Table A1. **Training recipe**. Building upon VILA, we introduce two additional stages for NVILA: Stage 2, which focuses on pretraining the visual encoder to reduce performance loss due to spatial token compression, and Stage 5, which focuses on video instruction tuning to improve the model's long video capability.

	Visual Encoder (ViT)	Projector (MLP)	Token Processor (LLM)	LR
Initial	from [14]	random	from [15]	–
Stage 1	*frozen*	trainable	*frozen*	1×10^{-3}
Stage 2	trainable	trainable	*frozen*	5×10^{-5}
Stage 3	*frozen*	trainable	trainable	5×10^{-5}
Stage 4	trainable	trainable	trainable	2×10^{-5}
Stage 5	trainable	trainable	trainable	2×10^{-5}

A.1. Related Work

A.1.1. Visual Language Models

VLMs, especially proprietary ones, have advanced rapidly over the past two years. For example, OpenAI has upgraded from GPT-4V [64] to GPT-4o [12], achieving a 5–10% performance gain across image and video QA benchmarks. Google has extended the context length to 1M in Gemini Pro 1.5 [65], a significant improvement over Gemini 1.0 [66]. It now ranks at the top of the Video-MME leaderboard [59] for long video understanding. Anthropic has released Claude 3.5 [13], which demonstrates better benchmark scores than GPT-4o, showcasing notable improvements over Claude 3 [67]. Other proprietary models have similar advancements, such as Apple's upgrade from MM1 to MM1.5 [68] and xAI's upgrade from Grok-1.5 [52] to Grok-2 [69].

Meanwhile, open-source VLMs continue to evolve, improving at both the system/framework level [70] and the algorithm/recipe level [2], progressively narrowing the performance gap between proprietary and open-source models [5, 19, 71–73]. These recent advancements have led many open VLM models to claim performance levels comparable to, or even exceeding, leading proprietary models such as GPT-4V and GPT-4o. Some representative examples include InternVL2 [3], Qwen2-VL [5], LLaVA-OneVision [4], Llama 3.2 Vision [74], Molmo [75], NVLM [73], and MiniCPM-V [18].

Despite significant advancements in model performance, much less focus has been placed on enhancing the efficiency of training, inference, and fine-tuning for these models. This paper aims to explore how to develop VLMs that are not only highly accurate but also optimized for end-to-end efficiency.

A.1.2. Efficiency

Prior works such as [61, 76–82] have explored token reduction techniques in both spatial and temporal dimensions.

Table A2. **Temporal localization**. LITA results are from their original paper, while VILA-1.5 results are based on our reproduction. Our NVILA uses the same data mixture as VILA-1.5; the only difference is the backbone VLM.

			ActivityNet-RTL	
		#Frames	Mean IoU	Precision@0.5
LITA	7B	100	24.1	21.1
LITA	13B	100	28.6	25.9
VILA-1.5	8B	256	32.1	29.3
NVILA	8B	256	**34.8**	**32.1**

However, none have focused on reducing the number of tokens for a frontier Vision-Language Model (VLM). For dataset pruning, promising approaches have been proposed for selecting pretraining data for Large Language Models (LLMs), such as domain-mixing [83], sample-wise data selection [27, 84], and theory-driven optimal selection [28]. In this work, we specifically focus on pruning supervised fine-tuning (SFT) datasets for VLMs. Regarding low-precision training, FP8 training [85, 86] has gained popularity for LLMs, yet no prior work has demonstrated its feasibility for VLMs without sacrificing accuracy. Techniques such as pruning, distillation, and quantization are commonly applied to LLMs. [87, 88] apply pruning/distillation to LLM. However, their application to VLMs presents an open question: Should an LLM be pruned or distilled first before integrating a vision encoder, or should the VLM itself be pruned or distilled after training? Similarly, quantization techniques like AWQ [40] and GPTQ [89] are well-documented for LLMs, and VILA [2] has shown that AWQ can be directly applied to VLMs. However, little attention has been given to quantizing vision encoders, which becomes critical when handling higher-resolution images or videos due to the increased computational demands. Parameter-efficient fine-tuning methods such as LoRA [90], DoRA [91], QLoRA [92], and GaLoRA [93] are widely used for LLMs to reduce memory requirements. However, for VLMs, which combine a vision encoder with an LLM, efficient fine-tuning techniques are still underexplored. Addressing this gap is crucial for advancing VLM fine-tuning with limited computational resources.

A.2. More Capabilities

A.2.1. Temporal Localization

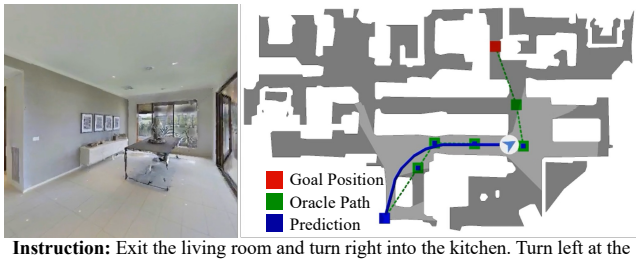
Following LITA, we also add support for temporal localization in NVILA. We add discrete time tokens to indicate the timestamps in the video, and use the smoothed cross entropy loss to train the model. From the results in Table A2, we can clearly see that NVILA substantially outperforms all baselines for all metrics.

Question: <image>What is the weather in this photo like?
Answer the question using a single word or phrase.
Answer: Snowy
DeltaLoss: 0.0343 (too easy ❌)

Question: <image>\nWhat color is the canopy?
A. white/yellow B. green/white
C. blue/white D. red/white
Answer with the option's letter from the given choices directly.
Answer: D
DeltaLoss: -1.916 (wrong answer ❌)

Question: <image> Which action depicted is a sign of respect?
Answer the question using a single word or phrase.
Answer: Hat over heart
DeltaLoss: 4.1605 (helpful ✅)

Figure A1. **Dataset pruning**. DeltaLoss visualizations in NVILA training: *Left*, *Middle*, and *Right* sections show examples that are too easy, distracting, and helpful for training, respectively.



Instruction: Exit the living room and turn right into the kitchen. Turn left at the end of the counter and wait in the room across the hallway slightly to the left.
Agent: The next action is turn left 15 degrees.

Instruction: Walk forward out of the room. Turn right and enter the other room and stop in front of the table.
Agent: The next action is move forward 75 cm.

Figure A2. **Robotic navigation**. NVILA deployed as a Vision-Language Navigation agent, navigating environments using language instructions and visual observations (Top: simulation, Bottom: real-world). The real-world setup features a Unitree Go2 robot equipped with a LiDAR sensor at the base of its head and an Intel RealSense Camera mounted on top. On the server side, an RTX 4090 GPU powers the NVILA-8B model, configured with an 8-frame context length for action generation.

A.2.2. Robotic Navigation

NVILA can serve as a strong foundation for robotic agents in Vision-Language Navigation [94] and empower real-time deployment on resource-constrained edge devices. At each time step t, the agent receives a language instruction and a video observation, plans the next action, and transitions to the next state $t+1$, where it receives a new observation. NVILA's efficient and flexible handling of multi-frame inputs enables seamless integration of historical and current observations into VLMs. The NaVILA framework [8] introduces a tailored navigation prompt and fine-tunes NVILA using navigation-specific SFT data curated from the simulator [95]. Quantitative results in Table A3 show that NVILA's straightforward design achieves state-of-the-art results on VLN-CE. Visual results of real-time deployment of the navigation model based on NVILA-8B on a single laptop GPU for navigation tasks are presented in Fig. A2. The entire system can operate seamlessly with an end-to-end (camera→GPU→action) pipeline running at 1Hz.

Table A3. **Robotic navigation**. All numbers are from NaVILA, except for those of NVILA. All models are provided with only RGB inputs. We refer the readers to NaVILA [8] for more details.

			R2R Val-Unseen			
		Obs.	NE ↓	OS ↑	SR ↑	SPL ↑
Seq2Seq	–	RGB	10.10	8.0	0.0	0.0
CMA	–	RGB	9.55	10.0	5.0	4.0
NaVid	7B	RGB	5.47	49.0	37.0	35.0
NVILA	8B	RGB	**5.43**	**60.4**	**53.3**	**48.8**

A.2.3. Medical Application

NVILA also offers transformative potential in the medical domain. Such integration promises advancements in diagnostic accuracy, clinical decision-making, and data interpretation. The NVILA-M3 framework [11] introduces a

Table A4. **Medical application**. Performance of best M3 model on key benchmarks is shown. Task-specific SOTA baselines and datasets are described in the experiments section [11]. Metrics for VQA is accuracy, for report generation BLEU-4 & ROUGE and for classification F1 score have been utilized

		VQA		Report Gen.		Classif.
		Rad	Path	CXR		CheXpert
Med-Gemini	–	78.8	83.3	20.5	28.3	48.3
VILA-M3	8B	84.7	91.0	21.1	32.0	**61.6**
NVILA	8B	**85.5**	**92.9**	**22.8**	**32.8**	61.1
Task-spfc. SOTA		84.2	91.7	15.4	30.6	51.5

novel approach by integrating multiple domain-expert models tailored to specific medical tasks, such as image segmentation and classification. These expert models are designed to extract and interpret intricate features that are otherwise difficult for general VLM's to discern. By coupling these specialized models with a vision-language learning paradigm, NVILA-M3 achieves enhanced performance, facilitating the learning of nuanced relationships between visual inputs and their textual annotations. This integration not only improves task-specific outcomes but also sets a foundation for the development of more robust and context-aware VLMs in the healthcare domain. NVILA-M3 indicated that an overall improvement of 9% can be achieved via usage of expert models over existing SOTA, a few key results can be observed in Table. A4. This underscores the importance of leveraging domain expertise to bridge the gap between generalized AI capabilities and the demands of specialized applications, demonstrating the potential for VLMs to revolutionize fields where precision and specificity are paramount.