MoTE: Mixture of Task Experts for Multi-Task Video Understanding

Muhammad Asad Ali[1,2]
muhammad_asad.ali@dfki.de
Umar Khan[1]
umaryousafzai9@gmail.com
Nadia Robertini[2]
nadia.robertini@dfki.de
Didier Stricker[1,2]
didier.stricker@dfki.de

[1] Department of Computer Science
University of Kaiserslautern-Landau (RPTU)
Kaiserslautern, Germany

[2] Augmented Vision Group
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany

Abstract

Procedural video-language models must solve heterogeneous tasks from the same visual evidence, including action recognition, forecasting, and procedure prediction. Dense transformer decoders share the same feed-forward networks across tasks, which can entangle task behavior and make controlled capability expansion difficult. Sparse Mixture-of-Experts (MoE) decoders provide conditional computation, but token-level learned routing is not naturally aligned with task-level procedural objectives. We propose **Mixture of Task Experts (MoTE)**, a decoder architecture that converts large language model feed-forward networks into task-specific experts while keeping the multimodal backbone shared. Each example follows one sample-level task route, so active task-expert computation remains independent of the number of stored task experts. We instantiate this design as **VideoLLM-MoTE** and evaluate it on five COIN benchmarks using explicit task routes. The five-expert model activates ~2B LLM parameters per sample and achieves higher average top-1 accuracy than recent VideoLLM baselines. Under the same expert topology, it improves over dense all-expert activation and learned sparse-routing controls. These results show that task-structured routing provides an interpretable and compute-efficient decoder alternative for multi-task video-language learning.

1 Introduction

Intelligent multimodal assistants are expected to follow human instructions over visual observations, from answering what is happening in a procedure to anticipating the next action or producing a concise procedural summary. Instruction tuning has become an effective way to align vision-language models with such diverse user requests, connecting pretrained visual encoders to language models through learned projectors and instruction-following supervision [9, 26, 33]. Video-language models extend this interface to temporal inputs, where a deployed assistant may receive either a complete clip or a live stream of frames [5, 6, 21, 30, 41]. In procedural video understanding, however, the challenge is not

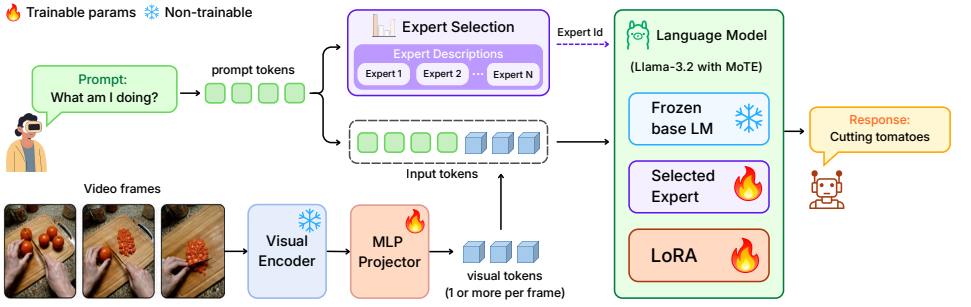

Figure 1: **VideoLLM-MoTE overview.** A user prompt and sampled video frames are converted into prompt and visual tokens before entering an LLM decoder modified with MoTE. Prompt-conditioned expert selection matches the prompt to registered expert descriptions and passes the selected expert id to the decoder. Given the input tokens and expert id, LLM generates a response. The visual encoder and base language model are non-trainable, while the MLP projector, LoRA adapters, and selected task experts are trainable.

only to encode time. The same visual evidence must support heterogeneous tasks such as recognizing the current action, forecasting future steps, identifying the goal, and producing procedure-level responses.

This task diversity creates a concrete modeling tension. Multi-task instruction tuning can improve generalization by exposing a model to many task formats [9, 45], but prior vision-language studies also show that mixing dissimilar tasks can create negative transfer when all tasks update the same parameters [14]. For procedural videos, task prompts often differ in temporal horizon and output structure even when the underlying frames are shared. A dense decoder therefore has to encode task-specific transformations inside the same feed-forward blocks used by every task.

Modern video-language systems have made progress on long and streaming inputs by improving memory, causal processing, token selection, and computation scheduling [5, 6, 18, 57]. These advances make video inputs more usable, but decoder-side task specialization often remains implicit: the same language-decoder path must express every prompt type. A useful multi-task video-language decoder should therefore preserve shared visual-language grounding while exposing task-specific computation in a controllable part of the model.

Sparse Mixture-of-Experts (MoE) transformers offer a natural starting point for conditional computation [12, 43, 48]. In language, vision, and large vision-language model settings, sparse experts can increase stored capacity while activating only a subset of parameters per input [25, 29]. Standard sparse MoE layers, however, usually route tokens or patches with learned gates. That granularity is effective for scaling, but it does not expose a stable task-level computation path for procedural prompts. Our setting uses known procedural task structure and asks whether the decoder itself can expose task-level routes.

We propose **Mixture of Task Experts (MoTE)**, a decoder architecture that makes task identity an explicit routing variable. MoTE converts each decoder feed-forward network (FFN) into task-specific experts and an always-active shared expert, while keeping the video-language backbone and attention path shared. Each video–prompt example follows one sample-level task route, so active task computation does not grow with the number of stored task experts. Because experts are explicit modules, the architecture can adapt to new tasks by

adding task experts while retaining previously learned knowledge in the shared backbone and existing experts. This serves as an architectural principle that we demonstrate is applicable across different domains. During training, task labels provide direct route supervision; at inference time, the route can be supplied externally or selected by matching the prompt to registered expert descriptions.

We evaluate this architecture as VideoLLM-MoTE on five COIN tasks. VideoLLM-MoTE outperforms recent COIN baselines while using a smaller active-parameter budget. To verify the cross-domain applicability of this architecture, we further conduct tests on document understanding, successfully demonstrating task separation for Key Information Extraction (KIE) and Optical Character Recognition (OCR) across two different datasets. Expert-addition analysis and the explicit expert-removal mechanism further illustrate how task-aligned FFN experts can serve as modular units.

Our contributions are:

1. We introduce MoTE, a modular decoder conversion that replaces shared LLM feed-forward networks with task-routed FFN experts, while retaining shared components for task-agnostic transformations.

2. We instantiate MoTE as VideoLLM-MoTE for multi-task procedural video understanding and show that task-level routing improves over recent VideoLLM baselines, and learned sparse-routing controls under the same expert topology.

3. We show that the same decoder expertization principle transfers to document analysis domain by converting GLM-OCR [11] into GLM-OCR-MoTE, separating OCR and receipt key information extraction (KIE) task routes.

4. We analyze the modularity of MoTE through expert addition and removal, showing that task experts can be introduced or removed as architectural modules while preserving previously learned knowledge.

2 Related Work

Streaming Procedural Video Understanding. Recent video-language work moves from offline video understanding toward causal, deployable assistants that must process frames as they arrive. VideoLLM-online [6] formalizes streaming dialogue, where a model must decide when to answer during an incoming video stream. VideoLLM-MoD [52] reduces redundant layer computation with mixture-of-depths routing while retaining richer visual tokens. Other systems improve long-context memory, causal temporal modeling, event-triggered reasoning, and always-on interaction [5, 17, 18, 35, 56, 60]. Recent benchmarks further show that online video understanding requires multiple procedural capabilities, including timestamped perception, anticipation, temporal reasoning, and multi-turn assessment [9, 21, 30, 38, 49, 51]. These requirements make procedural video-language modeling inherently multi-task. Existing streaming methods mainly improve memory, token selection, visual encoding, or computation depth. In contrast, MoTE studies how task-specific decoder computation can support heterogeneous procedural objectives while keeping the video-language backbone shared.

Method	Domain	Routing Level	Routing Type	Interpretability	Task-Aware
Sparse MoE [43]	Language	Token	Learned top-k	Low	No
GShard [28]	Language	Token	Learned top-k	Low	No
Switch Transformer [11]	Language	Token	Learned top-1	Low	No
V-MoE [44]	Vision	Patch	Learned top-k	Low	No
Sparse Upcycling [24]	Language/vision	Token	Expert Choice/Learned top-k	Low	No
LLaMA-MoE [53]	Language	Token	Learned top-k	Low	No
MoE-LLaVA [29]	Vision-language	Token	Learned top-k	Low	No
MoCLE [14]	LVLM instruction	Sample	Cluster-conditioned learned top-1	Medium	Yes
M^{3}ViT [27]	Multitask vision	Token/Patch	Task-conditioned learned top-k	Medium	Yes
Edge-MoE [46]	Multitask vision	Token/Patch	Task-conditioned learned top-k	Medium	Yes
Cross-task MoE [62]	Language	Sample	Task-conditioned learned top-k	Medium	Yes
Ours (MoTE)	Video-language	Sample	Task index / prompt-conditioned expert selection	High	Yes

Table 1: Comparison of MoE approaches. Prior sparse transformers primarily route tokens or patches with learned gates. Task-level methods make routing more aligned with task semantics by using task-conditioned features in the gating function, but they are not designed for modular video-language decoders. MoTE uses sample-level task routing over decoder FFN experts; the reported experiments use explicit task routes, while prompt-description matching provides an expert-selection component for inference.

Instruction Tuning and Task Conflict. Instruction-tuned vision-language models connect pretrained visual encoders to large language models through lightweight projectors, enabling general-purpose systems such as BLIP-2, LLaVA, and InstructBLIP [9, 26, 33]. Instruction tuning teaches these models to map diverse task templates to appropriate textual responses [45], and visual instruction tuning can improve held-out task generalization [9]. However, this shared-decoder design also creates a multi-task learning problem: heterogeneous prompts, objectives, and output formats compete for the same model capacity. This conflict is well studied in multi-task learning, where shared representations can improve transfer but may also cause negative transfer when task objectives interfere [7, 42, 47, 54, 63]. In procedural video-language modeling, this conflict is sharper because the same visual evidence supports recognition, forecasting, and procedure-level reasoning, yet each task requires different temporal decisions and output formats. MoTE targets this setting by keeping the video-language backbone shared while assigning task-specific transformations to decoder FFN experts.

Sparse MoE and Task-Level Routing. Sparse Mixture-of-Experts transformers add conditional capacity through routed experts. Foundational MoE systems route tokens or image patches with learned gates for language and vision scaling [12, 25, 43, 48], while dense-to-sparse construction reduces the cost of building MoE models by copying or partitioning pretrained FFNs into experts [23, 29, 58]. These designs scale capacity effectively, but token- or patch-level gates are not naturally aligned with user-facing task boundaries. Task-aware MoE methods move routing closer to task semantics by using task indices, task representations, or task-conditioned features in the routing function. Cross-task MoE uses task representations to select layer-wise expert mixtures across NLP tasks [62], M^{3}ViT uses task-dependent routing for multi-task vision [27], and Edge-MoE focuses on efficient execution for task-aware ViT MoE models [46]. MoCLE addresses task conflict in instruction tuning with LoRA experts conditioned on instruction clusters [14, 19]. MoTE differs by placing specialization inside the LLM decoder FFN path of a video-language model: each video–prompt sample selects one task route, activating the corresponding FFN expert at each decoder layer while attention, the visual encoder, and the MLP projector remain shared. Table 1 summarizes these distinctions across routing level, routing type, interpretability, and task awareness.

3 Method

3.1 Overview

We study multi-task video-language learning when the training set provides a discrete task structure. Each example is a tuple (v,x,t,y), where v is a video or sampled frame sequence, x is a natural-language prompt, $t \in \{1,\dots,M\}$ is the task identity, and y is the target response. The learning objective is to model $p_\theta(y \mid v,x,t)$ while retaining the ability to answer at inference time when the task is either explicitly provided or only implied by the prompt.

As shown in Figure 1, VideoLLM-MoTE consists of a vision encoder, an MLP projector, and a large language model with MoTE FFN expert blocks. We introduce the architecture of VideoLLM-MoTE in Section 3.2, the training objective in Section 3.3, and modular task expert adaptation in Section 3.4.

3.2 Architecture of VideoLLM-MoTE

Shared Backbone. The shared backbone follows the standard video-language pattern shown in Figure 1. A pretrained vision encoder maps the sampled frames in v to a visual token sequence $Z = [z_1,\dots,z_P] \in \mathbb{R}^{P\times C}$, where P is the total number of visual tokens and C is the visual feature dimension. A trainable two-layer MLP projector f_{proj} maps these features into the LLM hidden size D, giving $U_v = f_{\text{proj}}(Z) \in \mathbb{R}^{P\times D}$. The prompt x is embedded by the language model token embedding layer, yielding $U_x \in \mathbb{R}^{N\times D}$ for N text tokens. The chat template marks visual positions with <v>; after projection, those placeholders are replaced by U_v, producing the decoder input sequence $H_0 = \text{Template}(U_x, U_v)$, which concatenates visual and text token embeddings. All samples share this visual encoder, MLP projector, language token embedding layer, and attention path. Further implementation details such as frame rate, resolution, visual-token count, and backbone scale are reported in Section 4.1.

MoTE decoder blocks. MoTE modifies the transformer decoder layer in the LLM. Only the FFN sublayer is changed, so task-dependent transformations can be selected without duplicating the full video-language model. Concretely, let the dense decoder contain L layers. In layer l, the original FFN is denoted by F_l. As shown in Figure 2(a), MoTE replaces F_l with a set of task experts

$$\mathcal{F}_l = \{F_l^1, F_l^2, \dots, F_l^M\}$$

and a shared expert F_l^{sh}. Each task expert has the same architecture as the original FFN and is associated with one task. The shared expert is active for every task and captures transformations that should remain common, while the selected task expert specializes to task-specific supervision and output formats.

For hidden states H_l, the shared pre-norm attention block produces

$$\widetilde{H}_l = H_l + \text{GQA}_l(\text{LN}(H_l)),$$

where GQA_l denotes grouped-query attention and $\text{LN}(\cdot)$ denotes layer normalization. Given a task route $r \in \{1,\dots,M\}$, the expert block computes

$$H_{l+1} = \widetilde{H}_l + \lambda_{\text{sh}} F_l^{\text{sh}}(\text{LN}(\widetilde{H}_l)) + \lambda_{\text{task}} F_l^r(\text{LN}(\widetilde{H}_l)).$$

The coefficients λ_{sh} and λ_{task} are expert scaling factors; in our runs the routed scaling factor is 1.0. As illustrated in Figure 2(b), both routed and shared experts are initialized as copies

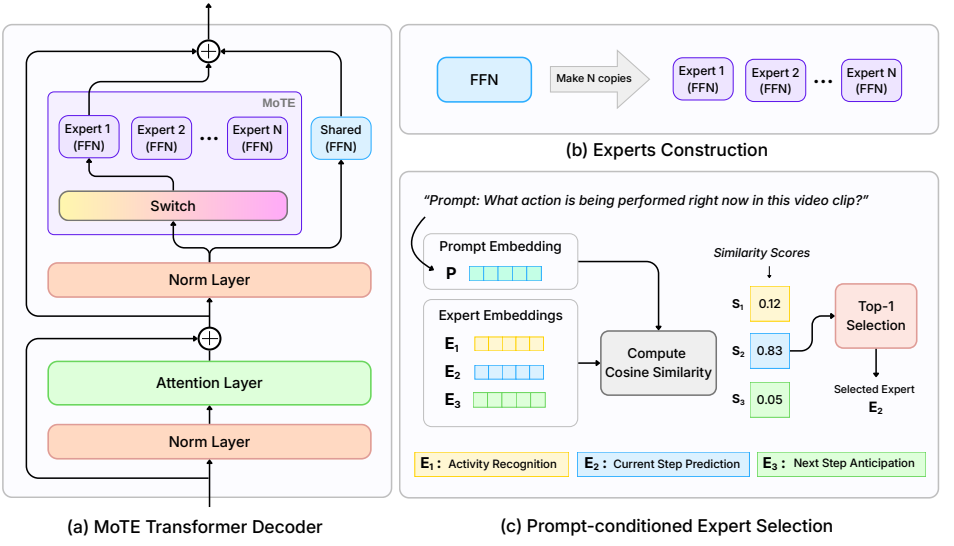

Figure 2: **MoTE decoder and expert selection.** (a) In each decoder layer, attention remains shared, while the FFN layer is replaced by a shared FFN expert and a switch-selected task FFN experts. Both expert outputs are added to the residual stream. (b) Task experts are initialized by copying the pretrained dense FFN into multiple task branches. (c) Prompt-conditioned expert selection embeds the prompt and registered expert descriptions, computes cosine similarities, and selects the top-1 expert for the full video–prompt sample.

of the pretrained dense FFN, following the dense-to-expert initialization principle used in sparse upcycling [23]. This avoids starting task experts from random transformations and allows the shared and task-specific paths to diverge during training.

This construction separates two roles that are entangled in a dense decoder. The shared expert learns common video-language transformations, while F_l^m can specialize to the supervision and output format of task m. Because only one task expert is activated for a sample, the number of stored experts can grow with the task set without requiring all experts to run for every input.

Task-route selection. MoTE routes the full video–prompt sample rather than individual tokens: all visual and textual tokens in an example pass through the same task expert at every decoder layer. This aligns computation with the user-facing task and avoids a learned-gate or load-balancing loss in the explicit task-index setting. For a training tuple (v,x,t,y), where v is the video input, x is the prompt, and $t \in \{1,\dots,M\}$ is the task identity among M task experts, the direct route is $r_{\text{idx}}(v,x,t) = t$.

During joint training, this known task identity provides route supervision. Only the selected task expert and the shared expert receive gradients for a given example, exposing each task expert to a consistent task distribution.

At inference time, the task identity may be unavailable. As illustrated in Figure 2(c), each expert index $m \in \{1,\dots,M\}$ is associated with a natural-language expert description d_m, such as "predict the next step" or "identify the overall procedure". Let $\phi(\cdot)$ be a frozen text encoder, and let $\bar{\phi}(z) = \phi(z)/\|\phi(z)\|_2$ denote the normalized embedding of any text

string z. We infer a prompt-conditioned route by nearest-neighbor matching between the prompt and the expert descriptions:

$$r_{\text{prompt}}(x) = \arg\max_{m \in \{1,\ldots,M\}} \bar{\phi}(x)^\top \bar{\phi}(d_m).$$

This selector is useful for modular deployment. In contrast to a learned gate whose output dimension is fixed by the experts observed during training, an expert description registry can add or remove natural-language expert descriptions while leaving the decoder parameters and remaining expert routes unchanged. We therefore use prompt-conditioned selection as an architectural component for inference when a task index is unavailable.

3.3 Training Objective

Following VideoLLM-online [6], the training objective supervises two behaviors in the same autoregressive video-language sequence. At response tokens, the model predicts the next answer token under the selected expert route. At streaming positions that don't require a response, it predicts EOS so it stays silent until needed. Let $p_\theta(\cdot \mid H_{\le j}, r)$ denote the next-token distribution after prefix $H_{\le j}$ under route r. We use two binary indicators: $l_j = 1$ when position j is a supervised language-response token, and $s_j = 1$ when position j is a frame position assigned streaming silence supervision. The loss is

$$\mathcal{L} = \frac{1}{N} \sum_{j=1}^{N} \Big(\underbrace{-l_{j+1} \log p_\theta(a_{j+1} \mid H_{\le j}, r)}_{\text{LM Loss}} \underbrace{- w s_j \log p_\theta(\text{EOS} \mid H_{\le j}, r)}_{\text{Stream Loss}} \Big),$$

where a_{j+1} is the next target token, w balances the streaming term, and N is the sequence length. We use $w = 1$ in reported runs. In our implementation, s_j is nonzero only for the last visual token of a frame when that frame is not immediately followed by a supervised language response. For all COIN offline benchmark tasks, stream messages are marked as non-learned positions, so $s_j = 0$ for frame tokens and the loss reduces to standard routed language modeling:

$$\mathcal{L}_{\text{COIN}} = -\frac{1}{N} \sum_{j=1}^{N} l_{j+1} \log p_\theta(a_{j+1} \mid H_{\le j}, r).$$

3.4 Modular Task Expert Adaptation

MoTE supports task adaptation by changing the task-specific FFN experts associated with the affected capability. New tasks can be introduced by attaching new routed experts, while tasks that are no longer needed can be removed from the active expert set without rewriting the shared video-language backbone or the remaining task experts.

Expert addition. Assume a model has been trained for tasks $\{1,\ldots,K\}$. To add task $K+1$, we insert a new expert F_l^{K+1} into every decoder layer and initialize it from the corresponding dense FFN or the shared expert. We then train on the new task using route $r = K+1$ while freezing the visual encoder, MLP projector, attention blocks, shared experts, and existing task experts. Only the new experts are updated:

$$\Theta_{\text{train}} = \{\theta(F_l^{K+1}) : l = 1,\ldots,L\}.$$

Method	Not use HowTo100M	COIN Benchmark Top-1 Accuracy↑ Step	Next	Proc.	Task	Proc.+	Avg.
Non-LLM procedural video methods							
ClipBERT [24]	✓	30.8	–	–	65.4	–	–
TimeSformer [3]	✗	46.5	34.0	17.0	85.3	40.1	44.6
Paprika [57]	✗	51.0	43.2	–	85.8	–	–
DistantSup [51]	✗	54.1	39.4	–	90.0	41.3	–
VideoTF [57]	✗	56.5	42.4	40.2	91.0	46.4	55.3
ProcedureVRL [56]	✗	56.9	46.8	–	90.8	–	–
VideoTaskGraph [1]	✗	57.2	40.2	–	90.5	–	–
LLM-based video-language methods							
VideoLLM-online-7B-v1 [6]	✓	59.8	48.1	47.9	92.1	52.9	60.2
VideoLLM-online-8B-v1+ [6]	✓	63.1	49.1	49.8	92.7	54.1	61.8
VideoLLM-MoD [57]	✓	63.4	49.7	49.8	92.8	53.3	61.8
VideoLLM-online-1B-v1+	✓	57.7	47.4	48.1	92.1	50.8	59.3
VideoLLM-online-3B-v1+	✓	59.5	47.9	48.3	92.4	51.4	59.9
VideoLLM-MoTE-1B+5E (ours)	✓	**65.1**	**50.3**	**50.5**	**94.4**	**54.2**	**62.9**

Table 2: Comparison with prior COIN baselines. *Not use HowTo100M* indicates whether the method avoids HowTo100M pretraining. The average is reported when all five task scores are available. Our method achieves 62.9% average accuracy outperforming other baselines.

This isolates new-task learning from previously learned task modules: old routes keep the parameters that produced their earlier behavior, which helps mitigate catastrophic forgetting during expansion and prevents knowledge overwriting.

Expert removal. When a task is outside the target deployment domain, or when an expert capability should be withdrawn (i.e. for compliance reasons), its experts can be dropped from every decoder layer and its description can be removed from the selector. This gives MoTE a direct capability-removal mechanism that is difficult to express in a dense decoder, where task behavior is distributed across shared parameters.

4 Experiments

4.1 Experimental Setup

Datasets and tasks. Following VideoLLM-online [6] and VideoLLM-MoD [57], we evaluate VideoLLM-MoTE in both offline procedural benchmarks and an online narration setting. For offline evaluation, we use COIN [51], an instructional-video dataset with temporally annotated procedural steps. We report five COIN benchmark tasks: *Step* recognition, *Next* step forecasting, *Task* recognition, *Proc.* procedure forecasting, and *Proc.+* task-conditioned procedure forecasting. We use VideoLLM-MoTE-1B+5E for this setting, with one routed expert for each COIN task. To evaluate task extension across datasets, we further use the Ego4D narration stream [16], where timestamped egocentric narrations define when the model should speak and what it should say. We first train an Ego4D narration route and then attach five COIN routes trained on the COIN tasks. The resulting VideoLLM-MoTE-1B+6E is evaluated as a single expanded model that preserves the original online narration capability while adding offline procedural reasoning tasks. Finally, we test whether

Model	LLM	Frame Strategy	Efficiency TFLOPs	Latency	Dec/s	Params (B) Total	Active	COIN Avg.↑
VideoLLM-online-7B-v1 [6]	Llama-2-7B	1	9.8	2.3	28.6	6.7	6.7	60.2
VideoLLM-online-8B-v1+ [6]	Llama-3-8B	1+3x3	98.7	2.8	27.8	8.0	8.0	61.8
VideoLLM-MoD [52]	Llama-3-8B	1+3x3	60.0	3.0	24.6	8.0	8.0	61.8
VideoLLM-online-1B-v1+	Llama-3.2-1B	1+3x3	15.2	1.3	54.7	1.2	1.2	59.3
VideoLLM-MoTE-1B+5E (ours)	Llama-3.2-1B	1+3x3	28.9	1.8	37.7	5.3	2.0	**62.9**

Table 3: Efficiency, LLM-only parameter summary and COIN average accuracy. Profiling uses one NVIDIA A100 with 600 frames, 64 prompt tokens, and 64 generated tokens. Parameter counts include token embeddings and transformer decoder layers, but exclude visual encoders and MLP projectors. VideoLLM-MoTE-1B+5E achieves the highest COIN average while activating only 2B LLM parameters per sample.

the same MoTE method transfers beyond video understanding. We convert GLM-OCR-0.9B into GLM-OCR-MoTE-0.9B+2E by retaining an OCR expert and adding a receipt key information extraction (KIE) expert. We evaluate direct image-to-structured-field extraction on SROIE [20] and CORD [40] datasets, while also measuring whether the original OCR route is preserved after adding the KIE route.

Implementation details. All reported VideoLLM-MoTE models are trained on four A100-80GB GPUs with videos sampled at 2 FPS and capped at 1200 frames. Our default COIN setting uses a SigLIP 2 ViT-L/16-384 visual encoder [53], a two-layer MLP projector, and a Llama-3.2-1B-Instruct decoder [15]. Given our limited computational resources, we select this 1B decoder as the default backbone rather than training larger variants; this keeps the study focused on routing behavior under a tractable training budget. Each frame is represented by one CLS token plus a 3×3 pooled spatial grid, giving 10 visual tokens per frame. We fine-tune the language model with LoRA [19] and keep the MLP projector trainable. The COIN model contains five routed task experts and one shared expert; each sample activates only the selected routed expert and the shared expert. Supplementary sections A.1– A.3 provide details about model parameters, training configuration and data sampling technique.

Evaluation Metrics. For COIN, we follow the generated-answer evaluation used by prior procedural video-language work [6, 57]: model outputs are decoded as text, normalized, mapped back to the task label space, and scored as top-1 accuracy. For *Proc.* and *Proc.+*, the generated numbered list is evaluated step-wise over the aligned future-action positions. We report the arithmetic mean over the five COIN tasks when all task scores are available. For Ego4D narration validation, following VideoLLM-online, we report perplexity (PPL) for language modeling quality, TimeDiff for temporal response alignment, and Fluency as a combined language-and-timing score. Lower is better for PPL and TimeDiff, while higher is better for Fluency. We refer readers to [6] for the detailed definition and computation of the online narration metrics.

Prompt-conditioned route selection. To evaluate inference without a supplied task index, we generate 50 prompt variants per COIN task and split them 70/10/20 for training, validation, and testing. A 22.7M-parameter MiniLM-L6-v2 selector reaches 100% test F1 for each of the five predefined intents, so the resulting end-to-end COIN scores equal those obtained with explicit routes. Selection takes 4.35 ms per prompt and each cached expert-

description embedding occupies 1.5 KiB. Supplementary Section A.6 gives protocol details and examples; ambiguous or compositional prompts remain outside this evaluation.

4.2 Comparison with Prior COIN Baselines

Table 2 compares VideoLLM-MoTE-1B+5E against prior COIN baselines under the same five-task protocol. The upper block contains earlier non-LLM procedural-video methods, while the lower block contains LLM-based VideoLLM variants. VideoLLM-MoTE-1B+5E achieves the highest accuracy on all five COIN tasks, improving the five-task average from 61.8% for the strongest 8B VideoLLM baseline to 62.9%.

The public 7B–8B baselines differ from our model in decoder scale, visual encoder, and training recipe. We therefore treat them as protocol-level comparisons rather than isolated architecture controls. Under this conservative interpretation, VideoLLM-MoTE-1B+5E remains competitive: despite using a smaller active LLM path, it exceeds the reported average accuracy of the larger VideoLLM baselines.

To reduce the scale confound, we also include VideoLLM-online-1B-v1+ and VideoLLM-online-3B-v1+, VideoLLM-online baselines trained from the same Llama-3.2 and SigLIP2 family initialization as our default model. In this comparison, the MoTE conversion achieves the highest five-task average of 62.9%. The controlled evidence for the MoTE decoder design therefore comes from two comparisons: the matched 1B/3B VideoLLM-online baselines in Table 2, and the fixed-topology routing controls in Table 8.

Table 3 makes the active-computation advantage explicit. The 8B VideoLLM baselines activate the full 8B-parameter LLM for each sample, whereas VideoLLM-MoTE-1B+5E activates only a 2B LLM path through the selected task expert and shared expert. Relative to VideoLLM-online-8B-v1+, MoTE reduces the fixed-request cost from 98.7 to 28.9 TFLOPs and latency from 2.8 to 1.8 s, while increasing decoding throughput from 27.8 to 37.7 tokens/s. It remains slower than the dense 1B baseline because it runs both shared and routed FFNs, in exchange for higher COIN accuracy.

4.3 Modular Task Adaptation via Expert Addition and Removal

Table 4 evaluates modular task adaptation through sequential expert addition and removal. In the addition phase, we first train the *Step* route and then introduce the *Next*, *Proc.*, *Task*, and *Proc.+* routes one at a time. At each expansion stage, only the newly added task route is trained, while previously trained routes are kept fixed. Consequently, earlier task accuracies remain unchanged across later stages, showing that adding a new task does not overwrite existing task-specific modules. The final addition stage reaches 61.0% average accuracy over the five active tasks, compared with 62.9% for the jointly trained VideoLLM-MoTE-1B+5E. This gap reflects a stability–plasticity trade-off: frozen expert addition prioritizes preservation of existing task behavior, whereas joint training allows shared components and task experts to adapt together for higher peak multi-task accuracy. In the removal phase, dropping an expert disables the corresponding task route, while retained routes keep their parameters and measured accuracies. This supports expert removal as a practical deployment mechanism for disabling task-specific capabilities.

COIN Benchmark	Stage 1 Step	Expert addition stages ✚Next	✚Proc.	✚Task	✚Proc.+	Expert removal stages ✖Proc.+	✖Task	✖Proc.	✖Next
Step	63.1	63.1	63.1	63.1	63.1	63.1	63.1	63.1	63.1
Next	–	46.2	46.2	46.2	46.2	46.2	46.2	46.2	–
Proc.	–	–	48.4	48.4	48.4	48.4	48.4	–	–
Task	–	–	–	92.8	92.8	92.8	–	–	–
Proc.+	–	–	–	–	54.6	–	–	–	–

Table 4: **Modular task adaptation through sequential expert addition and removal**. We first train a single *Step* expert, then add one task expert at a time for *Next*, *Proc.*, *Task*, and *Proc.+*. After all five experts are available, we remove them in reverse order. Accuracy is reported after each addition or removal stage, and a dash indicates that the corresponding task expert is not present. Stable scores for previously added tasks indicate that adding or removing other experts does not alter their routed inference paths.

4.4 Cross-Dataset Task Extension

Table 5 evaluates cross-dataset task extension in a single model. We start from an Ego4D narration model, which contains an online route for deciding when to speak and what narration to generate. We then expand the same model with five COIN experts and train only the newly added COIN routes. The shared video-language backbone, including the visual connector and decoder attention layers, is not retrained on COIN. Thus, the COIN experts operate on shared representations learned from Ego4D narration training, while only the task-specific decoder FFN routes are adapted to the new COIN objectives. This protocol tests whether a model trained for one dataset-specific task family can acquire a new set of procedural tasks while retaining the original online narration capability in the same checkpoint.

The split baseline rows clarify the evaluation scope. VideoLLM-online and VideoLLM-MoD report strong COIN and Ego4D results, but these scores are obtained from separate task-specific finetuned models rather than from one model that supports both task families. By contrast, VideoLLM-MoTE-1B+6E reports COIN and Ego4D metrics from a single expanded checkpoint that retains the Ego4D narration expert while adding five COIN experts. The expanded model remains competitive on the newly added COIN tasks and preserves online narration behavior, achieving the highest Fluency value in the table. The Ego4D-only baselines still obtain lower PPL and TimeDiff, indicating better standalone narration likelihood and timing. This result suggests that expert expansion can reuse a frozen shared backbone when the initial training domain and the added tasks have sufficient semantic overlap. Therefore, the comparison should be interpreted as evidence for backward-compatible task expansion across related video-language datasets, not as a direct claim of superior Ego4D-only narration quality or unrestricted transfer to unrelated domains.

4.5 Auxiliary Cross-Domain Evaluation

While our primary experiments focus on procedural video, we include an auxiliary evaluation on Key Information Extraction (KIE) from receipt images to verify if the MoTE FFN conversion mechanism applies outside video tokens. Unlike video benchmarks, this task requires parsing text-rich, static documents into structured outputs. We adapted the dense GLM-OCR-0.9B [□] backbone into the MoTE framework by introducing a shared expert alongside two task-specific routes: a retained Optical Character Recognition (OCR) route for the original capabilities and a newly trained KIE route. The KIE expert was trained on

Method	Model instance	COIN Benchmark Top-1 Accuracy↑ Step	Next	Proc.	Task	Proc.+	Ego4D Narration Stream Validation PPL↓	TimeDiff↓	Fluency↑
Separate finetuned models reported by prior work									
VideoLLM-online-7B-v1 [8]	COIN-only	59.8	48.1	47.9	92.1	52.9	–	–	–
VideoLLM-online-7B-v1 [8]	Ego4D-only	–	–	–	–	–	2.43	2.25	42.6
VideoLLM-online-8B-v1+ [8]	COIN-only	63.1	49.1	49.8	92.7	54.1	–	–	–
VideoLLM-online-8B-v1+ [8]	Ego4D-only	–	–	–	–	–	2.40	2.05	45.3
VideoLLM-MoD [53]	COIN-only	63.4	**49.7**	49.8	92.8	53.3	–	–	–
VideoLLM-MoD [53]	Ego4D-only	–	–	–	–	–	**2.41**	**2.04**	45.2
VideoLLM-MoTE-1B+6E expert expansion in a single model									
VideoLLM-MoTE-1B+6E (ours)	Single expanded model	**64.9**	47.6	**50.5**	**93.9**	**56.4**	2.57	2.25	**51.0**

Table 5: **Cross-dataset task extension from online narration to offline procedural tasks.** Prior VideoLLM-online and VideoLLM-MoD results are reported from separate COIN-only and Ego4D-only finetuned model instances. In contrast, the final row evaluates one expanded VideoLLM-MoTE-1B+6E checkpoint that starts from an Ego4D narration route and adds five COIN task routes. Ego4D validation reports PPL, TimeDiff, and Fluency.

Method	Input Modality	Micro-F1 (%) ↑
LayoutLLM-7B [24]	Image+Text+Layout	72.12
DocKylin [20]	Image + Text	69.60
Qwen-VL-7B [3]	Image + Text	58.59
LLaVA-1.5-7B [31]	Image + Text	3.83
LLaVAR-7B [51]	Image + Text	2.38
GLM-OCR-0.9B [1]	Image	55.04
GLM-OCR-MoTE-0.9B+2E (ours)	**Image**	**87.90**

Table 6: SROIE KIE Micro-F1 comparison for image-using models. GLM-OCR-MoTE-0.9B+2E reaches 87.90%, compared with 72.12% for the strongest image-using baseline in this compact table.

Method	Input Modality	Micro-F1 (%) ↑
UDOP [41]	Image+Text+Layout	97.60
LayoutLMv2 [48]	Image+Text+Layout	96.01
DocTr [18]	Image+Text+Layout	94.40
Donut [17]	Image + Text	91.60
GLM-OCR-0.9B [1]	Image	20.07
GLM-OCR-MoTE-0.9B+2E (ours)	**Image**	**95.79**

Table 7: CORD KIE Micro-F1 comparison for image-using models. GLM-OCR-MoTE-0.9B+2E raises the GLM-OCR baseline from 20.07% to 95.79%, approaching the strongest multimodal baseline while retaining the OCR route discussed in the text.

the SROIE and CORD datasets to generate structured field predictions directly from raw document images without an intermediate text-extraction pipeline.

The converted GLM-OCR-MoTE-0.9B+2E model achieves 87.90% Micro-F1 on SROIE KIE and 95.79% Micro-F1 on CORD, successfully raising the dense GLM-OCR CORD and SROIE baseline from 20.07% and 55.04% respectively to demonstrate effective adaptation to structured extraction. Crucially, the retained OCR route perfectly preserves the original model's performance on OmniDocBench v1.5 after conversion and KIE training. Full per-field metrics are provided in Supplementary Tables 14 and 15.

By leveraging the MoTE architecture, the resulting GLM-OCR-MoTE-0.9B+2E proves competitive with several open multimodal models, demonstrating that task-specific experts can successfully attach to non-video backbones to introduce entirely new capabilities without inducing catastrophic forgetting. We present these findings primarily as auxiliary evidence of the architecture's cross-domain applicability.

4.6 Ablation Study

4.6.1 Effect of Routing Granularity

Table 8 compares how the same five-expert topology behaves under different routing choices. The dense control removes sparsity by activating every routed expert for every sample. The token-level control learns an independent top-1 expert for each token. The sample-level

Method	COIN Benchmark Top-1 Accuracy↑ Step	Next	Proc.	Task	Proc.+	Avg.
VideoLLM-1B+5E-Dense	64.4 (0.0)	49.8 (0.0)	48.7 (0.0)	93.2 (0.0)	51.5 (0.0)	61.5 (0.0)
VideoLLM-MoE-Top1-token	63.7 (-0.7)	49.5 (-0.3)	48.5 (-0.2)	93.8 (+0.6)	51.7 (+0.2)	61.4 (-0.1)
VideoLLM-MoE-Top1-sample	64.9 (+0.5)	50.3 (+0.5)	49.2 (+0.5)	93.6 (+0.4)	52.3 (+0.8)	62.1 (+0.6)
VideoLLM-MoTE-1B+5E (ours)	**65.1** (+0.7)	**50.3** (+0.5)	**50.5** (+1.8)	**94.4** (+1.2)	**54.2** (+2.7)	**62.9** (+1.4)

Table 8: **Effect of Routing Granularity**. Numbers in parentheses are accuracy changes relative to the VideoLLM-1B+5E-Dense control; green and red indicate positive and negative changes, respectively. VideoLLM-MoE-Top1-token learns token-level routes, VideoLLM-MoE-Top1-sample learns one route for the full sample, and VideoLLM-MoTE-1B+5E uses explicit task-index routing.

control learns one top-1 expert for the full video–prompt sample, matching MoTE at the sample granularity but without explicit task supervision. MoTE instead uses the known task identity to select the task-aligned expert. The experiment therefore tests whether task-aligned sample routing is more effective than dense expert fusion or learned sparse routing at token and sample granularity.

With the same converted decoder capacity, VideoLLM-MoTE-1B+5E achieves 62.9% average accuracy, compared with 61.5% accuracy of VideoLLM-1B+5E-Dense for dense all-expert activation and 61.4% for token routing. This indicates that task structure is useful for choosing the FFN transformation: activating all routed experts dilutes specialization, while token-level routing breaks the task boundary across the generated sequence. The advantage is especially clear on procedure-oriented tasks, where the requested output depends on a longer temporal horizon and a more structured response format.

The sample-level baseline is closest to MoTE in routing granularity because both choose one expert route for the whole video–prompt sample. The difference is route semantics: sample-level MoE learns a latent top-1 route, whereas MoTE routes by task identity. Sample-level routing improves over dense activation, but MoTE gives the strongest average accuracy and the largest gains on structured procedure prediction. This suggests that, for task-structured procedural video understanding, MoTE is most effective when the route is aligned with the user-facing task.

4.6.2 Effect of Expert Construction Method

We compare the two expert construction methods in Table 9. Instead of copying the full pretrained FFN into each expert, **VideoLLM-MoTE-8B+5E-Split** uses the same 8B VideoLLM backbone and partitions each FFN into six equally sized experts, corresponding to one shared expert and five task experts, following LLaMA-MoE [68]. This keeps the experts disjoint inside each FFN rather than giving each task route a full copied FFN.

VideoLLM-MoTE-8B+5E-Split reaches 60.3% average accuracy, below the 62.9% average of the full-FFN-copy VideoLLM-MoTE-1B+5E. This suggests that task-routed video-language experts benefit from retaining the full pretrained FFN transformation in each route before specialization. Splitting neurons into small disjoint task groups is parameter efficient, but it reduces the per-task FFN capacity available for procedural output formats and long-horizon prediction.

Method	MoE Construction	COIN Benchmark Top-1 Accuracy↑ Step	Next	Proc.	Task	Proc.+	Avg.
VideoLLM-MoTE-8B+5E-Split	Splitting FFN	63.3	48.8	48.3	92.9	48.3	60.3
VideoLLM-MoTE-1B+5E (default)	Copying FFN	**65.1**	**50.3**	**50.5**	**94.4**	**54.2**	**62.9**

Table 9: **Effect of Expert Construction Method**. VideoLLM-MoTE-8B-5E-Split partitions each VideoLLM FFN into six equally sized expert groups (one shared expert and five task experts), while VideoLLM-MoTE-1B+5E initializes each shared and task expert as a full copied FFN. Copying FFN layer into multiple experts achieves better performance.

4.7 Discussion

The experiments support a simple claim: task identity is a useful structure for multi-task settings. Under the same expert topology, explicit task-index routing gives the strongest average accuracy across our tested benchmarks among the internal controls and uses the expert capacity more effectively than dense all-expert activation or learned one-expert sparse routing. The clearest gains appear on tasks with highly specialized objectives, where a single shared transformation is more likely to mix incompatible task requirements. This suggests that task routing helps most when task boundaries align with genuinely different computational needs.

The results also clarify where specialization occurs. In MoTE, attention and foundational representation alignment remain shared, while task-specific transformations are assigned to the feed-forward experts. This separates shared context modeling from task-dependent computation. This design sits between full parameter sharing and full model duplication: per-sample active expert computation is constant, but stored parameters grow linearly as task experts are added.

Finally, the expert-addition experiments expose a stability-plasticity tradeoff: freezing old routes preserves their measured behavior but gives lower peak accuracy than joint training. The auxiliary GLM-OCR-0.9B experiment provides a second, non-video test of the MoTE conversion. Adding a receipt-KIE route raises CORD micro-F1 from 20.07% to 95.79% from raw images while the retained route matches the reported OmniDocBench OCR metrics. Together, these results support backward-compatible modular expansion in the tested video and document settings.

5 Conclusion

We introduced MoTE (Mixture of Task Experts), a modular approach for extending LLM decoders by transforming their FFNs into task-specific experts while preserving a shared backbone. This design isolates task-specialized knowledge, activates only the expert required for a given task, and enables new capabilities to be added without modifying existing expert routes. Across five COIN video-understanding tasks, VideoLLM-MoTE consistently outperforms dense all-expert activation, learned sparse-routing variants, and reported larger VideoLLM baselines, despite using fewer active parameters. Our auxiliary document-understanding experiment further demonstrates the value of this modularity: MoTE can acquire new structured-output capabilities while fully retaining its foundational OCR performance. Together, these results show that explicit task specialization and route preservation provide an effective and parameter-efficient alternative to monolithic model expansion in the evaluated settings.

Acknowledgments

This work has been partially funded by the European Union, under Horizon Europe, grant number 101092889, project VICTOR-XR.

References

[1] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 67833–67846. Curran Associates, Inc., 2023.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. doi: 10.48550/arXiv.2308.12966. URL https://arxiv.org/abs/2308.12966.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 813–824. PMLR, 2021.

[4] Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. TemporalBench: Benchmarking fine-grained temporal understanding for multimodal video models. In *NeurIPS 2024 Workshop on Video-Language Models*, 2024. URL https://openreview.net/forum?id=bhuZn4URBH.

[5] Dibyadip Chatterjee, Edoardo Remelli, Yale Song, Bugra Tekin, Abhay Mittal, Bharat Bhatnagar, Necati Cihan Camgoz, Shreyas Hampali, Eric Sauser, Shugao Ma, Angela Yao, and Fadime Sener. Streaming VideoLLMs for real-time procedural video understanding. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22586–22598. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02097. URL https://doi.org/10.1109/ICCV51701.2025.02097.

[6] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. VideoLLM-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Chen_VideoLLM-online_Online_Video_Large_Language_Model_for_Streaming_Video_CVPR_2024_paper.html.

[7] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 794–803. PMLR, 2018. URL https://proceedings.mlr.press/v80/chen18a.html.

[8] Wei-Lin Chiang et al. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. LMSYS project page, 2023. URL https://vicuna.lmsys.org. Accessed 2023-04-14.

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N. Fung, and Steven C. Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 49250–49267. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9a6a435e75419a836fe47ab6793623e6-Paper-Conference.pdf.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

[11] Shuaiqi Duan, Yadong Xue, Weihan Wang, Zhe Su, Huan Liu, Sheng Yang, Guobing Gan, Guo Wang, Zihan Wang, Shengdong Yan, Dexin Jin, Yuxuan Zhang, Guohong Wen, Yanfeng Wang, Yutao Zhang, Xiaohan Zhang, Wenyi Hong, Yukuo Cen, Da Yin, Bin Chen, Wenmeng Yu, Xiaotao Gu, and Jie Tang. GLM-OCR Technical Report. *arXiv preprint arXiv:2603.10910*, 2026. doi: 10.48550/arXiv.2603.10910. URL https://arxiv.org/abs/2603.10910.

[12] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL https://jmlr.org/papers/v23/21-0998.html.

[13] Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Graliński. LAMBERT: Layout-aware language modeling for information extraction. In *Document Analysis and Recognition – ICDAR 2021*, volume 12821 of *Lecture Notes in Computer Science*, pages 532–547. Springer, 2021. doi: 10.1007/978-3-030-86549-8_34. URL https://doi.org/10.1007/978-3-030-86549-8_34.

[14] Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, James T. Kwok, and Yu Zhang. Mixture of cluster-conditional LoRA experts for vision-language instruction tuning. *IEEE Transactions on Image Processing*, 35: 3881–3892, 2026. doi: 10.1109/TIP.2026.3678087. URL https://doi.org/10.1109/TIP.2026.3678087.

[15] Aaron Grattafiori et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[16] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen

Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, June 2022.

[17] Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: VideoLLMs can watch and think simultaneously, 2026. URL https://arxiv.org/abs/2603.12262.

[18] Zhenghui Guo, Yuanbin Man, Junyuan Sheng, Bowen Lin, Ahmed Ahmed, Bo Jiang, Boyuan Zhang, Miao Yin, Sian Jin, Omprakash Gnawal, and Chengming Zhang. Event-VStream: Event-driven real-time understanding for long video streams, 2026. URL https://arxiv.org/abs/2601.15655.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[20] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. ICDAR2019 competition on scanned receipt OCR and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE, 2019. doi: 10.1109/ICDAR.2019.00244. URL https://doi.org/10.1109/ICDAR.2019.00244.

[21] Zhenpeng Huang, Xinhao Li, Jiaqi Li, Jing Wang, Xiangyu Zeng, Cheng Liang, Tao Wu, Xi Chen, Liang Li, and Limin Wang. Online video understanding: OVBench and VideoChat-Online. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3328–3338, 2025.

[22] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/8042_ECCV_2022_paper.php.

[23] Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=T5nUQDrM4u.

[24] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: CLIPBERT for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7331–7341, June 2021.

[25] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 2023. URL https://proceedings.mlr.press/v202/li23q.html.

[27] Hanxue Liang, Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, and Zhangyang Wang. M^3ViT: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. In *Advances in Neural Information Processing Systems*, volume 35, pages 28441–28457. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b653f34d576d1790481e3797cb740214-Paper-Conference.pdf.

[28] Haofu Liao, Aruni RoyChowdhury, Weijian Li, Ankan Bansal, Yuting Zhang, Zhuowen Tu, Ravi Kumar Satzoda, R. Manmatha, and Vijay Mahadevan. DocTr: Document transformer for structured information extraction in documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19584–19594, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Liao_DocTr_Document_Transformer_for_Structured_Information_Extraction_in_Documents_ICCV_2023_paper.html.

[29] Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. MoE-LLaVA: Mixture of experts for large vision-language models. *IEEE Transactions on Multimedia*, pages 1–14, 2026. doi: 10.1109/TMM.2026.3654458. URL https://doi.org/10.1109/TMM.2026.3654458.

[30] Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. StreamingBench: Assessing the gap for MLLMs to achieve streaming video understanding. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,

pages 12147–12151. IEEE, 2026. doi: 10.1109/ICASSP55912.2026.11463959. URL https://doi.org/10.1109/ICASSP55912.2026.11463959.

[31] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13853–13863, June 2022.

[32] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. doi: 10.48550/arXiv.2310.03744. URL https://arxiv.org/abs/2310.03744.

[33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. doi: 10.48550/arXiv.1907.11692. URL https://arxiv.org/abs/1907.11692.

[35] Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. AURA: Always-on understanding and real-time assistance via video streams, 2026. URL https://arxiv.org/abs/2604.04184.

[36] Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. LayoutLLM: Layout instruction tuning with large language models for document understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15630–15640, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Luo_LayoutLLM_Layout_Instruction_Tuning_with_Large_Language_Models_for_Document_CVPR_2024_paper.html.

[37] Medhini Narasimhan, Licheng Yu, Sean Bell, Ning Zhang, and Trevor Darrell. Learning and verification of task structure in instructional videos. In *ICCV 2023 Workshop on AI for Creative Video Editing and Understanding (CVEU)*, 2023. URL https://cveu.github.io/2023/papers/26.pdf.

[38] Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. OVO-Bench: How far is your Video-LLMs from real-world online video understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18902–18913, 2025.

[39] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. doi: 10.48550/arXiv.2303.08774. URL https://arxiv.org/abs/2303.08774.

[40] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. CORD: A consolidated receipt dataset for post-OCR parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019. URL https://openreview.net/forum?id=SJl3z659UH.

[41] Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video LLMs with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24045–24055, 2025.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

[43] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems*, volume 34, pages 8583–8595. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/48237d9f2dea8c74c2a72126cf63d933-Paper.pdf.

[44] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017. URL https://arxiv.org/abs/1706.05098.

[45] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

[46] Rishov Sarkar, Hanxue Liang, Zhiwen Fan, Zhangyang Wang, and Cong Hao. Edge-MoE: Memory-efficient multi-task vision transformer architecture with task-level sparsity via mixture-of-experts. In *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2023. doi: 10.1109/ICCAD57390.2023.10323651. URL https://doi.org/10.1109/ICCAD57390.2023.10323651.

[47] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/432aca3a1e345e339f35a30c8f65edce-Paper.pdf.

[48] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

[49] Guowei Tang, Tianwen Qian, Huanran Zheng, Yifei Wang, and Xiaoling Wang. StreamingEval: A unified evaluation protocol towards realistic streaming video understanding, 2026. URL https://arxiv.org/abs/2603.21493.

[50] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[51] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chengyu Chen, Jianfeng Gao, and Lijuan Wang. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264, 2023.

[52] Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. doi: 10.48550/arXiv.2307.09288. URL https://arxiv.org/abs/2307.09288.

[53] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Henaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL https://arxiv.org/abs/2502.14786.

[54] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. doi: 10.1109/TPAMI.2021.3054719. URL https://doi.org/10.1109/TPAMI.2021.3054719.

[55] Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. DocLLM: A layout-aware generative language model for multimodal document understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8529–8548, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.463. URL https://aclanthology.org/2024.acl-long.463/.

[56] Zichen Wen, Boxue Yang, Junlong Ke, Jiajie Huang, Chenfei Liao, Junxi Wang, Xuyang Liu, and Linfeng Zhang. EvoStreaming: Your offline video model is a natively streaming assistant, 2026. URL https://arxiv.org/abs/2605.10343.

[57] Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. VideoLLM-MoD: Efficient video-language streaming with mixture-of-depths vision computation. In *Advances in Neural*

Information Processing Systems, volume 37, 2024. doi: 10.52202/079017-3488. URL https://papers.nips.cc/paper_files/paper/2024/hash/c6a79e139ec4f371701ea8cc9e06018e-Abstract-Conference.html.

[58] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.201. URL https://aclanthology.org/2021.acl-long.201/.

[59] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020. doi: 10.1145/3394486.3403172. URL https://doi.org/10.1145/3394486.3403172.

[60] Yibin Yan, Jilan Xu, Shangzhe Di, Yikun Liu, Yudi Shi, Qirui Chen, Zeqian Li, Yifei Huang, and Weidi Xie. Learning streaming video representation via multitask training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9900–9912, 2025.

[61] Zhenyu Yang, Yuhang Hu, Zemin Du, Dizhan Xue, Shengsheng Qian, Jiahong Wu, Fan Yang, Weiming Dong, and Changsheng Xu. SVBench: A benchmark with temporal multi-turn dialogues for streaming video understanding. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Hz4BYVY8YM.

[62] Qinyuan Ye, Juan Zha, and Xiang Ren. Eliciting and understanding cross-task skills with task-level mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2567–2592. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-emnlp.189. URL https://doi.org/10.18653/v1/2022.findings-emnlp.189.

[63] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

[64] Jiaxin Zhang, Wentao Yang, Songxuan Lai, Zecheng Xie, and Lianwen Jin. DocKylin: A large multimodal model for visual document understanding with efficient visual slimming. *arXiv preprint arXiv:2406.19101*, 2024. doi: 10.48550/arXiv.2406.19101. URL https://arxiv.org/abs/2406.19101.

[65] Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. LLaVAR: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107*, 2023. doi: 10.48550/arXiv.2306.17107. URL https://arxiv.org/abs/2306.17107.

[66] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14825–14835, 2023.

[67] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10727–10738, June 2023.

[68] Tong Zhu, Xiaoye Qu, Daize Dong, Jiacheng Ruan, Jingqi Tong, Conghui He, and Yu Cheng. LLaMA-MoE: Building mixture-of-experts from LLaMA with continual pre-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15913–15923. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.890. URL https://doi.org/10.18653/v1/2024.emnlp-main.890.

A Supplementary Material

The supplementary material follows the experimental workflow. Sections A.1–A.3 document model accounting, training configuration, and data sampling. Section A.4 defines the evaluation protocols. Section A.5 presents the rendered model inputs, and Section A.6 groups the expert-description registry with the prompt-conditioned routing evaluation. Section A.7 then reports the expert-swap diagnostic. Section A.8 presents the auxiliary document architecture, protocol, and full benchmark results. Section A.9 concludes with scope limitations.

A.1 MoTE LLM Parameter Accounting

Table 3 reports LLM-only counts, excluding the shared visual encoder and projector. From the existing Llama-3.2-1B checkpoint, the required module totals are

$$P_{\text{dense}} = 1{,}235{,}814{,}400, \qquad P_{\text{FFN}} = 805{,}306{,}368,$$

where P_{FFN} sums the FFNs across all 16 layers. These checkpoint-derived totals are sufficient for the MoTE calculation.

Let T be the number of stored routed experts, S the number of shared experts, and k the number of routed experts activated for one sample. Because P_{dense} already contains one FFN path, replacing it with $T+S$ copies gives

$$\begin{aligned} P_{\text{total}}(T,S) &= P_{\text{dense}} + (T+S-1)P_{\text{FFN}}, \\ P_{\text{active}}(k,S) &= P_{\text{dense}} + (k+S-1)P_{\text{FFN}}. \end{aligned}$$

For VideoLLM-MoTE-1B+5E, $T=5$, $S=1$, and top-1 routing gives $k=1$:

$$\begin{aligned} P_{\text{total}} &= 1{,}235{,}814{,}400 + 5 \cdot 805{,}306{,}368 \\ &= 5{,}262{,}346{,}240 \simeq 5.3\text{B}, \\ P_{\text{active}} &= 1{,}235{,}814{,}400 + 805{,}306{,}368 \\ &= 2{,}041{,}120{,}768 \simeq 2.0\text{B}. \end{aligned}$$

These exact values round to the 5.3B total and 2.0B active counts in Table 3. Adding stored task routes increases total parameters linearly by P_{FFN}, whereas the active count remains fixed because each sample executes only the shared and one routed FFN path.

A.2 Training Configuration

Table 10 lists the main hyperparameters for the VideoLLM-MoTE experiments. For COIN, the tasks are trained as offline answer-generation problems: visual-frame positions are masked from the streaming silence loss and supervision covers the assistant answer plus EOS. Ego4D narration keeps the streaming silence term, so the narration route learns to produce short narrations at annotated response points and to remain silent elsewhere in the stream. In the main COIN results, task labels provide the route assignment directly.

A.3 Data Sampling Strategy

The routed MoTE decoder activates a task expert according to the expert id attached to each training sample. The expert buckets are not equally sized: for example, step recognition,

Hyper-parameter	Value
frame_fps / max_frames	2 / 1200
GPUs	4×A100-80GB
per_device_batch	1 train / 1 eval
gradient_acc_steps	8
Effective train batch	32 samples
num_train_epochs	10
learning_rate	1e-4
optimizer	AdamW
lr_scheduler_type	cosine
warmup_ratio	0.05
lora_targets	q/k/v/o, gate/up/down, lm_head
finetune_modules	MLP projector
lora_r / lora_alpha	128 / 256
lora_dropout	0.05
moe_router_aux_loss	0.0
attn_implementation	sdpa

Table 10: Hyper-parameters used for training VideoLLM-MoTE.

next-step forecasting, task recognition, and procedure forecasting can contribute different numbers of training examples. A uniform-over-experts schedule would repeatedly recycle smaller buckets and overtrain their experts, while larger buckets would be under-sampled relative to the evidence available for those tasks. Conversely, unconstrained mixed shuffling can give different data-parallel ranks different routed experts at the same local step.

Figure 3 illustrates the bucketed rank-aligned strategy used in our training runs. At the start of an epoch, samples are grouped by their expert id and each bucket is shuffled. At each local step, the sampler chooses one expert bucket with probability proportional to the number of fresh samples still unseen in that bucket. It then draws a global group of $G = bR$ indices from that selected bucket, where b is the per-rank local batch size and R is the number of data-parallel ranks. Distributed sharding splits this contiguous group across ranks, so every rank receives a local microbatch for the same routed expert at that step. When a selected bucket is exhausted, it is reshuffled and reused only as needed. This preserves the task-size bias of the training set while reducing both overexposure of small buckets and underexposure of large buckets.

A.4 Evaluation Protocols

COIN benchmarks. COIN outputs are decoded as free-form text and then mapped back to the task label space before scoring. For Step, Next, and Task, we lowercase the response, remove a trailing period, and first check for an exact match with the target label. If no exact match is found, the response is assigned to the closest valid class in the task taxonomy using Levenshtein distance. Top-1 accuracy is then computed over the mapped labels. For Proc. and Proc.+, the generated numbered list is split into individual lines, optional number prefixes are removed, and each predicted future step is mapped to the nearest COIN step label before computing step-wise top-1 accuracy over aligned future-action positions.

Ego4D narration stream. Ego4D narration is evaluated as an online stream rather than as an offline classification problem. Each narration turn has an annotated response time and a short target narration. We report three diagnostics following the prior online VideoLLM

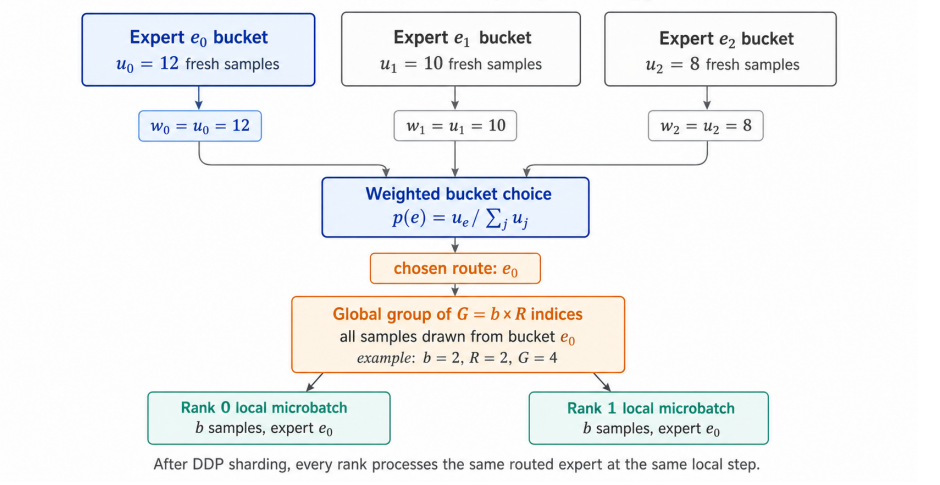

Figure 3: **Bucketed rank-aligned sampling.** The sampler groups a concatenated multi-task dataset into expert buckets. At each local training step, it samples one expert bucket with probability proportional to the number of fresh samples remaining in that bucket, draws one global group of $G = bR$ indices from the selected bucket, and lets distributed sharding split the group into rank-local microbatches. Here b is the local batch size and R is the number of data-parallel ranks.

protocol: PPL measures token-level narration likelihood, TimeDiff measures the absolute temporal offset between the predicted and annotated response time in seconds, and Fluency combines timing and token correctness into a percentage score. Lower PPL and TimeDiff are better, while higher Fluency is better.

Receipt KIE and OCR preservation. The auxiliary document experiment is evaluated separately from the video benchmarks. For SROIE Task 3, we report strict field-level F1 for company, date, address, and total, together with micro-F1 over the receipt fields in Table 14. For CORD, we report relaxed field-level F1 over the receipt categories used in the benchmark table and the corresponding micro-F1 in Table 15. The OCR route is checked on OmniDocBench v1.5 using the original GLM-OCR metrics: overall score, text edit distance, formula score, table score, table-structure score, and reading-order edit distance. These document metrics are used to test route adaptation and route preservation; they are not averaged with the video-language results.

A.5 Prompt Templates

We show examples of our prompts to illustrate the dialogue format. `<v>` markers are placeholders for visual tokens and orange text marks supervised assistant response during training.

The concrete targets below use official annotation labels from a `Make french fries` video. For Ego4D narration, the stream remains online: an initial frame chunk is followed by the narration instruction, then additional frame chunks and timestamped assistant narrations.

System Prompt applies to all routes

A multimodal AI assistant is helping users with some activities. Below is their conversation, interleaved with the list of video frames received by the assistant.

COIN Step Recognition Step route

<|begin_of_text|> [system prompt]
User: What is the action in the video? Format your answer concisely. No extra text output.
[<v>...<v>,<v>...<v>,...]
Assistant: *Cut potato into strips.*<|eot_id|>

COIN Next-Step Forecasting Next route

<|begin_of_text|> [system prompt]
User: What is the next action for the video? Format your answer concisely. No extra text output.
[<v>...<v>,<v>...<v>,...]
Assistant: *Soak them in water.*<|eot_id|>

COIN Task Recognition Task route

<|begin_of_text|> [system prompt]
User: What is the overall activity in the video? Format your answer concisely. No extra text output.
[<v>...<v>,<v>...<v>,...]
Assistant: *Make french fries.*<|eot_id|>

COIN Procedure Forecasting Proc. route

<|begin_of_text|> [system prompt]
User: What is the next 3 actions for the video? Format your answer concisely, listing each action on a new line with a number prefix. No extra text output.
[<v>...<v>,<v>...<v>,...]
Assistant: *1. Soak them in water.*
2. Dry strips.
3. Put in the oil to fry.<|eot_id|>

COIN Procedure Forecasting with Task Goal Proc.+ route

<|begin_of_text|> [system prompt]
User: To Make french fries, what is the next 3 actions for the video? Format your answer concisely, listing each action on a new line with a number prefix. No extra text output.
[<v>...<v>,<v>...<v>,...]
Assistant: *1. Soak them in water.*
2. Dry strips.
3. Put in the oil to fry.<|eot_id|>

Ego4D Narration Stream Narration route

<|begin_of_text|> [system prompt]
[<v>...<v>]
User: Please concisely narrate the video in real time. Use the tag 'C' to denote the camera wearer, and other letter tags, such as 'X', to denote other individuals in the scene.
[<v>...<v>,<v>...<v>]
Assistant: *C walks around a room.*<|eot_id|>
[<v>...<v>,<v>...<v>,<v>...<v>]
Assistant: *C picks up a wire from the floor.*<|eot_id|>

A.6 Prompt-conditioned Routing Evaluation

Table 11 records the natural-language registry used by the prompt-conditioned selector. These descriptions are not prompts shown to the decoder; they define the expert-selection component described in Section 3.2.

Task	Expert natural-language description
Current Step	Recognize the action currently being performed in the observed video segment; use this route for prompts asking what is happening now or which procedural step is visible.
Next Step	Predict the immediate next action that should follow the observed segment; use this route for short-horizon forecasting prompts asking what comes next.
Task	Identify the overall instructional goal or activity category of the video, independent of the current substep; use this route for prompts asking for the complete task being performed.
Procedure	Forecast a sequence of upcoming procedural actions from the current visual state without being given an explicit task goal; use this route for open-ended procedure-continuation prompts.
Task Procedure	Forecast upcoming procedural actions while conditioning on a provided goal or task name; use this route when the prompt states the target activity and asks for future steps.
Narration	Generate concise timestamp-aligned egocentric narrations for streaming video; use this route for online narration prompts that require deciding both what to say and when to say it.

Table 11: Task-to-description mapping used for prompt-conditioned expert selection.

Because no real user-prompt–task dataset is available for the five COIN intents, GPT-5.5 was used to generate 50 paraphrased prompts per task, giving 250 prompts in total. Each task is split 70/10/20 into 35 training, 5 validation, and 10 held-out test prompts. We instantiate the selector with MiniLM-L6-v2 (22.7M parameters) and evaluate the predicted top-1 task route. It reaches 100% test F1 for each task; consequently, end-to-end COIN scores under prompt-selected routing equal those reported with explicit task indices. Selection takes 4.35 ms per prompt, and each cached expert-description embedding occupies 1.5 KiB.

The held-out confusion matrix below makes the selector result explicit. All 10 test prompts for each intent are assigned to the correct route, yielding 100% precision, recall, and F1 for every class (50/50 correct overall).

Held-out prompt-routing confusion matrix (counts)

Actual route	Predicted route					F1 (%)
	Step	Next	Proc.	Task	Proc.+	
Step	10	0	0	0	0	100
Next	0	10	0	0	0	100
Proc.	0	0	10	0	0	100
Task	0	0	0	10	0	100
Proc.+	0	0	0	0	10	100
Macro average	50/50 correct					100

Rows denote ground-truth intents; columns denote selected expert routes.

Table 12 shows three of the 50 generated variants for each task. The examples use diverse phrasing while preserving clear task intent. This experiment is therefore a controlled selector sanity check rather than evidence for routing ambiguous, overlapping, or compositional requests.

Task route	Representative generated prompt variants
`Step`	**1.** Which action is visible in this clip? Use a minimal answer with no extra text. **2.** Recognize the step currently shown in the video. Use a minimal answer with no extra text. **3.** What action does this video segment depict? Answer briefly with only the action name. *. . . 47 additional variants*
`Next`	**1.** Name the step that follows the current video. Give just the next action label. **2.** What should the person do next? Keep it short and direct. **3.** What is the upcoming action after the video segment? Keep it short and direct. *. . . 47 additional variants*
`Task`	**1.** What larger task does this video demonstrate? Give just the overall task label. **2.** What is the person trying to accomplish overall? Return a short activity phrase only. **3.** State the overall procedure represented by the clip. Answer concisely with only the task name. *. . . 47 additional variants*
`Proc.`	**1.** Name the upcoming `{num_steps}` actions in order. Return only the ordered action list. **2.** What sequence of `{num_steps}` actions follows this segment? Keep the list concise and include no commentary. **3.** Determine the next `{num_steps}` steps after the shown action. Keep the list concise and include no commentary. *. . . 47 additional variants*
`Proc.+`	**1.** Using `{task}` as the overall activity, name the upcoming `{num_steps}` actions. Do not include any extra text beyond the answer. **2.** For `{task}`, identify the next action based on the current video. Return only the action or numbered action list. **3.** In the context of `{task}`, what step comes after the shown segment? Use numbered lines when multiple actions are requested. *. . . 47 additional variants*

Table 12: Representative generated prompt variants used to evaluate prompt-conditioned expert selection. Three of 50 variants are shown per task; the held-out split achieves the 100% test F1 reported in the text.

A.7 Offline Expert-Swap Routing

Table 13 provides an offline diagnostic for whether COIN task behavior is localized in the routed experts. We keep the evaluation prompts and route ids unchanged, but swap the trained FFN weights of two task experts across all decoder layers before evaluation. If the experts behave as task-specialized modules, the degradation should concentrate on the swapped tasks while the untouched routes remain close to the base model.

Method	COIN Benchmark Top-1 Accuracy↑				
	Step	Next	Proc.	Task	Proc.+
VideoLLM-MoTE-1B+5E	65.1	50.3	50.5	94.4	54.2
Swap Step↔Next	21.5	17.4	50.5	94.4	54.2
Swap Next↔Proc.	65.1	23.3	30.7	94.4	54.2
Swap Proc.↔Task	65.1	50.3	4.97	15.36	54.2
Swap Task↔Proc.+	65.1	50.3	50.5	15.98	5.36

Table 13: Offline expert-swap routing diagnostic on COIN. Highlighted cells indicate the two task columns whose expert weights are swapped at evaluation time. Untouched task routes keep the base route weights.

The diagnostic shows that accuracy collapses primarily on the swapped task pair. Swapping Step and Next damages only those two columns, while Proc., Task, and Proc.+ remain at the base values. The same locality appears for the other pairs, with the strongest failures occurring when Proc. is swapped with Task or when Task is swapped with Proc.+. This pattern supports the interpretation that routed FFN experts capture task-specific transformations rather than acting as interchangeable capacity. The diagnostic is not an expert-removal or unlearning evaluation; it is a stress test of expert specialization under deliberately incorrect route-to-weight assignments.

A.8 Document-Domain Evaluation

A.8.1 Architecture

Figure 4 expands the auxiliary document experiment from Section 4.5. PP-DocLayoutV3 detects and crops document regions, CogVIT encodes the resulting visual inputs, and the GLM-OCR attention path forms a shared representation. The MoTE conversion is confined to the MLP/FFN path: the shared expert is always active, while the switch selects one task expert for the sample. Their outputs are combined before structured decoding; dotted routes denote experts that are inactive in the illustrated example.

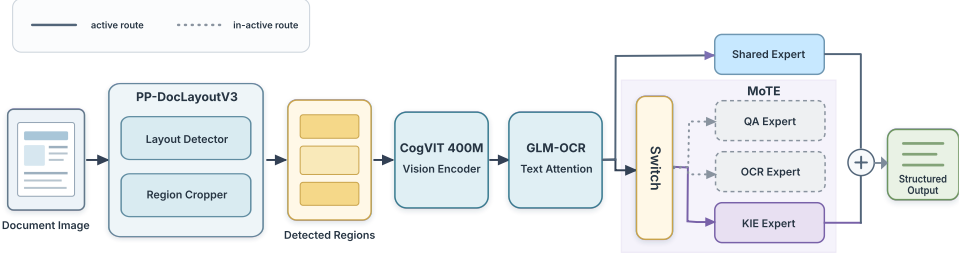

Figure 4: **GLM-OCR-MoTE document processing architecture.** Detected document regions pass through the shared CogVIT encoder and GLM-OCR attention path before the switch activates one task-specific MLP expert alongside the shared expert. Solid lines denote active routes and dotted lines denote inactive routes.

A.8.2 Detailed Evaluation

This section expands the compact document-domain tables from Section 4.5. The GLM-OCR-MoTE-0.9B+2E conversion uses the same high-level pattern as the video model: an always-active shared FFN path is paired with one selected task route. In this auxiliary setting, the two routed experts correspond to OCR and receipt KIE. During KIE adaptation, the new receipt route learns to emit structured receipt fields directly from image inputs, while the retained OCR route is evaluated separately on the original OCR benchmark.

The comparison groups baselines by input modality because this determines how much document structure is provided to the model. OCR-text-only and OCR-plus-layout baselines receive recognized text, bounding boxes, or both. Image-using baselines receive pixels, sometimes together with OCR or layout. GLM-OCR-MoTE-0.9B+2E is listed as an image-input model because the KIE route predicts fields from the receipt image without an external OCR-text prompt. Tables 14 and 15 report the full per-field values behind the compact main-paper comparisons.

Method	Input Modality	SROIE Task 3 Strict F1 (%) ↑ (KIE Domain)					OmniDocBench v1.5 (Original Task)					
		Comp.	Date	Addr.	Total	Micro	Over.↑	TxtEd.↓	Form.↑	Tab.↑	Tab-S↑	RdEd.↓
Text-Only Baselines (OCR Prompted)												
GPT-4 + OCR [58]	OCR Text Only	–	–	–	–	90.60	–	–	–	–	–	–
Llama2-7B-chat [61]	OCR Text Only	–	–	–	–	68.97	–	–	–	–	–	–
Vicuna-1.5-7B [8]	OCR Text Only	–	–	–	–	68.20	–	–	–	–	–	–
Llama2 + OCR [61]	OCR Text Only	–	–	–	–	56.40	–	–	–	–	–	–
Llama2-7B [61]	OCR Text Only	–	–	–	–	34.96	–	–	–	–	–	–
Spatial / Layout-Aware Baselines (No Image)												
DocLLM-7B [56]	OCR Text + BBoxes	–	–	–	–	91.90	–	–	–	–	–	–
Multimodal Baselines (Image + Text)												
LayoutLLM-7B [40]	Image+Text+Layout	–	–	–	–	72.12	–	–	–	–	–	–
DocKylin [29]	Image + Text	–	–	–	–	69.60	–	–	–	–	–	–
Qwen-VL-7B [1]	Image + Text	–	–	–	–	58.59	–	–	–	–	–	–
LLaVA-1.5-7B [30]	Image + Text	–	–	–	–	3.83	–	–	–	–	–	–
LLaVAR-7B [63]	Image + Text	–	–	–	–	2.38	–	–	–	–	–	–
Base OCR Architecture												
GLM-OCR-0.9B [30]	Image	49.76	78.22	17.81	74.40	55.04	94.62	0.040	93.90	93.96	96.39	0.044
Mixture of Task Experts Adaptation (Ours)												
GLM-OCR-MoTE-0.9B+2E (ours)	**Image**	**90.49**	**97.41**	**70.03**	**93.66**	**87.90**	**94.62**	**0.040**	**93.90**	**93.96**	**96.39**	**0.044**

Table 14: Full SROIE and OmniDocBench comparison corresponding to Table 6. The first metric block evaluates receipt KIE, while the second checks preservation of the original OCR route.

Method	Input Modality	CORD Relaxed F1 Score (%) ↑								
		Cash	Chng	Cred	Menu	Serv	SubTot	Tax	Total	Micro
Text-Only Baselines (OCR Prompted)										
RoBERTa [24]	OCR Text Only	–	–	–	–	–	–	–	–	91.98
BERT_large/T5 [11, 12]	OCR Text Only	–	–	–	–	–	–	–	–	90.25
Spatial / Layout-Aware Baselines (No Image)										
LayoutLM [53]	OCR Text + BBoxes	–	–	–	–	–	–	–	–	94.93
LAMBERT [3]	OCR Text + BBoxes	–	–	–	–	–	–	–	–	94.41
Multimodal Baselines (Image + Text + Layout)										
UDOP [51]	Image+Text+Layout	–	–	–	–	–	–	–	–	97.60
LayoutLMv2 [52]	Image+Text+Layout	–	–	–	–	–	–	–	–	96.01
DocTr [25]	Image+Text+Layout	–	–	–	–	–	–	–	–	94.40
Donut [20]	Image + Text	–	–	–	–	–	–	–	–	91.60
Base Architecture (Before MoTE)										
GLM-OCR-0.9B [30]	Image	11.43	20.59	16.67	0.00	47.06	42.55	15.69	28.12	20.07
Mixture of Task Experts Adaptation (Ours)										
GLM-OCR-MoTE-0.9B+2E (ours)	**Image**	**95.45**	**99.10**	**93.33**	**90.00**	**100.00**	**100.00**	**100.00**	**95.24**	**95.79**

Table 15: Full CORD KIE comparison corresponding to Table 7. GLM-OCR-MoTE-0.9B+2E uses the receipt KIE route and predicts structured fields directly from the receipt image.

A.9 Limitations

MoTE is most effective when the task set has meaningful and stable boundaries. The prompt-conditioned selector is evaluated on generated variants of five predefined COIN intents, but this controlled result does not cover ambiguous, overlapping, or compositional requests; softer routing, hierarchical experts, or expert composition may be needed in such cases. Incremental expert addition protects existing routes by freezing old modules, but this reduces plasticity for new tasks; stronger expansion may require controlled updates to shared representations or rehearsal.