-0.5B/3B/72B). Note that the parentheses denote the LLMs used within each VLM. Throughout this section, we divide a VLM architecture into four key modules: the vision encoder, vision projector, *VLM-body*, and *VLM-head* (*i.e.* language head). Recalibrator is designed to acquire shared feature representations within a common latent space. It consists of two decoder blocks and two projection layers, as depicted in Fig. 3(c). We denote the decoder blocks as *Rec-body*, and the projection layers as *Proj-pre* and *Proj-post*. The decoder blocks, *Rec-body*, adopt the same structural design as the student VLM's decoder, while both *Proj-pre* and *Proj-post* are implemented as single linear layers.

Algorithm 2 First stage Regularization (Alignment)

Input: $(q_l,\ a_l)$, gt_l (label index)
$[z_{q_l}, z_{a_l}] \leftarrow \textit{VLM-body}_l([q_l, a_l])$
$[r_{q_l}, r_{a_l}] \leftarrow \textit{Recalibrator}([z_{q_l}, z_{a_l}])$
$\mathcal{L}_{ar} \leftarrow \mathrm{CE}(\textit{VLM-head}_l(r_{a_l}), gt_l)$
$\mathcal{L}_{kl} \leftarrow \mathcal{D}_{\mathrm{KL}}(\textit{VLM-head}_l(z_{a_l}) \mid \textit{VLM-head}_l(r_{a_l}))$
Return: $\mathcal{L}_{ar} + \mathcal{L}_{kl}$

Algorithm 3 Second stage Loss Functions (Distillation)

Input: $(q_l,\ a_l)$, $(q_s,\ a_s)$, $(gt_l,\ gt_s)$ (label index)
$[z_{q_l}, z_{a_l}] \leftarrow \textit{VLM-body}_l([q_l, a_l])$
$[z_{q_s}, z_{a_s}] \leftarrow \textit{VLM-body}_s([q_s, a_s])$
$[r_{q_s}, r_{a_l}] \leftarrow \textit{Recalibrator}([z_{q_s}, z_{a_l}])$
$\mathcal{L}_{ar} \leftarrow \mathrm{CE}(\textit{VLM-head}_l(r_{a_l}), gt_l)$
$\mathcal{L}_{ar} \leftarrow \mathcal{L}_{ar} + \mathrm{CE}(\textit{VLM-head}_s(z_{a_s}), gt_s)$
$\mathcal{L}_{kl} \leftarrow \mathcal{D}_{\mathrm{KL}}(\textit{VLM-head}_l(z_{a_l}) \mid \textit{VLM-head}_l(r_{a_l}))$
Return: $\mathcal{L}_{ar} + \mathcal{L}_{kl}$

3.2 Design and Propagation of Recalibrator

To construct a shared representation space, we first extract and align hidden features from both the large and small VLMs. Since the hidden space of a large VLM typically encompasses a richer vocabulary and more expressive representations, we regard it as the shared latent space which the small VLM is projected into. However, direct alignment of the two VLMs' tokens within this space is infeasible due to discrepancies in vocabulary size, token split, and token index ordering. To overcome this challenge, we draw inspiration from the principle of *autoregressive modeling*, wherein a model predicts answer tokens for the given question tokens.

Specifically, we feed the same question-answer pair into both the large and small VLMs, obtaining their *VLM-body* outputs as:

$$\begin{aligned} [z_{q_l}, z_{a_l}] &= \textit{VLM-body}_l([q_l, a_l]), \qquad (l : \text{large}) \\ [z_{q_s}, z_{a_s}] &= \textit{VLM-body}_s([q_s, a_s]), \qquad (l : \text{small}) \end{aligned} \tag{1}$$

where q and a denote the question and answer tokens produced by each model's vision encoder, vision projector, and word embedding [41]. We then extract the question features z_{q_s} from the small VLM and the answer features z_{a_l} from the large VLM, and concatenate them to form a joint sequence: $[z_{q_s}, z_{a_l}]$, which is then passed through Recalibrator. Because directly embedding both features into a shared space is infeasible due to different token types, we instead design an autoregressive loss that predicts the answer token index of the large VLM given the question token index of the small VLM. This loss encourages Recalibrator to effectively project the small VLM's features into the large VLM's latent space.

However, the hidden dimensions of large and small VLMs often differ. To address this, in Recalibrator, we first apply *Proj-pre* to z_{a_l}, aligning its dimen-

Table 1: Evaluation of standard model size open-source VLMs and GenRecal on several challenging vision-language benchmarks: AI2D [27], ChartQA [40], MathVista [38], MMB [36], MMB^{CN} [36], MM-Vet [63], MMMU [65], MMMU-Pro [66], BLINK [21], SEED-2-Plus [30], and RealWorldQA (RWQA). We use the notations of 'XX-GenRecal (YY)' where 'XX' and 'YY' denote the student (small) and teacher (large) VLM.

VLMs	AI2D	ChartQA	MathVista	MMB	MMB^{CN}	MM-Vet	MMMU	MMMU-Pro	BLINK	SEED-2-Plus	RWQA
Cambrian-1-8B [57]	73.0	73.3	49.0	75.9	-	48.0	42.7	-	44.9	59.7	60.0
Cambrian-1-13B [57]	73.6	73.8	48.0	75.7	-	48.9	40.0	-	43.1	60.0	58.6
Eagle-8B [48]	76.1	80.1	52.7	75.9	-	33.3	43.8	-	22.4	57.5	63.8
Eagle-13B [48]	74.0	77.6	54.4	75.7	-	42.6	41.6	-	21.8	60.2	62.9
VILA1.5-8B [34]	58.8	-	37.3	75.3	69.9	43.2	38.6	-	39.5	45.2	43.4
VILA1.5-13B [34]	69.9	-	42.5	74.9	66.3	44.3	37.9	-	48.1	50.2	53.3
CogVLM2-19B [24]	73.4	81.0	38.6	80.5	-	60.4	44.3	-	-	66.0	62.9
LLaVA-OneVision-7B [29]	81.4	80.0	63.2	80.8	-	57.5	48.8	24.1	53.0	65.4	69.9
InternVL2-8B [14]	83.8	83.3	58.3	81.7	81.2	54.2	49.3	29.0	50.9	67.3	64.2
MiniCPM-V2.5-8B [62]	78.4	-	54.3	77.2	74.2	52.8	45.8	-	-	61.4	63.5
MiniCPM-V2.6-8B [62]	82.1	-	60.6	-	-	60.0	49.8	27.2	55.2	65.7	65.0
Qwen2-VL-7B [58]	77.5	83.0	58.2	83.0	80.5	62.0	54.1	30.5	53.8	68.6	68.5
InternVL2.5-8B [12]	84.8	84.8	64.4	84.6	82.6	62.8	56.0	34.3	54.8	69.7	70.1
Qwen2-VL-7B-GenRecal (InternVL2.5-78B)	**93.9**	**95.3**	68.8	88.4	87.4	70.4	65.6	**49.6**	64.3	70.7	79.2
InternVL2.5-8B-GenRecal (InternVL2.5-78B)	93.0	93.6	**74.9**	**89.5**	**88.2**	**73.2**	**68.1**	48.8	**65.3**	**72.3**	**81.4**
VILA1.5-3B [34]	57.9	-	31.6	-	-	38.8	34.2	-	39.7	41.4	53.2
Phi-3.5-Vision-4B [1]	77.8	81.8	43.9	76.0	66.1	43.2	43.0	19.7	58.3	62.2	53.6
InternVL2-4B [14]	78.9	81.5	58.6	78.6	73.9	51.0	34.3	32.7	46.1	63.9	60.7
InternVL2.5-4B [14]	81.4	84.0	60.5	81.1	79.3	60.6	52.3	32.7	50.8	66.9	64.3
InternVL2.5-4B-GenRecal (InternVL2.5-78B)	**90.0**	**91.1**	**70.4**	**88.4**	**86.2**	**66.1**	**58.3**	**45.9**	**63.5**	**69.3**	**75.2**
InternVL2-2B [14]	74.1	76.2	46.3	73.2	70.9	39.5	34.3	18.2	43.8	60.0	57.3
Qwen2-VL-2B [58]	60.2	73.5	43.0	74.9	73.5	49.5	41.1	21.2	45.2	61.2	62.6
Aquila-VL-2B [12]	75.0	76.5	59.0	-	-	43.8	47.4	-	34.1	63.0	65.0
InternVL2.5-2B [12]	74.9	79.2	51.3	74.7	71.9	60.8	43.6	23.7	44.0	60.0	60.1
Qwen2-VL-2B-GenRecal (InternVL2.5-78B)	89.1	**90.9**	60.5	82.9	81.0	57.3	52.9	**37.5**	53.4	65.5	72.8
InternVL2.5-2B-GenRecal (InternVL2.5-78B)	**91.8**	90.7	**62.5**	**84.1**	**80.4**	**61.2**	**59.2**	36.6	**55.5**	**67.3**	**73.0**
LLaVA-OneVision-0.5B [29]	57.1	61.4	34.8	61.6	55.5	32.2	31.4	-	52.1	45.7	55.6
InternVL2-1B [14]	64.1	72.9	37.7	65.4	60.7	32.7	36.7	14.8	38.6	54.3	50.3
InternVL2.5-1B [12]	69.3	75.9	43.2	70.7	66.3	48.8	40.9	19.4	42.0	59.0	57.5
InternVL2.5-1B-GenRecal (InternVL2.5-78B)	**82.9**	**89.4**	**56.5**	**80.1**	**74.4**	**54.4**	**45.6**	**28.7**	**48.5**	**65.1**	**70.8**

sionality to that of z_{q_s}. The concatenated sequence $[z_{q_s}, \textit{Proj-pre}(z_{a_l})]$ is then propagated through *Rec-body* decoder blocks. The output of *Rec-body* is subsequently passed through *Proj-post* to restore the hidden dimensionality of the large VLM. This forward rule of Recalibrator is illustrated in Fig. 3(c).

Finally, the answer part of the resulting features $[r_{q_s}, r_{a_l}] = \textit{Recalibrator}([z_{q_s}, z_{a_l}])$ is fed into the *VLM-head* of the large VLM to compute the autoregressive loss:

$$\mathcal{L}_{ar} = \text{CE}(\textit{VLM-head}_l(r_{a_l}), gt_l), \tag{2}$$

where gt_l denotes the answer token index from the large VLM. Additionally, we employ a KL divergence loss to further enhance the Recalibrator's projection capability in the first training stage and to distill knowledge from the teacher in the second training stage. This process enables the large VLM to interpret the hidden features of the small VLM, thus establishing a bridge for general-purpose distillation across heterogeneous token types.

3.3 Training Process

To achieve general-purpose distillation, we adopt a three-stage training process. In the first stage, only Recalibrator is trained while keeping all parameters of both the large and small VLMs frozen, as illustrated in Fig. 3(a). During this stage, we compute both the autoregressive loss and the KL divergence, described in Algorithm 1 where the purple components indicate the trainable parameters. This step primarily serves to align the feature representations between the large and small VLMs. For the first stage, we introduce regularization loss, as illustrated in Fig. 4, which is designed to prevent the feature representations of

Table 2: (b) Comparison of GenRecal with large-scale open-source and closed-source VLMs on challenging benchmarks: MMB [36], MM-Vet [63], MM-Vet-v2 [64], MMMU [65], MMMU-Pro [66], MMStar [10], AI2D [27], ChartQA [40], SEED-2-Plus [30], MathVista [38], BLINK [21], RealWorldQA (RWQA).

VLMs	MMB	MM-Vet	MM-Vet-v2	MMMU	MMMU-Pro	MMStar	AI2D	ChartQA	SEED-2-Plus	MathVista	BLINK	RWQA
NVLM-72B [17]	-	58.9	-	59.7	-	63.7	85.2	86.0	68.4	66.6	48.0	69.9
LLaVA-OneVision-72B [29]	85.8	60.6	-	56.8	31.0	65.8	85.6	83.7	-	67.5	55.4	71.9
Molmo-72B [18]	-	61.1	-	54.1	-	63.3	83.4	87.3	-	58.6	-	73.7
Qwen2-VL-72B [58]	86.5	74.0	68.7	64.5	46.2	68.3	88.1	88.3	**72.3**	70.5	60.5	77.8
InternVL2-76B [14]	86.5	65.7	68.4	62.7	40.0	67.4	87.6	88.4	69.7	65.5	56.8	72.2
InternVL2.5-78B [29]	88.3	72.3	65.5	**70.1**	48.6	**69.5**	89.1	88.3	71.3	72.3	63.8	78.7
Claude-3.5-Sonnet [3]	82.6	70.1	**71.8**	68.3	51.5	65.1	81.2	90.8	71.7	67.7	60.1	60.1
Gemini-1.5-Pro [56]	73.9	64.0	66.9	62.2	46.9	59.1	79.1	87.2	70.8	63.9	59.1	67.5
GPT-4o (0513) [43]	83.4	69.1	71.0	69.1	**51.9**	64.7	84.6	85.7	72.0	63.8	**68.0**	75.4
InternVL2.5-8B-GenRecal (NVLM-72B)	88.8	63.9	59.3	60.3	34.7	65.1	88.5	84.3	69.3	67.5	55.6	72.5
InternVL2.5-8B-GenRecal (InternVL2-76B)	89.0	72.4	65.0	64.6	42.8	65.8	92.1	90.4	70.9	71.9	57.7	75.5
InternVL2.5-8B-GenRecal (Qwen2-VL-72B)	**89.5**	71.4	62.6	65.6	43.8	66.1	92.2	87.5	71.3	71.4	61.4	78.8
InternVL2.5-8B-GenRecal (InternVL2.5-78B)	**89.5**	**73.2**	67.2	68.1	48.8	67.8	**93.0**	**93.6**	**72.3**	**74.9**	65.3	**81.4**

Recalibrator from drifting too far from those of the large VLM. We use both Algorithm 1 and Algorithm 2 together in the first stage.

In the second stage, we keep using the same loss functions of the first stage and additionally incorporate the small VLM's own autoregressive loss in Algorithm 3. As depicted in Fig. 3(b), this stage enables knowledge distillation from the large VLM to the small one by training the small VLM's *VLM-body*. In the final stage, we remove both Recalibrator and the large VLM, and fine-tune the student VLM. Specifically, we train all parameters of the student VLM except for the vision encoder through supervised fine-tuning (SFT) to further enhance its instruction-following capability.

4 Experiments

4.1 Implementation Detail

To ensure reproducibility, we outline three key technical aspects of GenRecal: (a) the structure of the Recalibrator, (b) the projection layers, and (c) training and evaluation details.

(a) Structure Detail of Recalibrator. This is composed of two transformer decoder blocks (*Rec-body*) and two projectors (*Proj-pre* and *Proj-post*). The decoder blocks follow all configurations of the small VLM's decoder block, such as the causal mask, hidden dimensions, number of heads, and FFN structure. We use transformer decoder blocks for *Rec-body* because they naturally process the concatenated student and teacher features in Fig. 3 as a sequence. Moreover, these features must preserve token ordering, so we introduce new positional embeddings (NPE). This design enables effective sequential modeling and representation alignment between heterogeneous VLMs. To this end, we employ an additional RoPE [53] to realign their positional embeddings, and their position IDs are reassigned accordingly. Lastly, we apply an additional layer norm [4] to the output features of the Recalibrator for stable adaptation.

(b) Projection Layer Design. Two projection layers (*Proj-pre* and *Proj-post*) are linear layers introduced for hidden dimension matching. Similar to the role of vision projector that matches the hidden dimensions between a CLIP-like vision encoder and an LLM, *Proj-pre* matches the teacher features to the student

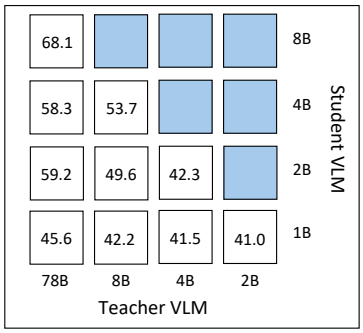


Fig. 5: Distillation performance on MMMU [65] for various pairings os teacher and student VLM. Each cell indicates the resulting score when using the corresponding teacher (rows) and student (columns) model sizes.

hidden dimension, while *Proj-post* restores the dimension back to the teacher hidden dimension so that the large *VLM-head* can be utilized for distillation. This explanation is technically depicted in Fig. 3(c) and Fig. 4.

(c) Details of Training and Evaluation. By using DeepSpeed engine with ZeRO-3 [45], we train and evaluate GenRecal on 128 NVIDIA A100 80GB GPUs. We use AdamW optimizer [37] and apply a linearly decayed learning rate from 1e-4 to 1e-5 at each training stage. The first and second stages use the entire 9M dataset (see Appendix E for dataset composition and analysis), taking approximately 2–4 hours and 8–11 hours, respectively, depending on model size. The last stage takes 4–6 hours on a 6M dataset obtained by removing general visual question answering samples [29]. For stable training, we handle large batch sizes via gradient accumulation with 16 steps. We use 4 (first and third stages) or 2 (second stage) samples per GPU, leading to a total batch size of 8192 ($128\times16\times4$) or 4096 ($128\times16\times2$). For evaluation, we remove the Recalibrator and the large VLM, and let the small VLM generate answers using the default generation hyperparameters.

(d) Dataset Composition. Since the 9M corpus has a decisive impact on the results, we summarize it here. It covers general visual question answering, dense captioning, chart/diagram/document understanding, commonsense knowledge, science & math, and multi-dimensional reasoning, curated from public sources such as LLaVA-OneVision [29], Cambrian [57], Infinity-MM [22], and Dense-Fusion [33], among others. For analysis we group these into three domains—*Knowledge*, *Science & Math*, and *Chart & Document*—which align with benchmarks such as MMMU [65], MathVista [38], and ChartQA [40], respectively. A full source list and per-domain breakdown are provided in Appendix E.

4.2 Validation and Analysis for GenRecal

Higher Size, Higher Performance. Tab. 1 demonstrates that GenRecal outperforms not only standard-size VLMs but also smaller ones, highlighting that GenRecal generalizes well across model sizes. This table also shows that choosing a higher-performing small VLM such as InternVL2.5-8B [12] substantially

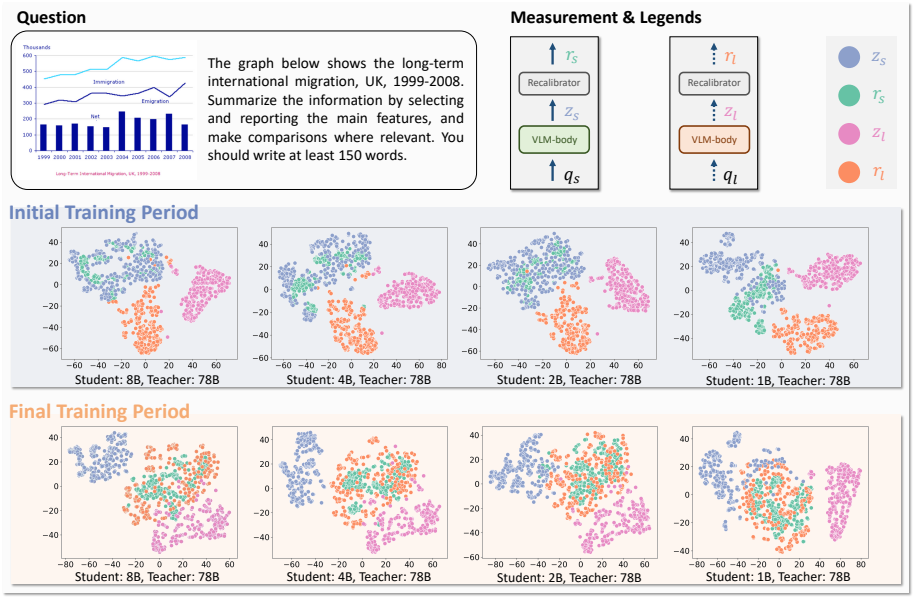


Fig. 6: An overview of our training pipeline, illustrating both the question prompt and the measurement/legend annotations (top), followed by t-SNE visualizations (bottom) of teacher and student VLM pairings at the initial and final training stages. The question prompt (upper-left) shows the format of the question, while the measurement and legend box (upper-right) shows key model components to measure. Each scatter plot in the lower panels corresponds to a different combination of teacher and student VLM sizes, capturing how the learned representations evolve from early to later training.

improves the overall distillation results. Furthermore, for a fixed large VLM, selecting a smaller student VLM leads to lower distillation performance, implying that employing more capable small VLMs is crucial for achieving higher distillation performance.

Next, we examine the effect of changing large VLMs to determine whether the aforementioned property holds consistently across different large VLMs. As shown in Fig. 2 and Tab. 2, selecting more powerful large VLMs leads to greater performance improvements. To investigate this further, we use the InternVL2.5 series [12] and experiment with various combinations of large and small VLM sizes, as illustrated in Fig. 5. The results consistently indicate that choosing larger teacher and student VLMs leads to higher distillation performance.

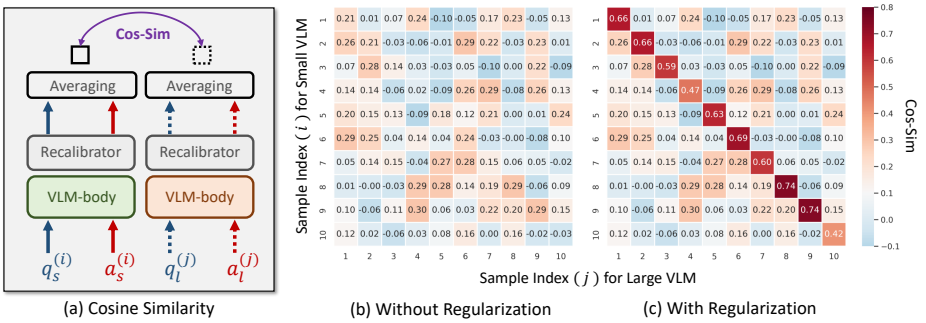
Analysis of Recalibrator. We now focus on the Recalibrator to explore its role in general-purpose distillation. First, we show that the loss curves for training the Recalibrator converge stably across various combinations of large and small VLMs (see Appendix D). Since the Recalibrator losses ('Recalib(q_s, a_l)' and 'Recalib(q_l, a_l)') are minimized well relative to the perplexity of 'SmallVLM(q_s, a_s)' and 'LargeVLM(q_l, a_l)', we infer that the Recalibrator successfully aligns the feature representations of large and small VLMs, and that it generalizes well across teacher–student combinations. Furthermore, Fig. 6 visualizes multiple feature spaces to assess whether the large- and small-VLM features after the Recali-

Table 3: Comparing the final distillation performances from the teacher model: InternVL2.5-78B [12] with and without the regularization term (denoted as "Reg"). Note that, the student models are each InternVL2.5-8B, 4B, 2B, and 1B.

VLMs	Reg	MMB	MathVista	MM-Vet	MMMU
InternVL2.5-8B	-	84.6	64.4	62.8	56.0
w. GenRecal	✗	88.2	69.8	63.5	58.9
w. GenRecal	✓	**89.5**	**74.9**	**73.2**	**68.1**
InternVL2.5-4B	-	81.1	60.5	60.6	52.3
w. GenRecal	✗	82.9	61.7	61.1	53.6
w. GenRecal	✓	**88.4**	**70.4**	**66.1**	**58.3**
InternVL2.5-2B	-	74.7	51.3	60.8	43.6
w. GenRecal	✗	75.8	53.2	60.9	45.4
w. GenRecal	✓	**84.1**	**62.5**	**61.2**	**59.2**
InternVL2.5-1B	-	70.7	43.2	48.8	40.9
w. GenRecal	✗	72.6	45.9	49.8	41.1
w. GenRecal	✓	**80.1**	**56.5**	**54.4**	**45.6**

Table 4: Comparing SFT, traditional distillations, and GenRecal where we use equal training dataset (Section 4.1(b)) and choose the teacher (Qwen2-VL-72B [58]) that has the same token type of student.

VLMs	MMB	MathVista	MM-Vet	MMMU
Qwen2-VL-7B	83.0	58.2	62.0	54.1
w. SFT	84.3	60.5	64.2	56.3
w. MiniLLM	84.4	61.3	65.1	57.4
w. DistiLLM	84.8	61.5	65.3	57.9
w. LLaVA-KD	85.0	61.8	65.9	58.2
w. GenRecal	**87.8**	**69.5**	**67.8**	**64.2**
Qwen2-VL-2B	74.9	43.0	49.5	41.1
w. SFT	76.3	45.8	51.8	44.3
w. MiniLLM	77.5	46.2	52.5	45.3
w. DistiLLM	77.9	46.6	53.0	46.1
w. LLaVA-KD	78.3	46.9	53.7	46.9
w. GenRecal	**82.9**	**58.3**	**57.1**	**51.4**



(a) Cosine Similarity (b) Without Regularization (c) With Regularization

Fig. 7: The process of (a) computing cosine similarity, (b) the result without regularization or (c) with regularization. Because the number of the embedded tokens for small and large VLMs are naturally different due to vocabulary size, token split, and token ordering, therefore we do average the output tokens and compute cosine similarity.

brator are truly matched and can be regarded as a shared representation. The small-VLM features before the Recalibrator are initially close to those after it, but by the end of training they become matched to the large-VLM features after the Recalibrator. This indicates that the Recalibrator is effective at forming a shared feature representation for large and small VLMs.

Importance of Regularization. We conduct an ablation study that investigates how important the regularization term is for the final distillation performances. We first remove the regularization term and analyze the differences before and after its removal. To illustrate this in detail, we simply prepare 10 different question-answer pairs and input them through *VLM-body* of both small and large VLMs. Afterwards, we propagate their output features into Recalibrator, as described in Fig. 7(a). We then compute a 10×10 cosine similarity matrix between Recalibrator's output features from the small and large VLMs. To ensure reliable cosine similarity values, we use all training dataset samples and compute the average of the cosine similarity values for all the samples.

With the regularization term included, as shown in Fig. 7(c), we observe that the diagonal values in the cosine similarity matrix are significantly higher than the off-diagonal values. This implies that the features of small and large VLMs are explicitly shared through Recalibrator, aligned with the results in

Table 5: Importance of the autoregressive loss ($\mathcal{L}_{ar}$) for representation alignment. We compare the baseline VLMs, GenRecal trained without $\mathcal{L}_{ar}$ (✗, using only $\mathcal{L}_{kl}$), and GenRecal trained with $\mathcal{L}_{ar}$ (✓), using InternVL2.5-78B [12] as the teacher VLM.

VLMs	$\mathcal{L}_{ar}$	AI2D	ChartQA	MathVista	MMB	MMB^{CN}	MM-Vet	MMMU	MMMU-Pro	BLINK	SEED-2-Plus	RWQA	Avg
Qwen2-VL-7B [58]	-	77.5	83.0	58.2	83.0	80.5	62.0	54.1	30.5	53.8	68.6	68.5	65.4
InternVL2.5-8B [12]	-	84.8	84.8	64.4	84.6	82.6	62.8	56.0	34.3	54.8	69.7	70.1	68.1
Qwen2-VL-7B-GenRecal (InternVL2.5-78B)	✗	66.2	67.5	47.8	63.4	55.9	49.6	45.8	30.5	52.1	65.3	60.9	55.0
InternVL2.5-8B-GenRecal (InternVL2.5-78B)	✗	68.4	70.1	55.2	68.9	62.5	55.4	50.6	35.1	58.7	68.2	64.4	59.8
Qwen2-VL-7B-GenRecal (InternVL2.5-78B)	✓	93.9	95.3	68.8	88.4	87.4	70.4	65.6	49.6	64.3	70.7	79.2	75.8
InternVL2.5-8B-GenRecal (InternVL2.5-78B)	✓	93.0	93.6	74.9	89.5	88.2	73.2	68.1	48.8	65.3	72.3	81.4	77.1

Fig. 6 as well. In contrast, when the regularization term is removed, as depicted in Fig. 7(b), the diagonal values are comparable with the off-diagonal values. This indicates that Recalibrator fails to explicitly align large and small VLMs' features, preventing feature matching and sharing. As a result, this leads to a significant drop in final distillation performances, as reported in Tab. 3. Insufficient alignment of large and student VLMs reduces efficient knowledge transfer during distillation. These findings highlight the crucial role of the regularization term in feature alignment and knowledge transfer for general-purpose distillation.

Autoregressive Loss for Representation Alignment. To align the student representations with the teacher's latent space, GenRecal uses two objectives: KL divergence loss ($\mathcal{L}_{kl}$) and autoregressive loss ($\mathcal{L}_{ar}$). The KL divergence loss performs standard distillation by matching the logit-level distribution of teacher and student. However, matching distributions alone does not guarantee precise alignment at the token level. The autoregressive loss provides explicit hard-target supervision, allowing it to predict the exact ground-truth token sequence. In short, $\mathcal{L}_{kl}$ transfers knowledge at the distribution level, whereas $\mathcal{L}_{ar}$ ensures exact alignment at the token level. As shown in Tab. 5, removing $\mathcal{L}_{ar}$ consistently degrades performance across benchmarks, confirming that autoregressive supervision is essential for effective representation alignment.

4.3 Ablation Studies: Training Computation and Configurations

In Tab. 6(a), we compare the model sizes and training FLOPs of the small VLM, large VLM, and Recalibrator to assess whether Recalibrator introduces additional computational overhead during training. The results show that Recalibrator's FLOPs are substantially lower than those of both the large and small VLMs. Note that Recalibrator is removed at inference time, thus incurring no additional computational cost during inference. We further conduct a series of ablation studies to examine various configurations affecting the distillation performance of GenRecal. As shown in Tab. 6(b), we vary the decoder depth of Recalibrator from 1 to 20 and select two depths for layer number of *Rec-body* based on a trade-off between computational efficiency and performance. Additionally, Tab. 6(c) demonstrates the importance of employing a new positional embedding (NPE) to handle newly introduced position IDs and to integrate RoPE [53] from both large and small VLMs, which have different token types.

To analyze the effect of dataset scale, we control the dataset size from 1M to 9M samples, as shown in Tab. 6(d). The performance improvements between 5M and 9M samples are marginal, suggesting that a 5M dataset is a practical

Table 6: FLOPs comparison and several ablation studies on various configurations influencing the performance of GenRecal. Note that the teacher VLM in (b), (c), (d), (e), and (f) is InternVL2.5-78B [12], and the student VLM in (b) is InternVL2.5-8B [12].

(a) Model Size and FLOPs

Small VLM		Large VLM		Recalibrator Size
Qwen2-VL-7B	(25.7×10^{12})	Qwen2-VL-72B	(260×10^{12})	524.8M (2.27×10^{12})
InternVL2.5-8B	(27.1×10^{12})	InternVL2.5-78B	(269×10^{12})	503.3M (2.2×10^{12})
InternVL2.5-8B	(27.1×10^{12})	Qwen2-VL-72B	(260×10^{12})	503.3M (2.2×10^{12})
Qwen2-VL-7B	(25.7×10^{12})	InternVL2.5-78B	(269×10^{12})	524.8M (2.27×10^{12})

(b) Recalibrator Depth

Depths	MMB	MathVista	MM-Vet	MMMU	Avg
20	89.6	74.5	73.6	68.3	76.5
15	89.8	75.2	73.5	69.4	76.7
10	89.6	75.0	73.4	68.3	76.6
5	89.4	74.8	73.3	68.2	76.4
3	89.5	74.9	73.2	68.1	76.4
1	89.0	73.0	72.0	67.5	74.9

(c) Effect of NPE

VLM	NPE	MMB	MathVista	MM-Vet	MMMU	Avg
Qwen2-VL-7B-GenRecal	✗	79.2	60.0	64.0	63.9	66.8
Qwen2-VL-7B-GenRecal	✓	**88.4**	**68.8**	**70.4**	**65.6**	**73.3**
InternVL2.5-8B-GenRecal	✗	85.2	65.1	67.3	64.4	70.5
InternVL2.5-8B-GenRecal	✓	**89.5**	**74.9**	**73.2**	**68.1**	**76.4**

(d) Training Dataset Size

Dataset Size	MMB	MathVista	MM-Vet	MMMU	Avg
9M	**89.5**	**74.9**	**73.2**	**68.1**	**76.4**
7M	89.4	74.7	73.0	67.8	76.2
5M	89.3	74.5	72.8	67.6	76.0
3M	88.9	73.6	71.9	66.6	75.3
1M	86.6	68.6	67.0	60.9	70.8
0M	84.6	64.4	62.8	56.0	67.0

(e) Fine-Tuning

VLM	Finetuned	MMB	MathVista	MM-Vet	MMMU	Avg
InternVL2.5-78B [12]	✗	88.3	72.3	72.3	70.1	75.8
InternVL2.5-78B-SFT	✓	**89.4**	**75.2**	**74.2**	**71.7**	**77.6**
InternVL2.5-8B [12]	✗	84.6	64.4	62.8	56.0	67.0
InternVL2.5-8B-SFT	✓	86.5	65.2	65.0	59.9	69.2
InternVL2.5-8B-GenRecal	✓	**89.5**	**74.9**	**73.2**	**68.1**	**76.4**

(f) Cross-Tokenizer Methods

VLM	MathVista	MMB	MM-Vet	MMMU	Avg
InternVL2.5-8B	64.4	84.6	62.8	56.0	67.0
InternVL2.5-8B-ULD [6]	66.5	86.2	64.1	60.8	69.4
InternVL2.5-8B-MOT [16]	68.4	87.5	65.0	62.3	70.8
InternVL2.5-8B-GenRecal	**74.9**	**89.5**	**73.2**	**68.1**	**76.4**

(g) Different LLMs

LLMs	MMLU	GPQA	MATH	Avg
Qwen2-7B [61]	70.8	38.4	48.6	52.6
Qwen2-7B-GenRecal (Llama-3-70B)	78.4	38.9	49.1	55.5
Llama-3-70B [19]	**82.5**	**39.5**	**50.4**	**57.5**
InternLM2.5-7B [8]	72.8	38.4	60.1	57.1
InternLM2.5-7B-GenRecal (Qwen2.5-72B)	83.7	43.6	61.8	63.0
Qwen2.5-72B [61]	**86.1**	**45.9**	**62.1**	**64.7**

(h) Different VLM Family

VLM	MM-Vet	MathVista	MMMU	Avg
Ovis1.6-Gemma-9B [39]	65.0	67.3	55.0	62.4
Ovis1.6-Gemma-9B-GenRecal (InternVL2.5-78B)	**74.0**	**75.3**	67.7	**72.3**
InternVL2.5-78B	72.3	72.3	**70.1**	71.6
InternVL2.5-8B [12]	62.8	64.4	56.0	61.1
InternVL2.5-8B-GenRecal (Ovis1.6-Gemma-27B)	**68.5**	**69.9**	61.2	**66.5**
Ovis1.6-Gemma-27B [39]	67.2	69.7	**61.7**	66.2

(i) Recent VLMs

VLM	MMB	MathVista	MM-Vet	MMMU	Avg
Qwen3-VL-8B [3]	86.8	77.2	74.5	69.6	77.0
Qwen3-VL-8B (Qwen3-VL-32B)	**89.9**	80.1	**77.6**	**72.7**	**80.1**
Qwen3-VL-8B (InternVL3.5-38B)	89.7	**80.3**	77.4	72.5	80.0
InternVL3.5-8B [60]	86.5	78.4	83.1	73.4	80.4
InternVL3.5-8B (Qwen3-VL-32B)	**89.6**	**81.5**	84.6	76.3	**83.4**
InternVL3.5-8B (InternVL3.5-38B)	89.4	81.3	**86.2**	**76.5**	**83.4**
Qwen3-VL-32B [3]	90.6	83.8	79.4	76.0	82.5
InternVL3.5-38B [60]	90.3	81.9	82.2	76.9	82.8

choice for users with limited training resources. Furthermore, Tab. 6(e) shows that fine-tuning the large VLM on our dataset yields further gains, indicating that the small VLM trained under the same setting has the potential to match or even surpass the large VLM's performance.

We also compare GenRecal with recent cross-tokenizer distillation methods in Tab. 6(f). These baselines perform distillation via cross-token matching in the logit space using the classical Wasserstein distance [6] or optimal transport [16], whereas GenRecal introduces the Recalibrator to align hidden features for semantic correspondence. GenRecal achieves a substantial improvement over these approaches, as discussed in the next section. Lastly, Tab. 6(g)–(i) demonstrate GenRecal's strong compatibility across different LLMs, VLM families, and recently released VLMs. In all cases, GenRecal consistently delivers superior performance, validating its general applicability to diverse model architectures.

4.4 Discussion

Representation Mismatch Beyond Tokenizer Mismatch. Even within a family the tokenizer can differ—InternVL2.5-8B uses InternLM2.5 [8] while InternVL2.5-78B uses Qwen2.5 [61], whereas Qwen2-VL-7B/72B [58] share the Qwen2 tokenizer (Fig. 1). More importantly, even with an *identical* tokenizer, a large teacher–student gap causes a *representation* mismatch: conventional distillation routes knowledge through the low-dimensional student head, creating a bottleneck. GenRecal instead distills through the high-capacity teacher head, with the Recalibrator warm-starting student features into the teacher space (Fig. 3). This is why it helps even under the same tokenizer: in Tab. 4 (Qwen2-VL-72B teacher, 7B/2B students), GenRecal beats both SFT and traditional KD—MiniLLM [23] (reverse KL), DistiLLM [28] (skewed KL), and LLaVA-KD [7] (3-stage KL)—by a clear margin, showing explicit feature alignment matters even with shared token types.

Cross-Token Matching. Cross-token matching takes two forms: (a) word-embedding matching, infeasible without retraining the decoder, and (b) semantic feature matching. For (b), ULD [6] and MOT [16] align logit-space probability mass via Wasserstein distance or optimal transport (Tab. 6(f)), but can-

Table 7: Comparison of computational cost (FLOPs and Training Time) and performance across cross-tokenizer distillation methods. We average performance across four benchmarks: MMB [36], MathVista [38], MM-Vet [63], and MMMU [65].

Method	Extra	FLOPs	Training Stages	Training Time	Avg. Perform.
GenRecal	Recalibrator	298.3×10^{12}	3	3.1+10.4+5.3=**18.8** hours	**76.4**
ULD [6]	Vocab Matching	296.1×10^{12}	1	11.1×2(Epochs)=22.2 hours	69.4
MOT [16]	Optimal Transport	296.1×10^{12}	1	12.5×2(Epochs)=25.0 hours	70.8

Table 8: Decomposing the source of GenRecal's improvement on Qwen2-VL-7B with the same-token-type teacher Qwen2-VL-72B, so that traditional logit-KD baselines remain valid. All rows use the same 9M data. Avg. is over MMB [36], MathVista [38], MM-Vet [63], and MMMU [65].

Setup (all use the *same* 9M data)	Avg.	Δ
Baseline Qwen2-VL-7B (no training)	64.3	–
+ 9M SFT (Stage-3 recipe only, no distill, no Recalib.)	66.3	+2.0
+ multi-stage SFT+logit KD (LLaVA-KD recipe, no Recalib.)	67.7	+3.4
+ GenRecal (Recalibrator)	**72.3**	+8.0

Table 9: Stage-wise checkpoints of GenRecal with InternVL2.5-8B (student) and InternVL2.5-78B (teacher), compared against an SFT-only baseline on the same data. Stage 1 trains only the Recalibrator and leaves the student unchanged.

Checkpoint	Trainable parameters	Data	MMB	MathVista	MM-Vet	MMMU	Avg
Baseline	—	—	84.6	64.4	62.8	56.0	67.0
After Stage 1	Recalibrator only (align)	9M	84.6	64.4	62.8	56.0	67.0
After Stage 2	Recalibrator + Student	9M	88.2	72.5	70.9	65.7	74.3
After Stage 3 (GenRecal full)	Student only (SFT polish)	6M	**89.5**	**74.9**	**73.2**	**68.1**	**76.4**
SFT only	Student only (SFT polish)	9M	86.5	65.2	65.0	59.9	69.2

not handle token-split indices and resort to zero-padding or truncation, losing information. GenRecal avoids logit-level matching altogether: the Recalibrator aligns features *before* the head so the teacher *VLM-head* reads student features directly, needing no token correspondence, probability sorting, padding, or transport cost. It adds only a small FLOPs fraction (Tab. 6(a)), so GenRecal's total cost is comparable to ULD/MOT yet reaches higher accuracy (Tab. 7; same data, InternVL2.5-8B student and -78B teacher).

Why a Non-Linear Recalibrator, and Where the Novelty Lies. Prior shared-space results rely on *linear* word-embedding maps [15,42,50], which act on input embeddings, not on the answer-conditioned hidden states after the *VLM-body*. Aligning these post-body features while preserving token order exceeds a linear map, so we use shallow transformer decoder blocks. The novelty is thus not the module itself—depth beyond two layers saturates (Tab. 6(b))—but the *stable recipe* that makes cross-tokenizer teacher-head distillation work.

Isolating the Recalibrator from the Recipe. To check whether the gains come from the Recalibrator or merely from the multi-stage recipe and final SFT, we disentangle them on Qwen2-VL-7B with the same-token-type teacher Qwen2-VL-72B (so logit-KD baselines stay valid), all on the same 9M data. The recipe alone (SFT+logit-KD, no Recalibrator) gives +3.4 Avg, while adding the Recalibrator adds a further +4.6 (Tab. 8). Nor is the gain from Stage-3

Table 10: Effect of training-data scale on GenRecal with InternVL2.5-8B (student) and InternVL2.5-78B (teacher), compared against SFT-only at full scale. Stage 3 uses a random ∼70% subsample of the Stage 2 data.

Setting	Align data	Distill data	Final SFT data	MMB	MathVista	MM-Vet	MMMU	Avg
Baseline (no train)	—	—	—	84.6	64.4	62.8	56.0	67.0
SFT only	—	—	9M	86.5	65.2	65.0	59.9	69.2
GenRecal @ 1M	1M	1M	0.7M	86.6	68.6	67.0	60.9	70.8
GenRecal @ 3M	3M	3M	2M	88.9	73.6	71.9	66.6	75.3
GenRecal @ 5M	5M	5M	3M	89.3	74.5	72.8	67.6	76.0
GenRecal @ 9M (full)	9M	9M	6M	**89.5**	**74.9**	**73.2**	**68.1**	**76.4**

SFT: after Stage 2, *before any final SFT*, GenRecal already reaches 74.3 Avg vs. 69.2 for SFT-only on the same 9M (Tab. 9). Nor is it mere data scale: GenRecal@1M (∼11% data) already beats SFT@9M, and @3M nears the full result (Tab. 10). Further analyses are provided in the supplementary material: open-weight teacher dependence and the SFT fallback (Appendix F), component-level ablations (Appendix G), the regularization formulation (Appendix H), error-profile and calibration (Appendix I), and a compute/fairness comparison against baselines (Appendix J). Note that GenRecal operates under *open-weight* teacher access, where the teacher's parameters and hidden features are available for training.

5 Conclusion

We present **Gen**eration after **Recal**ibration (GenRecal), a general-purpose distillation framework that transfers knowledge from large to small vision-language models (VLMs) even when they adopt different token types. Existing distillation methods assume that the teacher and student share the same token types, while GenRecal removes this constraint through a Recalibrator that aligns the student's hidden features into the teacher's representation space. Since the Recalibrator is discarded after training, inference preserves the student's original architecture and cost. Our analyses further indicate that these gains stem primarily from the recalibration mechanism rather than from the multi-stage recipe, and that feature alignment remains beneficial even when the teacher and student already share a tokenizer. For future work, we plan to extend the Recalibrator to intermediate layers to capture fine-grained, sequential knowledge, and to explore distillation from multiple VLM sources. We envision GenRecal as a pivotal and versatile framework for knowledge distillation.

Acknowledgements

We thank our colleagues at NVIDIA, in particular Sharath Turuvekere Sreenivas, for valuable discussions on cross-tokenizer and cross-model knowledge distillation. We further thank Sepehr Sameni and Daniel Korzekwa for experimenting with this approach on Minitron and Cosmos families; subsequently, Sharath Turuvekere Sreenivas and Pavlo Molchanov developed the follow-up X-Token [52], extending it to multi-teacher cross-tokenizer distillation on the logit space.

References

1. Abdin, M., Jacobs, S.A., Awan, A.A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Behl, H., et al.: Phi-3 technical report: A highly capable language model locally on your phone. arXiv preprint arXiv:2404.14219 (2024)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems **35**, 23716–23736 (2022)
3. Anthropic: The claude 3 model family: Opus, sonnet, haiku. `https://www.anthropic.com` (2024), `https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf`
4. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016), `https://arxiv.org/abs/1607.06450`
5. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
6. Boizard, N., Haddad, K.E., HUDELOT, C., Colombo, P.: Towards cross-tokenizer distillation: the universal logit distillation loss for LLMs. Transactions on Machine Learning Research (2025), `https://openreview.net/forum?id=bwRxXiGO9A`
7. Cai, Y., Zhang, J., He, H., He, X., Tong, A., Gan, Z., Wang, C., Bai, X.: Llava-kd: A framework of distilling multimodal large language models. arXiv preprint arXiv:2410.16236 (2024)
8. Cai, Z., Cao, M., Chen, H., Chen, K., Chen, K., Chen, X., Chen, X., Chen, Z., Chen, Z., Chu, P., et al.: Internlm2 technical report. arXiv preprint arXiv:2403.17297 (2024)
9. Cao, J., Zhang, Y., Huang, T., Lu, M., Zhang, Q., An, R., Ma, N., Zhang, S.: Move-kd: Knowledge distillation for vlms with mixture of visual encoders. arXiv preprint arXiv:2501.01709 (2025)
10. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., et al.: Are we on the right way for evaluating large vision-language models? arXiv preprint arXiv:2403.20330 (2024)
11. Chen, Y., Hu, H., Luan, Y., Sun, H., Changpinyo, S., Ritter, A., Chang, M.W.: Can pre-trained vision and language models answer visual information-seeking questions? arXiv preprint arXiv:2302.11713 (2023)
12. Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., et al.: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271 (2024)
13. Chen, Z., Wang, W., Tian, H., Ye, S., Gao, Z., Cui, E., Tong, W., Hu, K., Luo, J., Ma, Z., et al.: How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. arXiv preprint arXiv:2404.16821 (2024)
14. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Muyan, Z., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023)
15. Conneau, A., Lample, G., Ranzato, M., Denoyer, L., Jégou, H.: Word translation without parallel data. arXiv preprint arXiv:1710.04087 (2017)

16. Cui, X., Zhu, M., Qin, Y., Xie, L., Zhou, W., Li, H.: Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 23724–23732 (2025)
17. Dai, W., Lee, N., Wang, B., Yang, Z., Liu, Z., Barker, J., Rintamaki, T., Shoeybi, M., Catanzaro, B., Ping, W.: Nvlm: Open frontier-class multimodal llms. arXiv preprint (2024)
18. Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J.S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., et al.: Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. arXiv preprint arXiv:2409.17146 (2024)
19. Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
20. Feng, Q., Li, W., Lin, T., Chen, X.: Align-kd: Distilling cross-modal alignment knowledge for mobile vision-language model. arXiv preprint arXiv:2412.01282 (2024)
21. Fu, X., Hu, Y., Li, B., Feng, Y., Wang, H., Lin, X., Roth, D., Smith, N.A., Ma, W.C., Krishna, R.: Blink: Multimodal large language models can see but not perceive. arXiv preprint arXiv:2404.12390 (2024)
22. Gu, S., Zhang, J., Zhou, S., Yu, K., Xing, Z., Wang, L., Cao, Z., Jia, J., Zhang, Z., Wang, Y., et al.: Infinity-mm: Scaling multimodal performance with large-scale and high-quality instruction data. arXiv preprint arXiv:2410.18558 (2024)
23. Gu, Y., Dong, L., Wei, F., Huang, M.: Minillm: Knowledge distillation of large language models. In: The Twelfth International Conference on Learning Representations (2024)
24. Hong, W., Wang, W., Ding, M., Yu, W., Lv, Q., Wang, Y., Cheng, Y., Huang, S., Ji, J., Xue, Z., et al.: Cogvlm2: Visual language models for image and video understanding. arXiv preprint arXiv:2408.16500 (2024)
25. Hu, A., Xu, H., Ye, J., Yan, M., Zhang, L., Zhang, B., Li, C., Zhang, J., Jin, Q., Huang, F., et al.: mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. arXiv preprint arXiv:2403.12895 (2024)
26. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
27. Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., Farhadi, A.: A diagram is worth a dozen images. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 235–251. Springer (2016)
28. Ko, J., Kim, S., Chen, T., Yun, S.Y.: Distillm: Towards streamlined distillation for large language models. arXiv preprint arXiv:2402.03898 (2024)
29. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
30. Li, B., Ge, Y., Chen, Y., Ge, Y., Zhang, R., Shan, Y.: Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. arXiv preprint arXiv:2404.16790 (2024)
31. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)

32. Li, J., Li, D., Xiong, C., Hoi, S.: BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 12888–12900. PMLR (17–23 Jul 2022)
33. Li, X., Zhang, F., Diao, H., Wang, Y., Wang, X., Duan, L.Y.: Densefusion-1m: Merging vision experts for comprehensive multimodal perception. arXiv preprint arXiv:2407.08303 (2024)
34. Lin, J., Yin, H., Ping, W., Lu, Y., Molchanov, P., Tao, A., Mao, H., Kautz, J., Shoeybi, M., Han, S.: Vila: On pre-training for visual language models. arXiv preprint arXiv:2312.07533 (2023)
35. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023)
36. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)
37. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bkg6RiCqY7
38. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255 (2023)
39. Lu, S., Li, Y., Chen, Q.G., Xu, Z., Luo, W., Zhang, K., Ye, H.J.: Ovis: Structural embedding alignment for multimodal large language model. arXiv:2405.20797 (2024)
40. Masry, A., Long, D.X., Tan, J.Q., Joty, S., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning. arXiv preprint arXiv:2203.10244 (2022)
41. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
42. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013)
43. OpenAI: Gpt-4v(ision) system card (2023), https://openai.com/research/gpt-4v-system-card, Last accessed on 2024-02-13
44. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv e-prints (2019)
45. Rajbhandari, S., Rasley, J., Ruwase, O., He, Y.: Zero: Memory optimizations toward training trillion parameter models. In: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1–16. IEEE (2020)
46. Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., Houlsby, N.: Scaling vision with sparse mixture of experts. Advances in Neural Information Processing Systems **34**, 8583–8595 (2021)
47. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=B1ckMDqlg
48. Shi, M., Liu, F., Wang, S., Liao, S., Radhakrishnan, S., Huang, D.A., Yin, H., Sapra, K., Yacoob, Y., Shi, H., et al.: Eagle: Exploring the design space for multimodal llms with mixture of encoders. arXiv preprint arXiv:2408.15998 (2024)

49. Shu, F., Liao, Y., Zhuo, L., Xu, C., Zhang, G., Shi, H., Chen, L., Zhong, T., He, W., Fu, S., et al.: Llava-mod: Making llava tiny via moe knowledge distillation. arXiv preprint arXiv:2408.15881 (2024)
50. Smith, S.L., Turban, D.H., Hamblin, S., Hammerla, N.Y.: Offline bilingual word vectors, orthogonal transformations and the inverted softmax. arXiv preprint arXiv:1702.03859 (2017)
51. Soldaini, L., Kinney, R., Bhagia, A., Schwenk, D., Atkinson, D., Authur, R., Bogin, B., Chandu, K., Dumas, J., Elazar, Y., Hofmann, V., Jha, A., Kumar, S., Lucy, L., Lyu, X., Lambert, N., Magnusson, I., Morrison, J., Muennighoff, N., Naik, A., Nam, C., Peters, M., Ravichander, A., Richardson, K., Shen, Z., Strubell, E., Subramani, N., Tafjord, O., Walsh, E., Zettlemoyer, L., Smith, N., Hajishirzi, H., Beltagy, I., Groeneveld, D., Dodge, J., Lo, K.: Dolma: an open corpus of three trillion tokens for language model pretraining research. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 15725–15788. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024). https://doi.org/10.18653/v1/2024.acl-long.840, https://aclanthology.org/2024.acl-long.840/
52. Sreenivas, S.T., Hanasoge, A.V., Yang, M., Taghibakhshi, A., Muralidharan, S., Aithal, A., Molchanov, P.: X-token: Projection-guided cross-tokenizer knowledge distillation. arXiv preprint arXiv:2605.21699 (2026)
53. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
54. Sujet AI, Allaa Boutaleb, H.R.: Sujet-finance-qa-vision-100k: A large-scale dataset for financial document vqa (2024), https://huggingface.co/datasets/sujet-ai/Sujet-Finance-QA-Vision-100k
55. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490 (2019)
56. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
57. Tong, S., Brown, E., Wu, P., Woo, S., Middepogu, M., Akula, S.C., Yang, J., Yang, S., Iyer, A., Pan, X., et al.: Cambrian-1: A fully open, vision-centric exploration of multimodal llms. arXiv preprint arXiv:2406.16860 (2024)
58. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution (2024), https://arxiv.org/abs/2409.12191
59. Wang, W., Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Zhu, J., Zhu, X., Lu, L., Qiao, Y., et al.: Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. arXiv preprint arXiv:2411.10442 (2024)
60. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
61. Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)
62. Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., et al.: Minicpm-v: A gpt-4v level mllm on your phone. arXiv preprint arXiv:2408.01800 (2024)

63. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities. arXiv preprint arXiv:2308.02490 (2023)
64. Yu, W., Yang, Z., Ren, L., Li, L., Wang, J., Lin, K., Lin, C.C., Liu, Z., Wang, L., Wang, X.: Mm-vet v2: A challenging benchmark to evaluate large multimodal models for integrated capabilities. arXiv preprint arXiv:2408.00765 (2024)
65. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. arXiv preprint arXiv:2311.16502 (2023)
66. Yue, X., Zheng, T., Ni, Y., Wang, Y., Zhang, K., Tong, S., Sun, Y., Yu, B., Zhang, G., Sun, H., et al.: Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. arXiv preprint arXiv:2409.02813 (2024)
67. Yunxin Li, e.a.: Cognitive visual-language mapper: Advancing multimodal comprehension with enhanced visual knowledge alignment. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (2024)
68. Zhang, Y.F., Wen, Q., Fu, C., Wang, X., Zhang, Z., Wang, L., Jin, R.: Beyond llava-hd: Diving into high-resolution large multimodal models. arXiv preprint arXiv:2406.08487 (2024)