

Image-Text-to-Text Transformers Safetensors openbmb/RLAIF-V-Dataset

multilingual minicpmv feature-extraction minicpm-v vision ocr multi-image

video custom\_code conversational arxiv:2509.18154 arxiv:2403.11703

License: apache-2.0



Train ▾

Deploy ▾

Use this model ▾

Model card

Files

xet

Community 16

Downloads last month

77,046



Safetensors ⓘ

Model size 8.7B params

Tensor type BF16

Chat template

Files info

## ⚡ Inference Providers NEW

Image-Text-to-Text

This model isn't deployed by any Inference Provider.



16

Ask for provider support

## ↳ Model tree for openbmb/MiniCPM-V-4\_5

Adapters	1 model
Finetunes	3 models
Quantizations	3 models

## ▀ Dataset used to train openbmb/MiniCPM-V-4\_5

## ▀ Spaces using openbmb/MiniCPM-V-4\_5 20

## ▀ Collection including openbmb/MiniCPM-V-4\_5

### MiniCPM-o & MiniCPM-V

Collection

Multimodal models with leading perform... • 28 items • Updated 25 days ago • Δ 54

# A GPT-4o Level MLLM for Single Image, Multi Image and High-FPS Video Understanding on Your Phone

[GitHub](#) | [CookBook](#) | [Technical Report](#) | [Demo](#)

## ⌚ MiniCPM-V 4.5

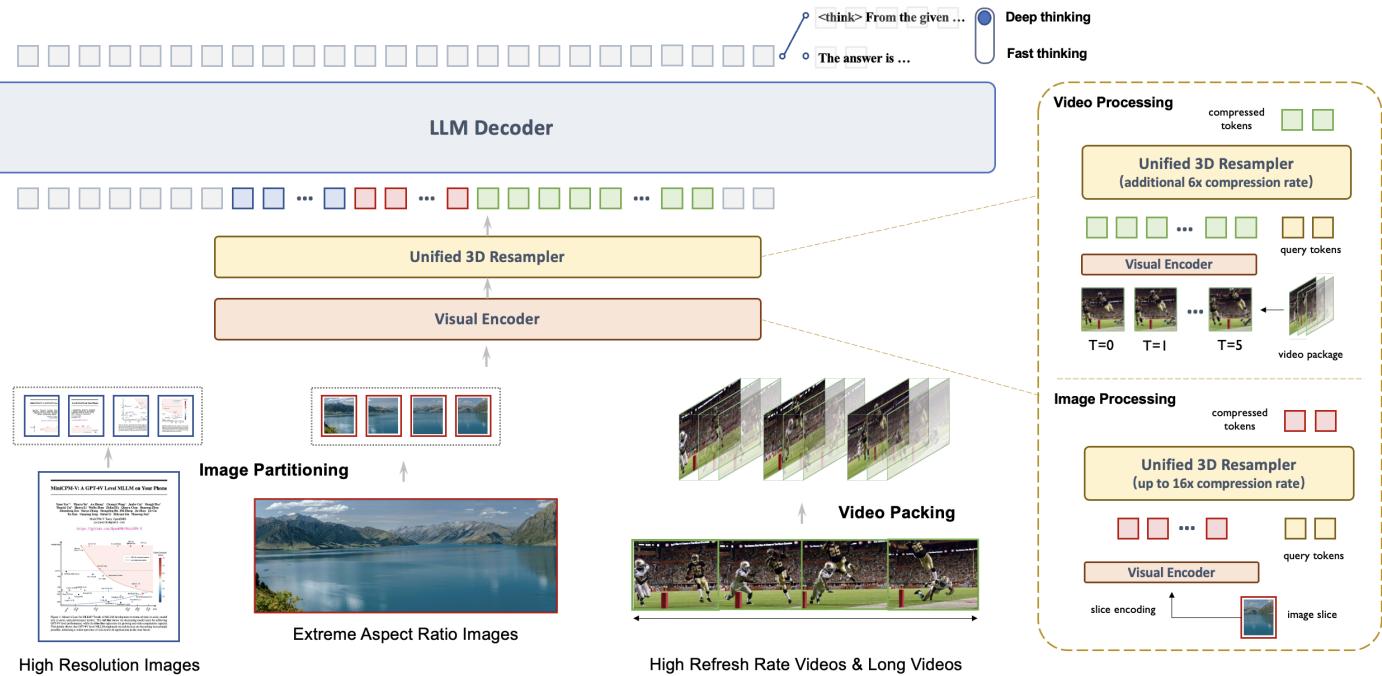
**MiniCPM-V 4.5** is the latest and most capable model in the MiniCPM-V series. The model is built on Qwen3-8B and SigLIP2-400M with a total of 8B parameters. It exhibits a significant performance improvement over previous MiniCPM-V and MiniCPM-o models, and introduces new useful features. Notable features of MiniCPM-V 4.5 include:

- **State-of-the-art Vision-Language Capability.** MiniCPM-V 4.5 achieves an average score of 77.0 on OpenCompass, a comprehensive evaluation of 8 popular benchmarks. **With only 8B parameters, it surpasses widely used proprietary models like GPT-4o-latest, Gemini-2.0 Pro, and strong open-source models like Qwen2.5-VL 72B** for vision-language capabilities, making it the most performant MLLM under 30B parameters.
- **Efficient High-FPS and Long Video Understanding.** Powered by a new unified 3D-Resampler over images and videos, MiniCPM-V 4.5 can now achieve 96x compression rate for video tokens, where 6 448x448 video frames can be jointly compressed into 64 video tokens (normally 1,536 tokens for most MLLMs). This means that the model can perceive significantly more video frames without increasing the LLM inference cost. This brings state-of-the-art high-FPS (up to 10FPS) video understanding and long video understanding capabilities on Video-MME, LVbench, MLVU, MotionBench, FavorBench, etc., efficiently.
- **Controllable Hybrid Fast/Deep Thinking.** MiniCPM-V 4.5 supports both fast thinking for efficient frequent usage with competitive performance, and deep thinking for more complex problem solving. To cover efficiency and performance trade-offs in different user scenarios, this fast/deep thinking mode can be switched in a highly controlled fashion.
- **Strong OCR, Document Parsing and Others.** Based on [LLaVA-UHD](#) architecture, MiniCPM-V 4.5 can process high-resolution images with any aspect ratio and up to 1.8 million pixels (e.g., 1344x1344), using 4x less visual tokens than most MLLMs. The model achieves **leading performance on OCRBench, surpassing proprietary models such as GPT-4o-latest and Gemini**.

**2.5.** It also achieves state-of-the-art performance for PDF document parsing capability on OmniDocBench among general MLLMs. Based on the latest RЛАIF-V and VisCPM techniques, it features **trustworthy behaviors**, outperforming GPT-4o-latest on MMHal-Bench, and supports **multilingual capabilities** in more than 30 languages.

- **Easy Usage.** MiniCPM-V 4.5 can be easily used in various ways: (1) llama.cpp and ollama support for efficient CPU inference on local devices, (2) int4, GGUF and AWQ format quantized models in 16 sizes, (3) SGLang and vLLM support for high-throughput and memory-efficient inference, (4) fine-tuning on new domains and tasks with Transformers and LLaMA-Factory, (5) quick local WebUI demo, (6) optimized local iOS app on iPhone and iPad, and (7) online web demo on server. See our Cookbook for full usages!

## ⌚ Key Techniques

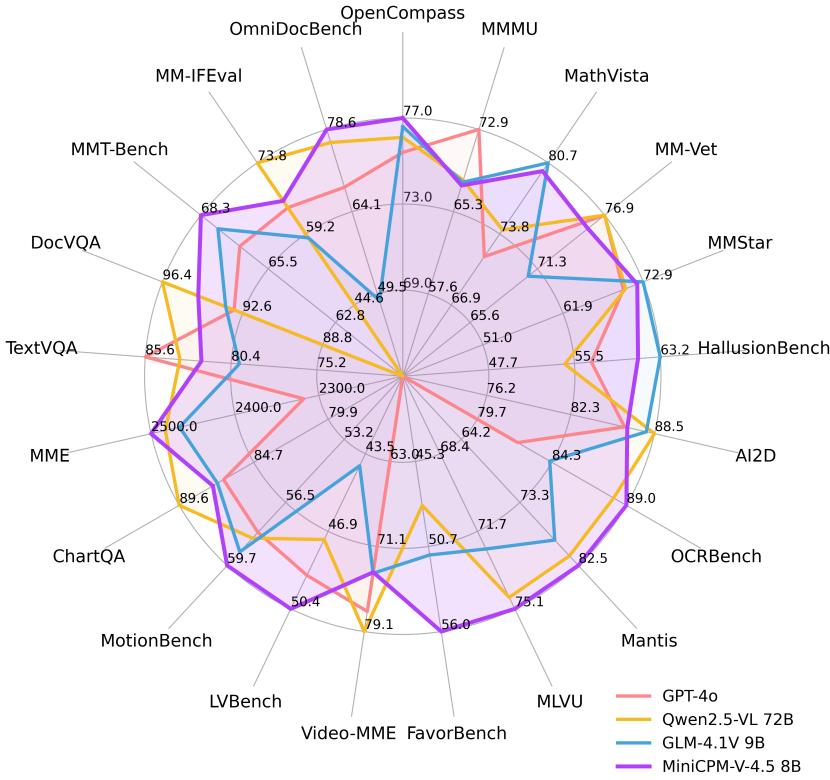


- **Architecture: Unified 3D-Resampler for High-density Video Compression.** MiniCPM-V 4.5 introduces a 3D-Resampler that overcomes the performance-efficiency trade-off in video understanding. By grouping and jointly compressing up to 6 consecutive video frames into just 64 tokens (the same token count used for a single image in MiniCPM-V series), MiniCPM-V 4.5 achieves a 96x compression rate for video tokens. This allows the model to process more video frames without additional LLM computational cost, enabling high-FPS video and long video

understanding. The architecture supports unified encoding for images, multi-image inputs, and videos, ensuring seamless capability and knowledge transfer.

- **Pre-training: Unified Learning for OCR and Knowledge from Documents.** Existing MLLMs learn OCR capability and knowledge from documents in isolated training approaches. We observe that the essential difference between these two training approaches is the visibility of the text in images. By dynamically corrupting text regions in documents with varying noise levels and asking the model to reconstruct the text, the model learns to adaptively and properly switch between accurate text recognition (when text is visible) and multimodal context-based knowledge reasoning (when text is heavily obscured). This eliminates reliance on error-prone document parsers in knowledge learning from documents, and prevents hallucinations from over-augmented OCR data, resulting in top-tier OCR and multimodal knowledge performance with minimal engineering overhead.
- **Post-training: Hybrid Fast/Deep Thinking with Multimodal RL.** MiniCPM-V 4.5 offers a balanced reasoning experience through two switchable modes: fast thinking for efficient daily use and deep thinking for complex tasks. Using a new hybrid reinforcement learning method, the model jointly optimizes both modes, significantly enhancing fast-mode performance without compromising deep-mode capability. Incorporated with RLPR and RLAIF-V, it generalizes robust reasoning skills from broad multimodal data while effectively reducing hallucinations.

## ⌚ Evaluation



Task	Benchmark	MiniCPM-V 4.5	Owen2.5-VL	Owen2.5-VL	InternVL3	GLM-4.1V	GPT-4o
Size		8B	7B	72B	8B	9B	-
Mode		hybrid	non-thinking	non-thinking	non-thinking	thinking	non-thinking
Comprehensive Multimodal	OpenCompass	<b>77.0<sup>†</sup></b>	70.5	76.1	73.6	76.6	75.4 <sup>‡</sup>
	MMVet	75.5 <sup>†</sup>	67.1	76.9	<b>81.3</b>	70.5 <sup>†</sup>	76.9 <sup>‡</sup>
	MMStar	72.1 <sup>†</sup>	63.9	70.5	68.2	<b>72.9</b>	70.2 <sup>‡</sup>
	MME	<b>2500</b>	2347	2483	2415	2466 <sup>†</sup>	2318*
	MBench V1.1	84.2	82.6	<b>87.8</b>	81.7	85.3	86.0 <sup>‡</sup>
	MMMU	67.7 <sup>†</sup>	58.6	68.2	62.7	68.0	<b>72.9<sup>‡</sup></b>
STEM	MathVista	79.9 <sup>†</sup>	68.2	74.2	71.6	<b>80.7</b>	71.6 <sup>‡</sup>
	AI2D	86.5	83.9	<b>88.5</b>	85.2	87.9	86.3 <sup>‡</sup>
	MathVerse MINI	58.8 <sup>†</sup>	49.2	47.3	39.8	<b>68.4</b>	40.6
	LogicVista	57.0 <sup>†</sup>	44.1	55.7	44.1	<b>60.4</b>	52.8
	EMMA	34.8 <sup>†</sup>	28.6 <sup>*</sup>	-	-	<b>35.7<sup>†</sup></b>	32.4
	OCRBench	<b>89.0</b>	86.4	88.2	88.0	84.2	82.2 <sup>‡</sup>
Document, OCR & Chart	ChartQA	87.4	87.3	<b>89.5</b>	86.6	87.1 <sup>†</sup>	86.7
	TextVQA	82.2	84.9	83.5	80.2	79.9 <sup>†</sup>	<b>85.6<sup>*</sup></b>
	DocVQA	94.7 <sup>†</sup>	95.7	<b>96.4</b>	92.7	93.4 <sup>†</sup>	93.0
	OmniDocBench (EN) ↓	<b>0.175</b>	0.316	0.214	0.335*	0.460*	0.233
	OmniDocBench (ZH) ↓	<b>0.253</b>	0.399	0.261	0.390*	0.573*	0.399
	HallusionBench	61.2 <sup>†</sup>	52.9	54.6	49.9	<b>63.2</b>	57.0 <sup>‡</sup>
Hallucination	ObjHalBench (CHAIRs) ↓	<b>9.3<sup>†</sup></b>	13.7*	17.0*	11.3*	12.3*	-
	ObjHalBench (CHAIRs) ↓	<b>5.2<sup>†</sup></b>	7.7*	8.9*	6.5*	6.4*	-
	MMHal (Score)	<b>5.0<sup>†</sup></b>	4.1*	4.2*	4.2*	4.6*	-
	MMHal (Rate) ↓	<b>19.4<sup>†</sup></b>	31.6*	38.2*	24.3*	22.9*	-
	Mantis	<b>82.5<sup>†</sup></b>	74.7*	81.1*	70.1	78.8 <sup>†</sup>	-
	MMT-Bench	<b>68.3</b>	63.6	-	65.0	67.6	66.7*
Multi-Image & Real World & Instruction Following	RealWorldQA	72.1 <sup>†</sup>	68.5	75.7	70.8	70.7 <sup>†</sup>	<b>76.8<sup>*</sup></b>
	MM-IEEval	66.0	51.3*	<b>73.8*</b>	53.2*	58.4 <sup>†</sup>	64.6
	Video-MME (w/o subs)	67.9	65.1	<b>73.3</b>	66.3	68.2	71.9
	Video-MME (w/ subs)	73.5	71.6	<b>79.1</b>	68.9	73.6	77.2
	LVBench	<b>50.4</b>	45.3	47.3	44.1*	44.0	48.9
	Video Understanding	<b>75.1</b>	70.2	74.6	71.4	72.5 <sup>†</sup>	-
Video Understanding	MLVU (Avg)	<b>59.7</b>	53.0	58.3	58.1	59.0	58.0
	LongVideoBench (val)	63.9	56.0	60.7	58.8	<b>65.7</b>	-
	MotionBench	<b>56.0</b>	42.3	48.1	45.3	51.2 <sup>†</sup>	-
	FavorBench	-	-	-	-	-	-

<sup>†</sup> Deep thinking mode used, where the average score of three runs is reported for robust evaluation.

\* We evaluate officially released checkpoints by ourselves.

‡ GPT-4o-latest evaluation results from OpenCompass. Otherwise GPT-4o-1120 is used in evaluation, since GPT-4o-latest is only accessible via Web API.

## ⌚ Inference Efficiency

### OpenCompass

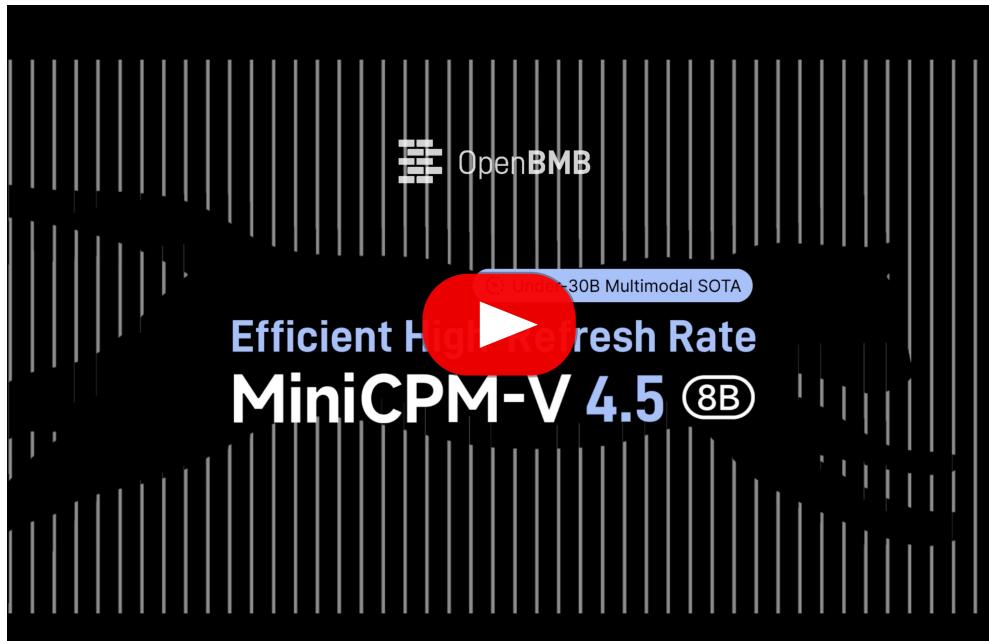
Model	Size	Avg Score ↑	Total Inference Time ↓
GLM-4.1V-9B-Thinking	10.3B	76.6	17.5h
MiMo-VL-7B-RL	8.3B	76.4	11h
MiniCPM-V 4.5	8.7B	77.0	7.5h

### Video-MME

Model	Size	Avg Score ↑	Total Inference Time ↓	GPU Mem ↓
Qwen2.5-VL-7B-Instruct	8.3B	71.6	3h	60G
GLM-4.1V-9B-Thinking	10.3B	73.6	2.63h	32G
MiniCPM-V 4.5	8.7B	73.5	0.26h	28G

Both Video-MME and OpenCompass were evaluated using 8×A100 GPUs for inference. The reported inference time of Video-MME includes full model-side computation, and excludes the external cost of video frame extraction (dependent on specific frame extraction tools) for fair comparison.

## ⌚ Examples





I need to go to the nearest exit. Tell me which way should I take. And if I'm really in a hurry, what's the shortest time it would take me to get there?



MiniCPM-V 4.5

To help you navigate, I'll analyze the road signs visible in the image to determine your nearest exit and estimate travel time based on speed limits.

Looking at the green directional signs along the highway, there are two main exits mentioned:

- The first sign indicates an exit for "East Perth" and "Welshpool," which is 700 meters away.
- The second sign points towards "James St & Wellington St," with its exit being just 1 kilometer (or approximately 1000 meters) from where these signs are positioned.

Since both distances refer to upcoming exits after this point of observation, we can compare them directly. A distance of 700 meters is shorter than 1000 meters. Therefore, if you're looking for the *nearest* exit among those clearly indicated by prominent roadside signage that's coming up soon enough to be relevant when considering a turn off the freeway, it would be the **East Perth / Welshpool exit located 700 meters ahead** down the road.

Regarding how quickly you could reach that specific exit: Based on standard Australian traffic regulations and common driving speeds:

- There's a clear white circular sign showing a '100' inside it, indicating a maximum speed limit of 100 kilometers per hour (km/h).
- If traveling straight through without slowing significantly before reaching the exit ramp itself or stopping due to other factors like congestion
  - At 100 km/h, converting units appropriately, one meter per second equals about 3.6 km/h. So 100 km/h divided by 3.6 km/h per meter/second gives roughly 27.78 meters per second.
  - To cover 700 meters at around 27.78 meters per second takes approximately **25 seconds**.

So, assuming ideal conditions – no sudden stops, consistent speed near the posted limit, and immediate acceleration onto the correct lane upon approaching the end of the current section of roadway shown here – theoretically, under such circumstances only, getting safely past any remaining curves until exiting might take somewhere close to **about 25 seconds** more depending slightly also perhaps too on actual vehicle acceleration rates relative starting speed plus possibly slight variations inherent within real-world driving scenarios versus perfectly uniform motion though generally aiming toward efficiency while ensuring safety compliance naturally.



Hi, I'm really interested in fossils. Could you tell me more about the fossil in the picture?



MiniCPM-V 4.5

Absolutely! The image shows a beautifully preserved fossil of an Archaeopteryx, often referred to as 'Urvogel' or 'first bird.' Discovered initially in 1861 in Solnhofen, Germany, it's one of the most significant finds for understanding avian evolution.

Archaeopteryx is crucial because it bridges the gap between dinosaurs and birds, showing features like feathers (both flight and body covering), wings with fingers ending in claws, and a partially toothed beak—traits that link theropod dinosaurs closely related to birds. Its discovery provided strong evidence supporting Darwin's theory of evolution by natural selection since its transitional characteristics were not found elsewhere at first but later confirmed through further discoveries globally over time.

This particular specimen appears well-preserved on limestone rock from the Jurassic period (~150 million years ago). It helps paleontologists understand how early birds evolved complex structures enabling powered flight—a major evolutionary milestone linking terrestrial reptiles back into arboreal lifestyles eventually leading up to modern-day aerial prowess seen among various bird species today!

Isn't this fascinating? If you have any specific questions about where else such specimens might be located around the world or their scientific significance within broader evolutionary contexts just let me know—I'd love to explore those topics too!

D

When it comes to retailing industry, we often remind the both part of realistic store and internet shopping. Both of them are all have their pros and cons, but according the picture, we can find out both of the internet sales counting and its profit are all growed up every year between twenty eighteen to twenty twenty one. The years rate began with twenty eighteen only 10.3%, next year 14.1%, and the next 20.3%, finally finished in twenty twenty one up to 24.5%. The sales profit also began with twenty eighteen only 2517 (million), next year 2893, and the next 3456, finally finished in twenty twenty one up to 4303. Therefore, we can find out the internet shopping is growed up between the four years. Begun 2019, according my observed, more of my friends change to internet shopping because of COVID-19. All above the results provided the picture is the realistic.

In my opinion, shopping on the internet can save many times to me, so I also do it when I

**Extract the information naturally from the image and format it as needed.**

### MiniCPM-V 4.5

When it comes to retailing industry, we often remind the both part of realistic store and internet shopping. Both of them are all have their pros and cons, but according to the picture, we can find out both of the internet sales counting and its profit are all growed up every year between twenty eighteen to twenty twenty one. The years () rate began with twenty eighteen only 10.3%, next year 14.1%, and the next 20.3%, finally finished in twenty twenty one up to 24.5%. The sales profit also began with twenty eighteen only 2517 (million), next year 2893, and the next 3456, finally finished in twenty twenty one up to 4303. Therefore, we can find out the internet shopping is growed up between the four years. Begun 2019, according my observed, () more of my friends change to internet shopping because of COVID-19. All above the results provided the picture is the realistic. In my opinion, shopping on the internet can save many times to me, so I also do it when I

D

Model	Fine-grained Perception		Coarse Perception		Creative Generation		OCR		Relation Reasoning		Attribute Reasoning		Logical Reasoning		Time series Reasoning		Average	
	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win
MiniGemini [31]	51.6	34.4	51.0	42.0	42.4	25.8	41.1	37.5	51.3	48.7	43.8	34.4	37.5	31.2	59.1	59.1	48.1	38.1
LLaVA 1.5 [33]	50.0	15.6	31.0	18.0	22.7	6.1	33.9	19.6	36.8	22.4	42.2	15.6	12.5	0.0	40.9	9.1	36.8	15.5
+ RLAIF-V	59.8	18.0	46.0	21.0	39.4	12.1	37.5	17.9	39.5	29.0	35.9	15.6	31.3	0.0	36.4	9.1	44.4	18.2
+ RLAIF-V BoN	66.4	20.5	51.0	25.0	47.0	12.1	35.7	16.1	38.2	25.0	37.5	18.8	37.5	0.0	54.6	0.0	48.7	18.8
OmniLMM [46]	54.1	15.6	56.0	25.0	43.9	6.1	33.9	14.3	35.5	25.0	48.4	17.2	6.3	0.0	36.4	0.0	45.4	16.5
+ RLAIF-V	65.6	26.5	55.0	29.7	54.0	18.7	32.1	16.7	56.6	39.5	55.7	25.0	29.2	6.3	63.6	21.2	54.8	25.9
+ RLAIF-V BoN	65.8	32.2	61.3	31.0	53.5	14.1	40.5	15.5	56.6	31.1	53.6	22.4	27.1	6.3	71.2	18.2	56.9	25.2
GPT-4V [43]	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0

Table 7. The trustworthiness win rate / overall win rate of different MLLMs on eight capabilities of ReFoMB test split. Trust.: trustworthiness win rate, Win.: overall win-rate.

Extract what's shown in the image, return tables in HTML.

D

### MiniCPM-V 4.5

Model	Fine-grained Perception		Coarse Perception		Creative Generation		OCR		Relation Reasoning		Attribute Reasoning		Logical Reasoning		Time series Reasoning		Average	
	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win	Trust.	Win
MiniGemini [31]	51.6	34.4	51.0	42.0	42.4	25.8	41.1	37.5	51.3	48.7	43.8	34.4	37.5	31.2	59.1	59.1	48.1	38.1
LLaVA 1.5 [33]	50.0	15.6	31.0	18.0	22.7	6.1	33.9	19.6	36.8	22.4	42.2	15.6	12.5	0.0	40.9	9.1	36.8	15.5
+ RLAIF-V	59.8	18.0	46.0	21.0	39.4	12.1	37.5	17.9	39.5	29.0	35.9	15.6	31.3	0.0	36.4	9.1	44.4	18.2
+ RLAIF-V BoN	66.4	20.5	51.0	25.0	47.0	12.1	35.7	16.1	38.2	25.0	37.5	18.8	37.5	0.0	54.6	0.0	48.7	18.8
OmniLMM [46]	54.1	15.6	56.0	25.0	43.9	6.1	33.9	14.3	35.5	25.0	48.4	17.2	6.3	0.0	36.4	0.0	45.4	16.5
+ RLAIF-V	65.6	26.5	55.0	29.7	54.0	18.7	32.1	16.7	56.6	39.5	55.7	25.0	29.2	6.3	63.6	21.2	54.8	25.9
+ RLAIF-V BoN	65.8	32.2	61.3	31.0	53.5	14.1	40.5	15.5	56.6	31.1	53.6	22.4	27.1	6.3	71.2	18.2	56.9	25.2
GPT-4V [43]	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0

Table 7. The trustworthiness win rate / overall win rate of different MLLMs on eight capabilities of ReFoMB test split. Trust.: trustworthiness win rate, Win.: overall win-rate.





## Extract what's shown in the image, return tables in HTML.

rather than requiring full re-optimization. By linearizing the network's behavior around specific weights, NTKs make the process of forgetting efficient, enabling the model to quickly process data deletion requests without the computational burden of extensive retraining and while preserving most of the model's predictive performance. Again, the unlearning mechanism proposed for this method is the same as Eq. (1). In our experiments, to show our methods performance on neural networks we utilized the mixed linear unlearning approach. Our proposed method in the next section is easily applicable to this setup.

### 4. Methodology

Given a differentiable convex loss  $\mathcal{L}$ , we can write the Taylor approximation of this around the optimal classifier  $w^*$  as follows:

$$\begin{aligned}\mathcal{L}(w) \approx & \mathcal{L}(w^*) + \nabla(w^*)^\top(w - w^*) \\ & + \frac{1}{2}(w - w^*)^\top H(w^*)(w - w^*)\end{aligned}$$

where the higher-order terms of the Taylor expansion are neglected here due to its relatively small magnitude. We define the loss difference  $\delta\mathcal{L}$  as follows:

$$\delta\mathcal{L} = \mathcal{L}(w) - \mathcal{L}(w^*)$$

Assuming this  $\delta\mathcal{L}$  is computed over the whole training data, we denote the loss difference with respect to the retain data  $x^r \in \mathcal{D}_r$  as  $\delta\mathcal{L}_r$ . So,

$$\begin{aligned}\delta\mathcal{L}_r \approx & \nabla_r(w^*)^\top(w - w^*) \\ & + \frac{1}{2}(w - w^*)^\top H_r(w^*)(w - w^*)\end{aligned}\quad (2)$$

Assuming that the training converges to the global optima  $w^*$ , we can safely assume that  $\nabla(w^*) = 0$ , which also means  $\nabla_r(w^*) + \nabla_f(w^*) = 0 \implies \nabla_r(w^*) = -\nabla_f(w^*)$ . Plugging this in Eq. (2) we get the following:

data. However, since we do not have access to  $\mathcal{D}_r$ , we can not explicitly compute  $\delta\mathcal{L}_r(w_i)$ . Instead, we can replace it with next best value which is  $\delta\mathcal{L}_f(w_i)$ . This is reasonable replacement since  $\delta\mathcal{L}(w_i) \leq L\|w_i - w^*\|$  where  $L$  is the Lipschitz constant corresponding to the loss. As a result both  $\delta\mathcal{L}_r(w_i)$  and  $\delta\mathcal{L}_f(w_i)$  can be upper bounded by  $L\|\delta w\| \rightarrow 0$ , for small perturbations. So with the small upper bound we can approximately say that both the quantities are very close to each other.

So we define an approximate version of  $f_i$  as follows:  $\tilde{f}_i(H_r) = \frac{1}{2}(\delta w)_i^\top H_r(w^*)(\delta w)_i - \nabla_f(w^*)^\top(\delta w)_i - \delta\mathcal{L}_f(w_i)$ . Our final objective becomes:

$$\tilde{\Psi}(H_r) = \frac{1}{m} \sum_{i=1}^m \left( \tilde{f}_i(H_r) \right)^2$$

Clearly the  $H_r$  is positive semi definite (PSD) for any convex loss functions. Based on this observation, we formulate the following optimization as a Semi Definite Program (SDP) as follows:

$$\begin{aligned}& \text{minimize} && \tilde{\Psi}(X) \\ & \text{subject to} && X \succeq 0\end{aligned}\quad (3)$$

Since we are approximating the value of  $\delta\mathcal{L}_r(w_i)$  instead of using the actual ground truth value, we anticipate that the solution to optimization problem Eq. (3) will be approximately close to the true retained Hessian  $H_r$ . In fact, we can bound the error between the true and estimated Hessian using the following lemma.

**Lemma 1.** Consider choosing  $\delta w \in \mathbb{R}^d$  where each element  $\delta w(j)$  of is sampled from  $\mathcal{N}(0, 1)$ . Assuming that the solution of the optimization Eq. (3) converges to  $\hat{H}_r$ , then the frobenius norm of the difference between the Hessian  $H_r$  (the actual ground truth Hessian with respect to  $\mathcal{D}_r$ ) and  $\hat{H}_r$  can be upper bounded as:

## MiniCPM-V 4.5

rather than requiring full re-optimization. By linearizing the network's behavior around specific weights, NTKs make the process of forgetting efficient, enabling the model to quickly process data deletion requests without the computational burden of extensive retraining and while preserving most of the model's predictive performance. Again, the unlearning mechanism proposed for this method is the same as Eq. (1). In our experiments, to show our methods performance on neural networks we utilized the mixed linear unlearning approach. Our proposed method in the next section is easily applicable to this setup.

### 4. Methodology

Given a differentiable convex loss  $\mathcal{L}$ , we can write the Taylor approximation of this around the optimal classifier  $w^*$  as follows:

$$\begin{aligned}\mathcal{L}(w) \approx & \mathcal{L}(w^*) + \nabla(w^*)^\top(w - w^*) \\ & + \frac{1}{2}(w - w^*)^\top H(w^*)(w - w^*)\end{aligned}$$

where the higher-order terms of the Taylor expansion are neglected here due to its relatively small magnitude. We define the loss difference  $\delta\mathcal{L}$  as follows:

$$\delta\mathcal{L} = \mathcal{L}(w) - \mathcal{L}(w^*)$$

Assuming this  $\delta\mathcal{L}$  is computed over the whole training data, we denote the loss difference with respect to the retain data  $x^r \in \mathcal{D}_r$  as  $\delta\mathcal{L}_r$ . So,

$$\begin{aligned}\delta\mathcal{L}_r \approx & \nabla_r(w^*)^\top(w - w^*) \\ & + \frac{1}{2}(w - w^*)^\top H_r(w^*)(w - w^*)\end{aligned}\quad (2)$$

Assuming that the training converges to the global optima  $w^*$ , we can safely assume that  $\nabla(w^*) = 0$ , which also means  $\nabla_r(w^*) + \nabla_f(w^*) = 0 \implies \nabla_r(w^*) = -\nabla_f(w^*)$ . Plugging this in Eq. (2) we get the following:

data. However, since we do not have access to  $\mathcal{D}_r$ , we can not explicitly compute  $\delta\mathcal{L}_r(w_i)$ . Instead, we can replace it with next best value which is  $\delta\mathcal{L}_f(w_i)$ . This is reasonable replacement since  $\delta\mathcal{L}(w_i) \leq L\|w_i - w^*\|$  where  $L$  is the Lipschitz constant corresponding to the loss. As a result both  $\delta\mathcal{L}_r(w_i)$  and  $\delta\mathcal{L}_f(w_i)$  can be upper bounded by  $L\|\delta w\| \rightarrow 0$ , for small perturbations. So with the small upper bound we can approximately say that both the quantities are very close to each other.

So we define an approximate version of  $f_i$  as follows:

$$\tilde{f}_i(H_r) = \frac{1}{\alpha}(\delta w)^\top H_r(w^*)(\delta w)_i - \nabla_f(w^*)^\top (\delta w)_i - \delta\mathcal{L}_f(w_i).$$

Our final objective becomes:

$$\tilde{\Psi}(H_r) = \frac{1}{m} \sum_{i=1}^m \left( \tilde{f}_i(H_r) \right)^2$$

Clearly the  $H_r$  is positive semi definite (PSD) for any convex loss functions. Based on this observation, we formulate the following optimization as a Semi Definite Program (SDP) as follows:

minimize  $\tilde{\Psi}(X)$

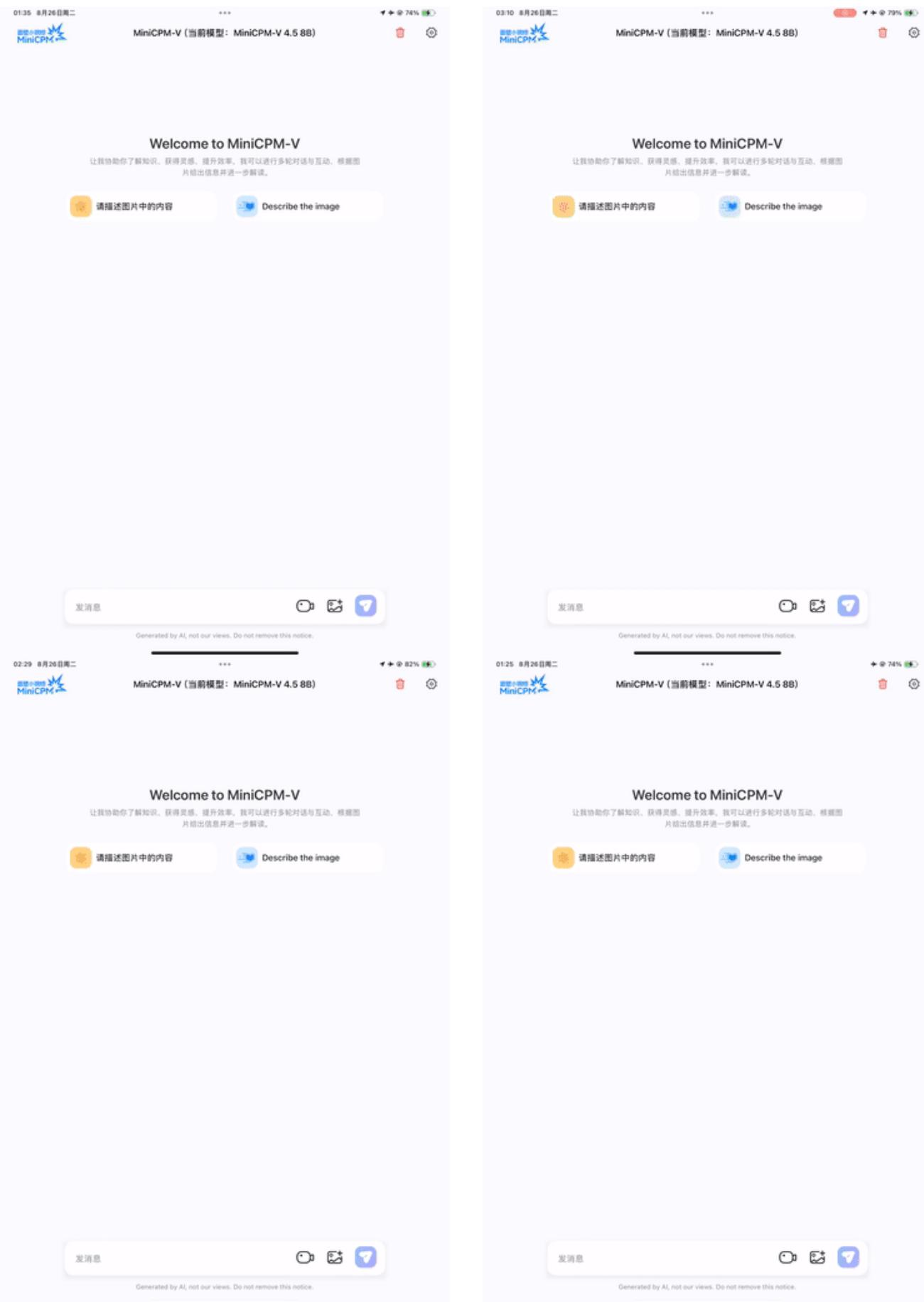
subject to  $X \succeq 0$

(3)

Since we are approximating the value of  $\delta\mathcal{L}_r(w_i)$  instead of using the actual ground truth value, we anticipate that the solution to optimization problem Eq. (3) will be approximately close to the true retained Hessian  $H_r$ . In fact, we can bound the error between the true and estimated Hessian using the following lemma.

**Lemma 1.** Consider choosing  $\delta w \in \mathbb{R}^d$  where each element  $\delta w(j)$  of is sampled from  $\mathcal{N}(0, 1)$ . Assuming that the solution of the optimization Eq. (3) converges to  $\hat{H}_r$ , then the frobenius norm of the difference between the Hessian  $H_r$  (the actual ground truth Hessian with respect to  $\mathcal{D}_r$ ) and  $\hat{H}_r$  can be upper bounded as:

We deploy MiniCPM-V 4.5 on iPad M4 with [iOS demo](#). The demo video is the raw screen recording without editing.



## 🔗 Framework Support Matrix

Category	Framework	Cookbook Link	Upstream PR	Supported since(branch)	Supported since(release)
Edge(On-device)	Llama.cpp	<a href="#">Llama.cpp Doc</a>	#15575(2025-08-26)	master(2025-08-26)	<a href="#">b6282</a>
	Ollama	<a href="#">Ollama Doc</a>	#12078(2025-08-26)	Merging	Waiting for official release
Serving(Cloud)	vLLM	<a href="#">vLLM Doc</a>	#23586(2025-08-26)	main(2025-08-27)	<a href="#">v0.10.2</a>
	SGLang	<a href="#">SGLang Doc</a>	#9610(2025-08-26)	Merging	Waiting for official release
Finetuning	LLaMA-Factory	<a href="#">LLaMA-Factory Doc</a>	#9022(2025-08-26)	main(2025-08-26)	Waiting for official release
Quantization	GGUF	<a href="#">GGUF Doc</a>	—	—	—
	BNB	<a href="#">BNB Doc</a>	—	—	—
	AWQ	<a href="#">AWQ Doc</a>	—	—	—
Demos	Gradio Demo	<a href="#">Gradio Demo Doc</a>	—	—	—

*“Note: If you'd like us to prioritize support for another open-source framework, please let us know via this short form.”*

## ⌚ Usage

If you wish to enable thinking mode, provide the argument `enable_thinking=True` to the chat function.

## ⌚ Chat with Image

```
import torch
from PIL import Image
```

```

from transformers import AutoModel, AutoTokenizer

torch.manual_seed(100)

model = AutoModel.from_pretrained('openbmb/MiniCPM-V-4_5', trust_remote_code=True,
    attn_implementation='sdpa', torch_dtype=torch.bfloat16) # sdpa or flash_attention
model = model.eval().cuda()
tokenizer = AutoTokenizer.from_pretrained('openbmb/MiniCPM-V-4_5', trust_remote_code=True)

image = Image.open('./assets/minicpmo2_6/show_demo.jpg').convert('RGB')

enable_thinking=False # If `enable_thinking=True`, the thinking mode is enabled.
stream=True # If `stream=True`, the answer is string

# First round chat
question = "What is the landform in the picture?"
msgs = [ {'role': 'user', 'content': [image, question]}]

answer = model.chat(
    msgs=msgs,
    tokenizer=tokenizer,
    enable_thinking=enable_thinking,
    stream=True
)

generated_text = ""
for new_text in answer:
    generated_text += new_text
    print(new_text, flush=True, end='')

# Second round chat, pass history context of multi-turn conversation
msgs.append({ "role": "assistant", "content": [generated_text] })
msgs.append({ "role": "user", "content": ["What should I pay attention to when trav..."] })

answer = model.chat(
    msgs=msgs,
    tokenizer=tokenizer,
    stream=True
)

```

```
generated_text = ""

for new_text in answer:
    generated_text += new_text
    print(new_text, flush=True, end='')
```

You will get the following output:

```
# round1
```

The landform in the picture is karst topography. Karst landscapes are characterized by their unique geological features, such as sinkholes, caves, and underground rivers.

This scene closely resembles the famous karst landscape of Guilin and Yangshuo in China.

```
# round2
```

When traveling to a karst landscape like this, here are some important tips:

1. Wear comfortable shoes: The terrain can be uneven and hilly.
2. Bring water and snacks for energy during hikes or boat rides.
3. Protect yourself from the sun with sunscreen, hats, and sunglasses—especially sunglasses.
4. Respect local customs and nature regulations by not littering or disturbing wildlife.

By following these guidelines, you'll have a safe and enjoyable trip while appreciating the natural beauty of the area.

## ⌚ Chat with Video

```
## The 3d-resampler compresses multiple frames into 64 tokens by introducing temporal context.
# To achieve this, you need to organize your video data into two corresponding sequences:
#   frames: List[Image]
#   temporal_ids: List[List[Int]].
```

```
import torch
from PIL import Image
from transformers import AutoModel, AutoTokenizer
from decord import VideoReader, cpu      # pip install decord
from scipy.spatial import cKDTree
import numpy as np
import math
```

```

model = AutoModel.from_pretrained('openbmb/MiniCPM-V-4_5', trust_remote_code=True,
    attn_implementation='sdpa', torch_dtype=torch.bfloat16) # sdpa or flash_attention
model = model.eval().cuda()
tokenizer = AutoTokenizer.from_pretrained('openbmb/MiniCPM-V-4_5', trust_remote_code=True)

MAX_NUM_FRAMES=180 # Indicates the maximum number of frames received after the video
MAX_NUM_PACKING=3 # indicates the maximum packing number of video frames. valid range [1, 3]
TIME_SCALE = 0.1

def map_to_nearest_scale(values, scale):
    tree = cKDTree(np.asarray(scale)[:, None])
    _, indices = tree.query(np.asarray(values)[:, None])
    return np.asarray(scale)[indices]

def group_array(arr, size):
    return [arr[i:i+size] for i in range(0, len(arr), size)]

def encode_video(video_path, choose_fps=3, force_packing=None):
    def uniform_sample(l, n):
        gap = len(l) / n
        idxs = [int(i * gap + gap / 2) for i in range(n)]
        return [l[i] for i in idxs]
    vr = VideoReader(video_path, ctx=cpu(0))
    fps = vr.get_avg_fps()
    video_duration = len(vr) / fps

    if choose_fps * int(video_duration) <= MAX_NUM_FRAMES:
        packing_nums = 1
        choose_frames = round(min(choose_fps, round(fps)) * min(MAX_NUM_FRAMES, video_duration))
    else:
        packing_nums = math.ceil(video_duration * choose_fps / MAX_NUM_FRAMES)
        if packing_nums <= MAX_NUM_PACKING:
            choose_frames = round(video_duration * choose_fps)
        else:
            choose_frames = round(MAX_NUM_FRAMES * MAX_NUM_PACKING)
            packing_nums = MAX_NUM_PACKING

```

```

frame_idx = [i for i in range(0, len(vr))]
frame_idx = np.array(uniform_sample(frame_idx, choose_frames))

if force_packing:
    packing_nums = min(force_packing, MAX_NUM_PACKING)

print(video_path, ' duration:', video_duration)
print(f'get video frames={len(frame_idx)}, packing_nums={packing_nums}')

frames = vr.get_batch(frame_idx).asnumpy()

frame_idx_ts = frame_idx / fps
scale = np.arange(0, video_duration, TIME_SCALE)

frame_ts_id = map_to_nearest_scale(frame_idx_ts, scale) / TIME_SCALE
frame_ts_id = frame_ts_id.astype(np.int32)

assert len(frames) == len(frame_ts_id)

frames = [Image.fromarray(v.astype('uint8')).convert('RGB') for v in frames]
frame_ts_id_group = group_array(frame_ts_id, packing_nums)

return frames, frame_ts_id_group

```

```

video_path="video_test.mp4"
fps = 5 # fps for video
force_packing = None # You can set force_packing to ensure that 3D packing is forced
frames, frame_ts_id_group = encode_video(video_path, fps, force_packing=force_packing)

question = "Describe the video"
msgs = [
    {'role': 'user', 'content': frames + [question]},
]

answer = model.chat(
    msgs=msgs,
    tokenizer=tokenizer,
    use_image_id=False,
)

```

```
    max_slice_nums=1,  
    temporal_ids=frame_ts_id_group  
)  
print(answer)
```

## 🔗 Chat with multiple images

- ▶ Click to show Python code running MiniCPM-V 4.5 with multiple images input.

## 🔗 In-context few-shot learning

- ▶ Click to view Python code running MiniCPM-V 4.5 with few-shot input.

## 🔗 License

### 🔗 Model License

- The MiniCPM-o/V model weights and code are open-sourced under the [Apache-2.0](#) license.
- To help us better understand and support our users, we would deeply appreciate it if you could consider optionally filling out a brief registration ["questionnaire"](#).

## 🔗 Statement

- As an LMM, MiniCPM-V 4.5 generates contents by learning a large amount of multimodal corpora, but it cannot comprehend, express personal opinions or make value judgement. Anything generated by MiniCPM-V 4.5 does not represent the views and positions of the model developers
- We will not be liable for any problems arising from the use of the MinCPM-V models, including but not limited to data security issues, risk of public opinion, or any risks and problems arising from the misdirection, misuse, dissemination or misuse of the model.

## 🔗 Key Techniques and Other Multimodal Projects

👉 Welcome to explore key techniques of MiniCPM-V 4.5 and other multimodal projects of our team:

[VisCPM](#) | [RLPR](#) | [RLHF-V](#) | [LLaVA-UHD](#) | [RLAIF-V](#)

## 🔗 Citation

If you find our work helpful, please consider citing our papers  and liking this project ❤️ !

```
@misc{yu2025minicpmv45cookingefficient,  
    title={MiniCPM-V 4.5: Cooking Efficient MLLMs via Architecture, Data, and Ti  
    author={Tianyu Yu and Zefan Wang and Chongyi Wang and Fuwei Huang and Wenshu  
    year={2025},  
    eprint={2509.18154},  
    archivePrefix={arXiv},  
    primaryClass={cs.LG},  
    url={https://arxiv.org/abs/2509.18154},  
}  
  
@article{yao2024minicpm,  
    title={MinicPM-V: A GPT-4V Level MLLM on Your Phone},  
    author={Yao, Yuan and Yu, Tianyu and Zhang, Ao and Wang, Chongyi and Cui, Junbo  
    journal={Nat Commun 16, 5509 (2025)},  
    year={2025}  
}
```