

# MotionEnhancer: Leveraging Video Diffusion for Motion-Enhanced Vision-Language Models

## Supplementary Material

### 1. Preliminaries

#### 1.1. DDIM Framework

Denosing Diffusion Implicit Models (DDIM) [21] provide an efficient and flexible sampling framework for diffusion models. Traditional diffusion models, such as Denosing Diffusion Probabilistic Models (DDPM) [9], rely on a stochastic generation process, where random noise is gradually removed through a series of probabilistic denoising steps. This approach, while effective, often requires a large number of iterations to generate high-quality samples, resulting in high computational cost. DDIM, in contrast, introduces a deterministic alternative to the sampling process. It leverages a non-Markovian forward process that adds Gaussian noise to the original data  $\mathbf{x}_0$  according to a predefined noise schedule  $\alpha_t$ . Specifically, the forward process can be formulated as:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad (1)$$

where  $\alpha_t$  decreases monotonically over time, gradually corrupting the data.

The key innovation of DDIM lies in its reverse process. Instead of sampling new noise at every step, DDIM uses a deterministic mapping to reconstruct the data. The denoising step is given by:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\hat{\mathbf{x}}_0(\mathbf{x}_t) + \sqrt{1 - \alpha_{t-1}}\epsilon_\theta(\mathbf{x}_t, t), \quad (2)$$

where  $\hat{\mathbf{x}}_0(\mathbf{x}_t)$  is the predicted clean data:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) = \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}}. \quad (3)$$

By removing the stochasticity from the reverse process, DDIM enables faster and more controllable sampling. This deterministic formulation allows the model to generate high-quality samples with significantly fewer steps compared to DDPM, while preserving the overall data structure and ensuring boundary consistency. As a result, DDIM has emerged as a widely favored option in diffusion-based generative modeling, especially within applications where both efficiency and fidelity are critical requirements.

#### 1.2. DDIM Inversion

DDIM inversion [18] extends the DDIM framework by enabling the mapping of real data  $\mathbf{x}_0$  back into its corresponding latent noise  $\mathbf{x}_T$ . This process is achieved through an iterative application of the reversed DDIM denoising step,

effectively tracing the generative trajectory in the opposite direction.

Formally, starting from the observed data  $\mathbf{x}_0$ , the inversion procedure sequentially estimates the intermediate noisy representations  $\mathbf{x}_t$  for  $t = 0, \dots, T - 1$  via:

$$\mathbf{x}_{t+1} = \sqrt{\alpha_{t+1}}\hat{\mathbf{x}}_0(\mathbf{x}_t) + \sqrt{1 - \alpha_{t+1}}\epsilon_\theta(\mathbf{x}_t, t). \quad (4)$$

After  $T$  iterations, this process yields a latent code  $\mathbf{x}_T = \mathcal{G}^{-1}(\mathbf{x}_0)$ , which encodes the original data within the noise space of the diffusion model.

A notable feature of DDIM inversion is its path determinism: if the noise prediction function  $\epsilon_\theta$  remains fixed, the entire sequence is reversible. In other words, if one starts from  $\mathbf{x}_0$ , performs the inversion step, and then follows with the standard forward generative process, the original input will be reconstructed exactly:

$$\mathbf{x}_0 \xrightarrow{\text{invert}} \mathbf{x}_T \xrightarrow{\text{reconstruct}} \mathbf{x}_0. \quad (5)$$

This deterministic and bijective mapping ensures that each data point has a unique latent representation and that the model’s internal states can be precisely analyzed.

Such inversion capability is especially valuable for investigating the internal mechanisms of diffusion models, enabling controllable editing, semantic manipulation, and deeper interpretability. It provides a systematic approach to relate observable samples with their latent origins, thus opening avenues for fine-grained analysis and intervention in generative modeling tasks.

#### 1.3. Introduction of Our Used Models

We utilize CogVideoX-5B as our Video Diffusion Model (VDM) and Qwen2.5-VL (3B and 7B) and InternVL3 (2B and 8B) as our Vision-Language Models (VLM). Below is a concise overview of these models, covering their architectural designs and operational workflows.

##### **CogVideoX: A DiT-based Video Generative Model.**

CogVideoX [26] is a diffusion-transformer model [19] for text-to-video generation that tackles the twin challenges of high-resolution fidelity and long-range temporal coherence in a single, end-to-end pipeline. Its first stage is a 3D causal VAE that compresses both spatial and temporal axes with an  $8 \times 8 \times 4$  reduction factor, turning seconds of raw pixels into a compact latent cube while preserving fine detail and motion continuity; the causal design guarantees that decoding can proceed frame-by-frame for streaming applications. The latent volume is patched into spatio-temporal tokens and sim-

ply concatenated with T5 text embeddings [20], eliminating cumbersome cross-attention modules and allowing the same transformer blocks to process both modalities. Inside each transformer layer, 3D full attention lets every patch attend to any other patch in its causal window, capturing large motions without drift. Separate expert adaptive LayerNorm branches scale and shift visual and textual features before they merge, aligning modalities without extra parameters.

**Qwen2.5-VL: Flagship VLM in Qwen Series.** Qwen2.5-VL [2] is a VLM that ingests images and text at their native resolution without ever cropping them into fixed grids. At its core is a redesigned Vision Transformer (ViT) [4] trained from scratch, which incorporates window attention in most layers—with only four layers using full attention—to reduce computational complexity from quadratic to linear relative to input patches, enabling native handling of dynamic resolutions without normalization artifacts. The resulting dense visual tokens are streamed into a lightweight vision-language merger that groups neighboring patches and projects the ensemble through a two-layer MLP, yielding a compact set of vision embeddings that align seamlessly with the text token space while retaining fine-grained spatial cues. For video, Qwen2.5-VL introduces Multimodal Rotary Position Embedding (MRoPE) tied to absolute timestamps: temporal position IDs are explicitly synchronized with real-world clock time, so the model perceives frame-rate variations naturally and reasons about event order without external preprocessing.

**InternVL3: Native Multimodal Pre-training at Scale.** InternVL3 [30] departs from the dominant “language-first, vision-second” paradigm by performing native multimodal pre-training from the outset: a single unified stage interleaves large-scale text corpora with diverse vision–language data, allowing linguistic and visual capacities to co-evolve without later alignment. Architecturally, the model keeps the clean ViT–MLP–LLM stack: an image-agnostic ViT feeds a lightweight two-layer projector that directly maps raw visual tokens into the language model’s embedding space, eliminating resolution-specific modules or fixed-size crops. To gracefully accommodate long visual contexts, InternVL3 adopts Variable Visual Position Encoding (V2PE) [8] that modulates positional increments for visual tokens, keeping the overall sequence length within the LLM’s native window. Operationally, all components—vision encoder, projector, and language backbone—are jointly optimized on the combined corpus; subsequent post-training refines conversational quality through supervised fine-tuning followed by Mixed Preference Optimization (MPO) [24], a unified objective that blends preference, quality, and generation losses. Test-time computation is amplified by a visual process-supervised critic that selects the best among multiple decoded chains, pushing reasoning quality without enlarging the model.

---

### Algorithm 1: MotionEnhancer Pipeline

---

**Input:** Video  $V$ ; Question  $I$ ; Target response  $s$ ;  
Pretrained VLM; Pretrained VDM

**Output:** Fine-tuned VLM; Total loss  $\mathcal{L}_{total}$

- 1 **Step 1: VDM-based Attention Extraction**
- 2  $z_{\text{vision}} \leftarrow \text{VDM.VAE.encode}(V)$ ; // Encode video,  
 $z_{\text{vision}} \in \mathbb{R}^{F \times H \times W}$
- 3  $z_{\text{noise}} \leftarrow \text{DDIM\_Inversion}(z_{\text{vision}}, \text{step} = 5)$ ;  
// Obtain DDIM-inverted noise
- 4  $z_{\text{text}} \leftarrow \text{VDM.TextEncoder}(I)$ ; // Encode text
- 5  $z_{\text{mm}} \leftarrow [z_{\text{text}}, z_{\text{noise}}]$ ; // Form multimodal sequence
- 6 **foreach** *DDIM denoising step do*
- 7     Compute attention:  $A_{\text{mm}} \leftarrow \text{Softmax}\left(\frac{Q_{\text{mm}} K_{\text{mm}}^T}{\sqrt{d}}\right)$ ;
- 8     Average  $A_{\text{mm}}$  for all layers;
- 9     Store  $A_{\text{mm}}$  for all heads;
- 10 **Step 2: Motion-sensitive Head Selection**
- 11 **foreach** *head  $h$  in VDM do*
- 12     Extract vision-to-vision attention  $A_{\text{v2v}}^{(h)}$  from  
    $A_{\text{mm}}$ ; //  $[S \times S], S = H \times W$
- 13     Compute DFC, TCS, and DSR
- 14  $\text{score}_h = \text{Norm}(\text{DFC}) + \text{Norm}(\text{TCS}) + \text{Norm}(\text{DSR})$ ;  
   ; // Compute head score
- 15  $A_{\text{mm}}^{\mathcal{H}_m} \leftarrow \text{Avg}(\{A_{\text{mm}}^{(h)}\}_{h \in \mathcal{H}_m})$ ; // Select top-50%  
   heads  $\mathcal{H}_m$ , and aggregate motion heads
- 16 **Step 3: Motion-salient Text Token Identification**
- 17  $A_{\text{t2v}} \leftarrow \text{extract text-to-vision region}(A_{\text{mm}}^{\mathcal{H}_m})$ ;  
   // Extract text-to-vision attention
- 18 **foreach** *text token  $t$  do*
- 19      $A_{t,\text{t2f}} \leftarrow \text{Avg}_{(h,w)} A_{\text{t2v}}[t, f, h, w]$ ; // Pool over  
   spatial dimensions
- 20      $m_t \leftarrow \text{Mean}_f(A_{t,\text{t2f}})$ ; // Compute mean
- 21      $d_t \leftarrow \frac{1}{F-1} \sum_{f=1}^{F-1} |A_{t,\text{t2f}}[f+1] - A_{t,\text{t2f}}[f]|$ ;  
   // Compute first-order diff mean
- 22      $MS(t) = m_t + d_t$ ; // Motion score
- 23 Select top-50% tokens  $\mathcal{T}_m$  by  $MS(t)$
- 24  $A_{\text{VDM}} \leftarrow A_{\text{t2v}}[\mathcal{T}_m, :]$ ; // Obtain VDM motion prior
- 25 **Step 4: VLM Attention Extraction**
- 26 Extract VLM attention maps  $A_{\text{VLM,raw}}$  for  $(V, I, s)$
- 27 Average pool across heads/layers and obtain  $A_{\text{VLM}}$
- 28  $A_{\text{VLM}} \leftarrow A_{\text{VLM}}[\mathcal{T}_m, :]$ ; // Select same tokens
- 29 Interpolate  $A_{\text{VLM}}$  to match  $A_{\text{VDM}}$
- 30 **Step 5: Supervised Fine-tuning with Attention Alignment**
- 31  $A_{\text{VLM,aligned}} \leftarrow \text{MLP}(A_{\text{VLM}})$  // Align VLM attention  
    $\mathcal{L}_{\text{MSE}} = \|A_{\text{VLM,aligned}} - A_{\text{VDM}}\|_2$ ; // Compute  
   attention alignment loss
- 32  $\mathcal{L}_{\text{AR}} = -\sum_{i=1}^{|s|} \log P(s[i]|V, I, \theta, s[1:i-1])$ ;  
   // Compute auto-regressive loss
- 33  $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{AR}} + \lambda \mathcal{L}_{\text{MSE}}$ ; // Compute total loss
- 34 Update trainable parameters in VLM with  $\mathcal{L}_{\text{total}}$

---

## 2. Algorithm

The MotionEnhancer pipeline is shown in Algorithm 1. **(1) Step 1: VDM-based Attention Extraction.** The input video is encoded into latent representations using a pre-trained VDM. During the DDIM inversion and denoising process, multi-head attention maps are extracted from the VDM across all layers. **(2) Step 2: Motion-sensitive Head Selection.** For each attention head, motion relevance is quantitatively evaluated using the Diagonal Focus Coefficient, Temporal Continuity Score, and Diagonal Saliency Ratio. The top-50% heads most sensitive to temporal motion are selected, and their attention maps are aggregated. **(3) Step 3: Motion-salient Text Token Identification.** The aggregated attention maps are analyzed to compute temporal attention statistics for each text token. Tokens exhibiting high temporal attention and significant inter-frame variation are selected as motion-salient text tokens. The VDM attention maps filtered by both MHS and MTTI are used to construct motion priors, focusing on motion-relevant regions and tokens. **(4) Step 4: VLM Attention Extraction.** For the same video-text input, attention maps are extracted from the VLM and pooled across heads and layers. The attention corresponding to the selected motion-salient tokens is retained and interpolated if necessary. **(5) Step 5: Supervised Fine-tuning with Attention Alignment** During fine-tuning, the VLM is optimized by minimizing the L2 distance between its own attention maps and the motion priors from the VDM, while also being supervised with the standard auto-regressive loss for video question answering. All trainable parameters in the VLM are updated with respect to the combined loss, enabling the model to acquire enhanced motion understanding without modifying its architecture or adding new trainable modules.

## 3. Detailed Theoretical Analysis

This section provides a comprehensive theoretical extension from the main text, offering deeper insights into the distributional properties of VLMs and VDMs that form the foundation of MotionEnhancer.

### 3.1. Preliminaries and Notation

Let a video be denoted as

$$V = (V_1, V_2, \dots, V_F), \quad (6)$$

where  $V_f$  is the  $f$ -th frame and  $F$  is the number of frames. Each frame is discretized into spatial locations  $s \in \mathcal{S}$ ; we denote the visual feature at location  $s$  and frame  $f$  as  $v_{s,f}$ .

A VLM is trained with an autoregressive objective:

$$\mathcal{L}_{\text{AR}} = - \sum_i \log p_{\theta}(r_i | V, r_{<i}), \quad (7)$$

where  $\theta$  denotes the VLM parameters and  $r_i$  is the  $i$ -th response token.

Internally, the VLM produces text-to-video attention scores. For a token  $t$ , the attention weight at location  $(s, f)$  can be interpreted as a discriminative conditional distribution:

$$A^{\text{VLM}}(t, s, f) \approx p_{\theta}(t | v_{s,f}, r_{<i}), \quad (8)$$

which reflects a recognition-style distribution  $p_{\theta}(t|V)$  learned under the autoregressive training paradigm.

By contrast, motion understanding requires discovering *evidence* in the video that supports a motion-related token  $t$  and its associated answer  $a$ . As argued in Sec. 3, this can be expressed via latent evidence  $E$  and its factorization over semantic units  $t \in T(q, a)$  as:

$$p^*(a | V, q) = \sum_E p^*(a | E, q) p^*(E | V, q), \quad (9)$$

$$E = \{E_t | t \in T(q, a)\}, \quad (10)$$

$$p^*(E | V, q) = \prod_{t \in T(q, a)} p^*(E_t | V, t, q), \quad (11)$$

where  $E_t$  identifies the spatiotemporal regions that visually realize token  $t$ .

Within each  $E_t$ , the relevance of a location  $(s, f)$  can be represented by a *concept-conditioned visual likelihood*:

$$r_t(s, f) = p^*(v_{s,f} | t), \quad (12)$$

which naturally leads to an *evidence-seeking* distribution  $p^*(V | t)$ : “given concept  $t$ , where and how is it visually realized over space and time?”

For motion-related tokens (e.g., verbs, motion modifiers), the latent evidence must encode *frame-to-frame transitions*, which, at the feature level, can be summarized by a motion magnitude:

$$\text{Motion}(s, f) \propto \|V_{f+1}(s) - V_f(s)\|. \quad (13)$$

### 3.2. Distributional Mismatch in More Detail

We now formalize the mismatch between  $p_{\theta}(t | V)$  (what the VLM is trained on) and  $p^*(V | t)$  (what motion understanding requires). Using Bayes’ rule at the level of per-location features:

$$p^*(t | v_{s,f}) = \frac{p^*(v_{s,f} | t) p^*(t)}{p^*(v_{s,f})}. \quad (14)$$

If the VLM were trained with sufficiently rich supervision and under a well-specified model,  $A^{\text{VLM}}(t, s, f)$  could in principle approximate  $p^*(t | v_{s,f})$ . However, in realistic settings, the supervision usually consists of high-level QA or captions rather than dense, per-location labels. And the set of questions and responses  $(q, a)$  is highly imbalanced:

many queries can be answered from a static context without modeling temporal changes. Also, the model capacity and optimization are shaped to maximize accuracy, not to invert the distribution to recover  $p^*(v_{s,f} | t)$ .

Hence, a VLM can achieve high training and evaluation performance by relying mostly on static appearance cues. Formally, decompose the visual feature as:

$$v_{s,f} = v_{s,f}^{\text{stat}} + v_{s,f}^{\text{dyn}}, \quad (15)$$

where  $v^{\text{stat}}$  captures static background and identity cues, and  $v^{\text{dyn}}$  captures temporal variations. If most training questions can be answered using  $v^{\text{stat}}$  alone, then gradients of  $\mathcal{L}_{\text{AR}}$  with respect to  $v^{\text{dyn}}$  are small or sparse, and the model converges to a solution where:

$$p_{\theta}(t | v_{s,f}) \approx p_{\theta}(t | v_{s,f}^{\text{stat}}), \quad (16)$$

i.e., the conditional depends only weakly on temporal dynamics.

On the other hand, motion understanding requires estimating where and when motion-related evidence occurs, which is precisely governed by  $p^*(v_{s,f} | t)$  and its temporal variation across  $f$ . In particular, the expectation

$$\mathbb{E}[\text{Motion}(s, f) | t] \propto \mathbb{E}[\|V_{f+1}(s) - V_f(s)\| | t] \quad (17)$$

is a functional of the evidence-seeking distribution  $p^*(V | t)$ , not of the recognition distribution  $p^*(t | V)$ .

Thus, unless the training signal explicitly forces the VLM to approximate or invert  $p^*(V | t)$ , there is a **structural mismatch**: the model is optimized for predicting tokens given videos, while fine-grained motion reasoning requires discovering the video given a concept. This explains why VLMs exhibit appearance bias and weak temporal sensitivity, as observed empirically.

### 3.3. VDM Attention as Evidence-Seeking Posterior

Video diffusion models (VDMs) are trained to generate videos conditioned on text. At the denoising step  $k$ , the model receives noisy latent video  $z_k$  and text tokens  $t$ , and outputs a cleaner latent that approximates the true video latent  $z_0$ . The cross-attention at this step is computed as:

$$A^{\text{VDM}}(t, s, f) = \text{Softmax}\left(\frac{Q(t)K(s, f)^{\top}}{\sqrt{d}}\right), \quad (18)$$

where  $Q(t)$  is the query for token  $t$  and  $K(s, f)$  is the key derived from the latent at location  $(s, f)$ . Aggregating across layers and timesteps, as described in the main paper, yields a stable attention estimate.

Under mild assumptions (e.g., linear attention maps and sufficiently expressive  $Q/K$  projections), this attention can

be interpreted as a *normalized relevance score* between  $t$  and  $v_{s,f}$  conditioned on the current noisy latent:

$$A^{\text{VDM}}(t, s, f) \approx p_{\phi}(v_{s,f} | t, z_k), \quad (19)$$

where  $\phi$  denotes VDM parameters. This resembles the concept-conditioned likelihood  $p^*(v_{s,f} | t)$  required for motion reasoning.

Two properties make VDM attention particularly suited as a motion prior: **Evidence-seeking nature**. Since the model must generate video content that visually realizes the text token  $t$ , the attention highlights locations where the semantics of  $t$  should appear. This approximates the evidence-seeking distribution  $p^*(V | t)$ : regions receiving higher attention are those where the generated video is more likely to exhibit the concept. **Motion calibration**. The denoising objective penalizes reconstruction errors, which are larger in regions with strong motion (large  $\|V_{f+1}(s) - V_f(s)\|$ ) because these regions are harder to predict from noisy latents. Consequently, gradients concentrate on dynamic areas, making cross-attention for motion-related tokens sensitive to temporal variations. Locations with higher motion magnitude receive higher or more fluctuating attention across timesteps, aligning with the notion of motion evidence.

In the idealized limit where the VDM perfectly models the true data distribution,  $A^{\text{VDM}}(t, s, f)$  converges to a calibrated approximation of  $p^*(v_{s,f} | t)$ . Even in practice, it serves as a high-quality, motion-aware prior that complements the VLM’s appearance-focused attention.

### 3.4. Attention Alignment as Approximate Posterior Matching

MotionEnhancer introduces an auxiliary MSE loss between VDM and VLM attention:

$$\mathcal{L}_{\text{MSE}} = \|\text{Aligner}(A^{\text{VLM}}) - A^{\text{VDM}}\|_2^2, \quad (20)$$

and optimizes

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{AR}} + \lambda \mathcal{L}_{\text{MSE}}. \quad (21)$$

For a given motion-salient token  $t$ , we treat each attention row as a discrete distribution over locations:

$$\pi_{\theta}^{\text{VLM}}(s, f | t) := A^{\text{VLM}}(t, s, f), \quad (22)$$

$$\pi_{\phi}^{\text{VDM}}(s, f | t) := A^{\text{VDM}}(t, s, f). \quad (23)$$

Ignoring the Aligner for a moment (or assuming it is near identity), we may view  $\mathcal{L}_{\text{MSE}}$  as minimizing an  $L^2$  distance between these distributions. In expectation over data, we approximately solve:

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}_{(V, q, a)} [-\log p_{\theta}(a | V, q)] + \\ & \lambda \mathbb{E}_{t \in T_{\text{mot}}(q, a)} \left[ \|\pi_{\theta}^{\text{VLM}}(\cdot | t) - \pi_{\phi}^{\text{VDM}}(\cdot | t)\|_2^2 \right]. \end{aligned} \quad (24)$$

This expanded objective has two important interpretations: **Posterior matching in attention space.** If we interpret  $\pi_\theta^{\text{VLM}}(\cdot | t)$  as an approximate posterior over evidence locations for token  $t$ , and  $\pi_\phi^{\text{VDM}}(\cdot | t)$  as an approximate concept-conditioned distribution from the generative model, then  $\mathcal{L}_{\text{MSE}}$  acts to push the VLM distribution toward the motion-sensitive evidence-seeking distribution encoded by the VDM. This reduces the discrepancy between the recognition distribution (where a VLM can rely heavily on static cues) and the generative distribution that better respects motion. **Constrained optimization and regularization.** The optimization in Eq. 24 can equivalently be viewed as:

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}[-\log p_\theta(a | V, q)] \\ \text{s.t.} \quad & \mathbb{E}[\|\pi_\theta^{\text{VLM}} - \pi_\phi^{\text{VDM}}\|_2^2] \leq \epsilon(\lambda), \end{aligned} \quad (25)$$

where  $\epsilon(\lambda)$  decreases as  $\lambda$  increases. Thus, the feasible set of VLM parameters is restricted to those producing attention maps that are compatible with motion priors distilled from the VDM.

In practice, the Aligner is implemented as a small MLP. Its limited capacity enables mild, smooth transformations (e.g., rescaling or slight warping) to account for architectural differences between VLM and VDM, while preventing degenerate solutions in which the VLM attention remains unchanged and the Aligner absorbs the full discrepancy. In this sense, the Aligner functions as a preconditioner rather than a substitute for matching attention distributions.

### 3.5. MHS and MTTI as Projection onto a Motion Subspace

Both MHS and MTTI modules act as selection operators that isolate the motion-relevant components of the VDM attention. Let  $\mathcal{A}$  denote the full text-to-video attention tensor from the VDM.

**Head-level projection.** MHS selects a subset of heads exhibiting strong temporal structure. At a high level, this can be written as a projection:

$$A_{\text{head}} = \Pi_{\text{head}}(\mathcal{A}), \quad (26)$$

where  $\Pi_{\text{head}}$  keeps only motion-sensitive heads and discards others.

**Token-level projection.** Similarly, MTTI selects text tokens whose attention varies across frames. This yields a second projection:

$$A_{\text{mot}} = \Pi_{\text{token}}(A_{\text{head}}). \quad (27)$$

**Overall effect.** Together, MHS and MTTI produce a compact motion-focused attention map:

$$A_{\text{mot}} = \Pi_{\text{token}}(\Pi_{\text{head}}(\mathcal{A})), \quad (28)$$

Table 1. The experimental system and hardware setups.

System & Hardware Overview	
CPU	Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz
GPU	8×NVIDIA A100 Tensor Core GPU
Memory	1T DRAM
Operating System	Ubuntu 22.04.4 LTS
CUDA Version	12.1
NVIDIA Driver	530.30.02
ML Framework	Python 3.10.12 Pytorch 2.5.1
GPU Specifications	
CUDA Cores	6912
Memory Capacity	80GB
Memory Bandwidth	1935GB/s

Table 2. Statistical information of the two used benchmarks, including each task type and quantity.

MotionBench	MR	LM	AO	RC	MO	CM
	1478	546	519	400	690	385
FAVOR-Bench	AS	HAC	SAD	MAD	CM	NSM
	2637	1541	1662	1205	1075	64

which represents the projection of the original VDM attention onto a motion subspace. This projected attention is the signal aligned with the VLM in MotionEnhancer.

## 4. Discussion

### 4.1. Dependence on VDM quality

MotionEnhancer’s effectiveness depends on the informativeness of the VDM’s motion priors. As discussed in the main content, when a VDM produces diffuse attention (e.g., large static subjects), the alignment signal weakens. Nevertheless, using such priors can yield improvements, and our paper’s focus is on how to transfer those priors rather than on improving their intrinsic quality. To mitigate this, we will employ solutions like confidence gating (low-motion-confidence cases get minimal alignment) or ensemble attention maps from multiple VDMs to reduce model-specific artifacts.

### 4.2. Role of $\lambda$ in Alignment

Tuning  $\lambda$  effectively trades off trust in the VDM prior versus the VLM’s own training signal.  $\lambda = 0$  means no prior injection and standard VLM. Moderate  $\lambda$  presents motion-sensitive heads and tokens are guided by priors, while static reasoning remains mostly unaffected. Very large  $\lambda$  denotes that VLM may overfit the VDM’s specific biases, potentially harming robustness.

Table 3. Quantitative results of MotionBench. \* denotes results we reproduced using their open-source code, while other results are taken from the original benchmark.

Model	Frames	Overall	Average	MR	LM	CM	MO	AO	RC
Small Size Series									
Qwen2.5-VL-3B* [2]	1fps	53.56	49.45	59.54	53.11	38.44	70.14	40.46	35.00
<b>Qwen2.5-VL-3B + MotionEnhancer (Ours)</b>	1fps	56.60 <sup>↑3.04</sup>	52.51 <sup>↑3.06</sup>	63.06	61.72	47.01	68.84	43.16	31.25
InternVL3-2B* [30]	8	53.96	49.69	60.01	57.69	43.90	70.00	40.27	26.25
<b>InternVL3-2B + MotionEnhancer (Ours)</b>	8	55.50 <sup>↑1.54</sup>	51.35 <sup>↑1.66</sup>	61.57	57.51	46.23	71.30	42.00	29.50
Medium Size Series									
MiniCPM-V2.6-7B [27]	64	52	-	-	-	-	-	-	-
CogVLM2-Video-8B [10]	24	41	-	-	-	-	-	-	-
GLM4-9B + TE Fusion [11]	16	58	-	-	-	-	-	-	-
Qwen2.5-VL-7B* [2]	1fps	52.81	48.29	59.00	54.58	35.58	71.30	38.54	30.75
Qwen2.5-VL-7B + MotionSight* [5]	1fps	55.30	51.56	59.88	57.33	47.01	73.91	40.46	30.75
<b>Qwen2.5-VL-7B + MotionEnhancer (Ours)</b>	1fps	57.04 <sup>↑4.23</sup>	52.92 <sup>↑4.63</sup>	63.40	61.54	47.27	70.29	43.55	31.50
InternVL3-8B* [30]	8	54.88	50.81	60.42	58.06	43.64	70.29	43.93	28.50
<b>InternVL3-8B + MotionEnhancer (Ours) [30]</b>	8	57.69 <sup>↑2.81</sup>	53.22 <sup>↑2.41</sup>	64.14	60.07	48.83	75.94	40.85	29.50
Large Size Series									
PLLaVA-34B [25]	16	52	-	-	-	-	-	-	-
LLaVA-NeXT-Video-34B [13]	32	48	-	-	-	-	-	-	-
Qwen2.5-VL-72B* [2]	1fps	58.30	54.32	64.00	60.30	48.60	73.20	46.80	33.00

Table 4. Quantitative results of FAVOR-Bench. \* denotes results we reproduced using their open-source code, while other results are taken from the original benchmark.

Model	Frames	Overall	Average	AS	HAC	SAD	MAD	CM	NSM
Small Size Series									
VideoLLaMA3-2B [28]	1fps	32.98	34.61	28.97	36.60	34.90	38.01	28.56	40.62
InternVL2.5-2B [3]	8	22.90	23.45	18.70	28.23	23.71	27.47	19.16	23.44
Qwen2.5VL-3B* [2]	1fps	37.43	38.07	38.45	38.16	39.35	43.40	23.72	45.31
<b>Qwen2.5VL-3B + MotionEnhancer (Ours)</b>	1fps	44.53 <sup>↑7.10</sup>	43.94 <sup>↑5.87</sup>	45.01	51.59	44.40	48.96	28.37	45.31
InternVL3-2B* [30]	1fps	39.27	39.11	37.66	43.28	40.49	44.98	29.21	39.06
<b>InternVL3-2B + MotionEnhancer (Ours)</b>	1fps	43.71 <sup>↑4.44</sup>	45.35 <sup>↑6.24</sup>	38.53	54.57	42.60	51.78	33.02	51.56
Medium Size Series									
LLaVA-Video-7B-Qwen2 [29]	64	38.60	39.94	36.14	41.27	41.28	44.48	29.58	46.88
VideoChat-Flash-Qwen2-7B [14]	1fps	43.82	44.86	41.90	48.41	42.84	50.95	35.07	50.00
VideoLLaMA3-7B [28]	1fps	41.46	41.46	40.20	44.13	42.42	48.30	31.53	42.19
Video-LLaVA-7B [15]	8	25.37	25.09	24.91	21.54	25.45	30.54	26.23	21.88
LLaVA-NeXT-Video-7B [13]	8	23.45	22.27	21.27	22.45	26.05	26.72	23.07	14.06
Tarsier-7B [23]	8	17.46	20.50	12.55	21.16	17.87	17.93	22.23	31.25
Qwen2.5VL-7B* [2]	1fps	42.61	42.58	41.64	47.83	44.89	47.55	28.28	45.31
Qwen2.5VL-7B + MotionSight* [5]	1fps	45.47	45.99	46.23	51.59	45.01	50.04	29.95	53.12
<b>Qwen2.5VL-7B + MotionEnhancer (Ours)</b>	1fps	46.88 <sup>↑4.27</sup>	47.01 <sup>↑4.43</sup>	49.34	50.62	45.37	53.20	30.42	53.12
InternVL3-8B* [30]	1fps	45.82	46.35	45.39	48.54	47.59	51.45	33.58	51.56
<b>InternVL3-8B + MotionEnhancer (Ours)</b>	1fps	48.94 <sup>↑3.12</sup>	49.25 <sup>↑2.90</sup>	47.17	57.11	46.57	56.35	36.74	51.56
Large Size Series									
LLaVA-NeXT-Video-34B [13]	8	30.44	32.58	31.70	31.99	32.31	22.99	29.58	46.88
LLaVA-Video-72B-Qwen2 [29]	64	46.08	46.49	48.35	47.50	45.25	51.70	33.02	53.12
Qwen2.5-VL-72B* [2]	1fps	48.14	48.17	50.28	46.98	48.13	51.78	40.28	51.56
InternVL2.5-78B [3]	8	38.54	38.36	38.38	40.62	39.05	43.65	29.40	39.06

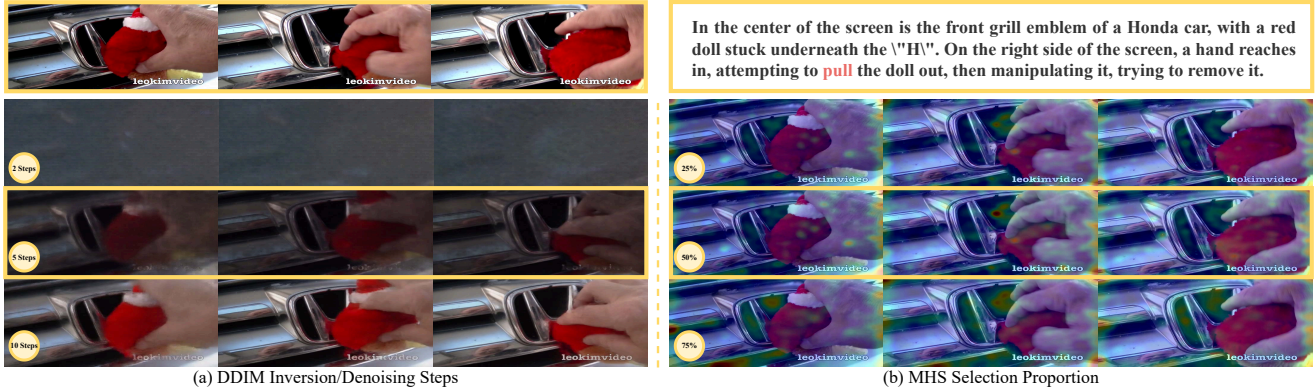


Figure 1. Visualization of different DDIM inversion/denoising steps and MHS selection proportion (left top: original video).

## 5. Experiments

### 5.1. System & Hardware Setups

The system and hardware details with GPU specifications of our experimental setup are provided in Tab. 1.

### 5.2. Benchmark Details

**Basic information.** In our experiments, we select two public benchmarks, namely MotionBench [11] and FAVOR-Bench [22], containing questions of different types from different video sources. They have been used in previous VLM for video understanding studies, making them suitable for evaluating the performance of our method. The details of these benchmarks are as follows:

- **MotionBench** is a benchmark with 5,385 videos and a total of 8,052 question-answer pairs. It is divided into a development set with 4,018 samples and a test set with 4,036 samples. MotionBench evaluates six core capabilities for understanding fine details in motion: Motion Recognition (MR), Location-related Motion (LM), Action Order (AO), Repetition Count (RC), Motion-related Objects (MO), and Camera Motion (CM). This allows for a comprehensive evaluation of motion-level perception. The videos come from diverse sources, including the web, existing public datasets, and synthetic videos created using Unity3. This mix ensures broad coverage of real-world applications. All data underwent careful human annotation and a multi-stage quality control process.
- **FAVOR-Bench** is a benchmark of 1,776 carefully selected videos covering diverse domains, each with detailed manual annotations of various motions. It evaluates models through both close-ended and open-ended tasks. For close-ended evaluation, it contains 8,184 challenging question-answer pairs across six tasks: Action Sequence (AS), Holistic Action Classification (HAC), Single Action Detail (SAD), Multiple Action Details (MAD), Camera Motion (CM), and Non-Subject Motion

(NSM). For open-ended evaluation, FAVOR-Bench offers both a novel cost-efficient LLM-free caption assessment method and a GPT-assisted evaluation approach.

**Benchmark Statistics.** We provide statistical information of the MotionBench dev set and FAVOR-Bench close-ended evaluation in Tab. 2, including each task type and its corresponding quantity.

### 5.3. Comparison with More VLMs

Given the page constraint of the main text, the comparative results of additional VLMs on the FAVOR-Bench benchmark are presented in Tab. 3 and Tab. 4. Empowered by MotionEnhancer, Qwen2.5-VL and InternVL3 not only outperform their counterparts within the same model size category but also attain performance comparable to that of larger-scale VLMs.

### 5.4. More Ablations

We conduct ablation studies on DDIM inversion/denoising steps and MHS/MTTI selection ratio. Tab. 5 shows that the number of DDIM steps has a clear impact on performance. Using only 2 steps leads to obvious degradation on both benchmarks, especially on FAVOR-Bench, indicating that overly coarse inversion/denoising fails to recover reliable motion-sensitive attention from the VDM. Increasing the number of steps consistently improves the results, as finer inversion and denoising better preserve temporally meaningful motion cues. However, more steps also introduce higher computational cost. The visual differences under different step settings are shown in Fig. 1(A), and we use 5 steps because the reconstructed dynamics are good for human eyes at low cost.

The attention head and text token selection ratio also affects performance by controlling the purity and coverage of the motion prior. Fig. 1(B) and Tab. 5 show the visual and quantitative changes under different ratios. A small ratio (25%) makes the prior overly sparse and may discard useful motion heads or motion-related text tokens, resulting in

Table 5. Ablation of DDIM inversion/denoising steps and MHS/MTTI selection ratio. **All results are trained with 25k data.**

Model Variant	MotionBench		FAVOR-Bench	
	Over.	Aver.	Over.	Aver.
Qwen2.5VL-7B	54.83	51.51	44.83	44.54
- 2 steps for DDIM	52.76	48.68	24.01	25.16
- 5 steps for DDIM	57.04	52.92	46.88	47.01
- 10 steps for DDIM	57.51	53.35	49.02	47.99
- 25% head/token select	55.33	51.00	47.12	47.28
- 50% head/token select	57.04	52.92	46.88	47.01
- 75% head/token select	54.88	50.69	45.69	44.10

incomplete motion supervision. In contrast, a large ratio (75%) introduces more non-motion attention, which weakens the motion signal and hurts alignment quality. The 50% setting gives the best trade-off between retaining sufficient motion information and filtering out irrelevant noise, leading to the most balanced overall performance. Therefore, we use 50% as a stable default.

### 5.5. MotionEnhancer VS. More Training Data

Tab. 6 analyzes the effectiveness of MotionEnhancer compared to direct data scaling on MotionBench and FAVOR-Bench using Qwen2.5-VL-7B as the baseline VLM. Notably, MotionVid-QA [5] is a large dataset containing 133k samples. While trained solely on MotionVid-QA, we observe clear improvements over the baseline VLM (row 2). Crucially, despite this substantial difference in training data volume, MotionEnhancer achieves superior performance on MotionBench (both Overall and Average scores) while delivering comparable performance on FAVOR-Bench with only 25k training data. These results highlight MotionEnhancer’s ability to achieve competitive or superior outcomes with significantly less training data, underscoring its data efficiency and effectiveness in enhancing motion understanding for VLMs.

### 5.6. Results on Conventional Benchmarks

**SEED-Bench** [12] is a large-scale benchmark with 19K human-annotated multiple-choice questions across 12 spatial and temporal dimensions on images and videos, designed to objectively evaluate the generative comprehension capabilities of multimodal large language models. **Video-MME** [6] is the first comprehensive benchmark for evaluating multimodal large language models on video understanding, featuring 900 manually curated videos across diverse domains and durations, with 2,700 expert-annotated multiple-choice questions. It uniquely incorporates subtitles and audio to assess models’ capabilities in handling complex, long-form, and multimodal video content.

As shown in Tab. 7, across both benchmarks, our

Table 6. Experimental results of MotionEnhancer VS. More Training Data on MotionBench and FAVOR-Bench using Qwen2.5-VL-7B as our backbone.

Model	MotionBench		FAVOR-Bench	
	Over.	Aver.	Over.	Aver.
Qwen2.5VL-7B	52.81	48.29	42.61	42.58
+ MotionVid-QA (133k)	55.70	51.25	47.12	47.28
+ MotionEnhancer (25k)	57.04	52.92	46.88	47.01

method demonstrates that enhancing motion modeling does not come at the cost of traditional video understanding performance. Instead, MotionEnhancer yields consistent improvements or preserves strong baselines, confirming its compatibility with general-purpose video comprehension tasks. For Qwen2.5-VL-7B, MotionEnhancer slightly reduces the SeedBench “All” score, but improves the video subset from 61.3 to 62.1 and maintains competitive image-level performance. On VideoMME, the model equipped with MotionEnhancer achieves an overall score of 61.4, while achieving competitive results on short, medium, and long video categories—demonstrating that temporal enhancements remain stable across video durations. For InternVL3-8B, the trend is similarly positive, where MotionEnhancer preserves strong SeedBench results. Importantly, on VideoMME, MotionEnhancer improves the overall score from 62.1 to 62.3, with consistent benefits across short and medium segments. This suggests that MotionEnhancer strengthens temporal reasoning without materially harming static-image understanding.

Overall, these results indicate that MotionEnhancer not only boosts performance on motion-centric tasks but also maintains or even subtly improves general video understanding ability. This demonstrates that injecting motion-aware priors does not introduce instability or unwanted trade-offs, reinforcing MotionEnhancer’s suitability as a lightweight and broadly applicable enhancement for video-language models.

## 6. Visualization Examples

### 6.1. A Complete Training Sample

To demonstrate the motion-aware token selection process, we provide a detailed visualization of a training instance in Fig. 3. The video depicts an elderly man performing a sequence of actions: raising his arms, clenching his fists, bowing his head, and speaking continuously. These actions are annotated with descriptive text, from which we extract tokens using the MTTI. Tokens such as “raise”, “bow”, “lowering”, and “speaking” receive high MS values, indicating strong correlation with motion dynamics. In contrast, static or contextual tokens like “his”, “the”, or “glasses” are as-

Table 7. Performance on SEED-Bench and VideoMME (w/o sub). \* denotes results we reproduced using their open-source code, while other results are taken from the original benchmark.

Model	SEED-Bench			VideoMME (w/o sub)			
	All	Image	Video	Overall	Short	Medium	Long
GPT-4V [1]	67.3	69.1	60.5	59.9	70.5	55.8	53.5
LLaVA-1.5-13B [17]	61.6	68.2	42.7	-	-	-	-
Weitu-VL-1.0-13B	69.2	74.2	50.5	-	-	-	-
SPHINXv2-1k-13B [16]	67.5	74.8	39.8	-	-	-	-
VideoLLaMA3-7B [28]	-	-	-	66.2	80.1	63.7	54.9
VideoChat-Flash-7B [14]	-	-	-	65.3	78.0	67.8	55.6
VITA1.5-7B [7]	-	-	-	56.1	67.0	54.2	47.1
Qwen2.5-VL-7B* [2]	74.1	77.5	61.3	60.7	70.9	59.6	51.6
<b>Qwen2.5-VL-7B+MotionEnhancer (Ours)</b>	72.9	75.9	62.1	61.4	72.8	61.1	50.2
InternVL3-8B* [30]	72.7	76.2	60.4	62.1	71.8	61.9	52.6
<b>InternVL3-8B+MotionEnhancer (Ours)</b>	72.4	75.8	60.6	62.3	73.0	62.2	51.7

signed low scores and are filtered out. This selection process ensures that only semantically and dynamically relevant tokens contribute to the motion prior.

Furthermore, we visualize the aggregated text-to-vision attention maps for the selected tokens in Fig. 2 and Fig. 4. These maps highlight spatial regions in the video frames that correspond to the described actions, confirming that the model attends to the correct locations where motion occurs. This not only validates the effectiveness of MTTI but also provides interpretable evidence of how MotionEnhancer grounds language in dynamic visual contexts.

## 6.2. DDIM Inversion and Reconstruction

Fig. 5 illustrates the DDIM inversion and reconstruction pipeline, a core component of our attention extraction mechanism. The process begins with the original video frames, which are encoded into latent representations using the VDM’s VAE. These latents are then inverted through a deterministic DDIM process to obtain a noise sequence that corresponds to the original video. This reversibility is crucial for ensuring that the attention maps extracted during the denoising process are semantically aligned with the original video content.

## 6.3. Video Understanding

We present qualitative comparisons on MotionBench (Fig. 6) and FAVOR-Bench (Fig. 7) between the baseline Qwen2.5-VL-7B model and its MotionEnhancer-enhanced version across a range of motion-centric tasks. These case studies underscore two key contributions of MotionEnhancer: *Improved temporal grounding*: the model better aligns textual descriptions with the actual sequence of events. *Enhanced motion sensitivity*: the model becomes more responsive to both foreground and background motion cues. Importantly, these improvements are achieved with-

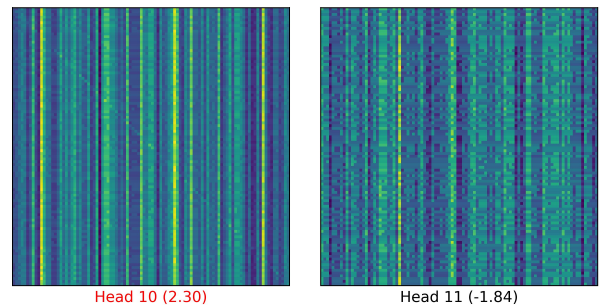
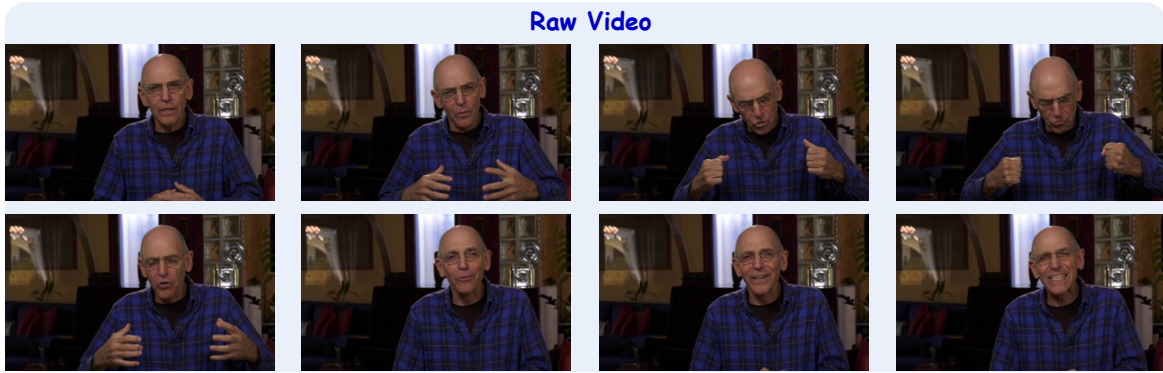


Figure 2. Zoom-in view of two specific heads. We selected the first 100 spatial location tokens for visualization. It is demonstrated that heads with high scores exhibit a diagonal pattern, which is consistent with the main paper.

out modifying the VLM architecture, highlighting the flexibility and generalizability of our attention alignment approach.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 9
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 6, 9
- [3] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 6
- [4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov,



Selected Text Token by MTTI

An elderly bald man wearing glasses, with a black shirt and a blue and black plaid shirt over it, is sitting towards the right of the center of the picture. He first raises his bent arms, with his palms open and facing his body, placed in front of his chest. Then he clenches his fists, bows his head in coordination with the hand gesture, and raises his head again before relaxing and lowering his arms from the bottom of the frame. Throughout the process, he is continuously speaking, and at the end, he even smiles and moves his eyebrows.

Motion Score

Selected	Word	raise	man	open	bow	lowering	speaking
	Score	0.73	0.85	0.83	0.74	0.61	1.29
Unselected	Word	even	his	from	in	the	glasses
	Score	-1.25	-0.42	-1.67	-0.85	-0.35	-1.50

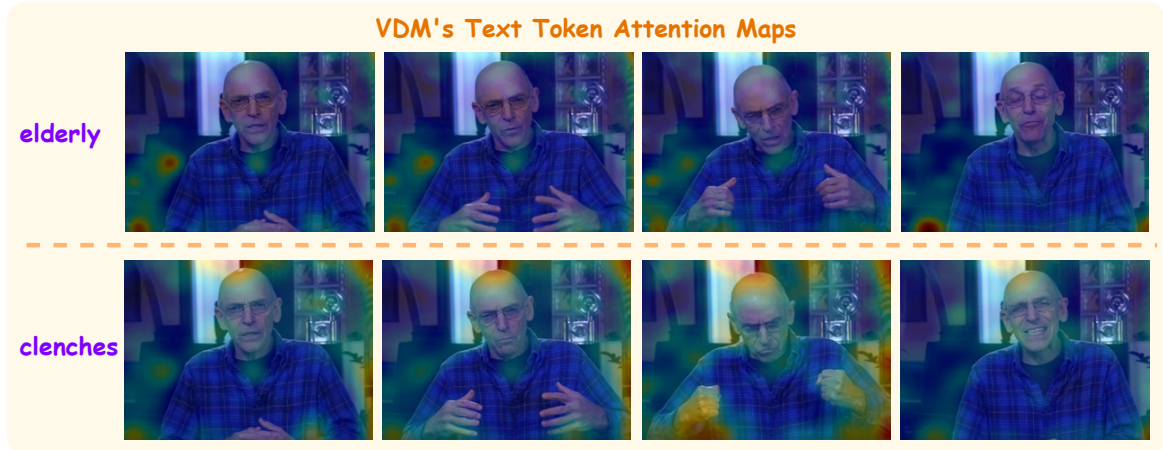


Figure 3. A complete training sample. We first show the selected text token by MTTI (in red). We then present the motion scores of some text tokens based on the mean value and the first-order difference mean. We further illustrate two aggregated text-to-vision attention maps.

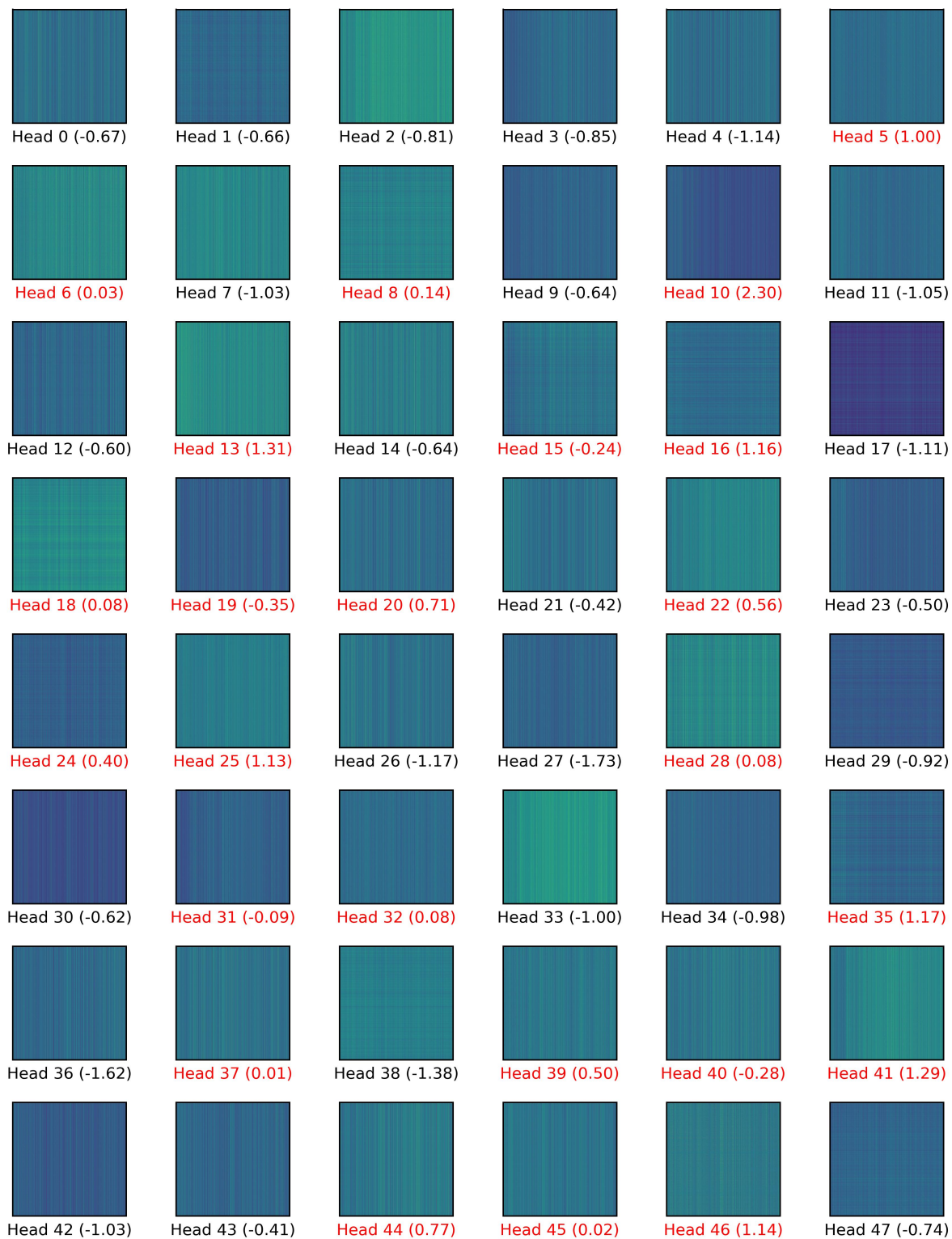
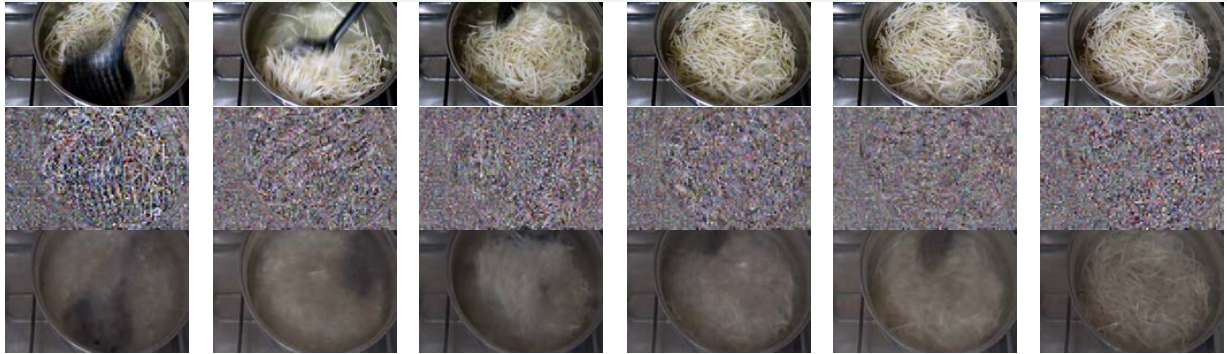


Figure 4. Vision-to-vision attention maps of different heads in VDM, with each head bearing a total score of DFC, TCS, and DSR. Heads in red are selected for aggregation.



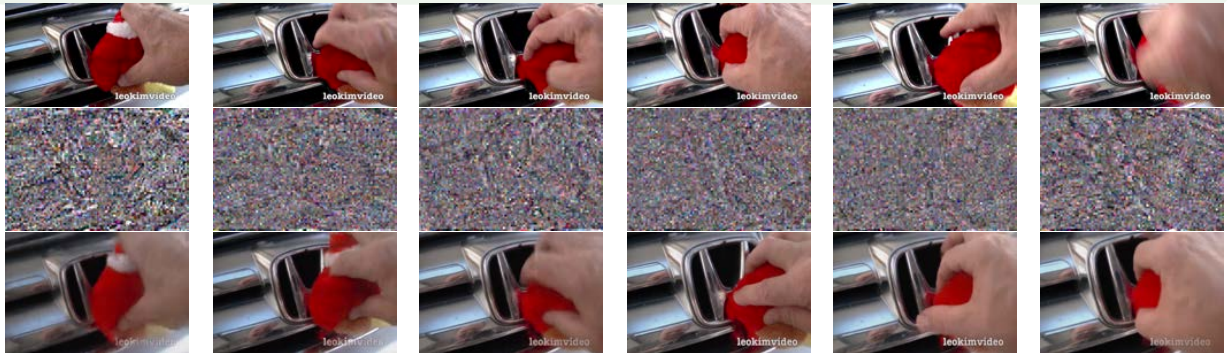
On the left side of the screen, there is a car with its door open. A man wearing sunglasses is getting out of the car. His body is facing towards the right of the screen, and with his left hand, he closes the car door and walks towards the camera.



The center of the screen shows a pot of soaked bean sprouts being cooked. At the top right of the screen, a black spoon is stirring the bean sprouts, lifting a spoonful and then setting it back down, gently manipulating the bean sprouts.



On the left side of the screen, a person is walking forward on a patch of grass with their back to the camera. Their right hand is placed behind the body, holding a hat, while the left arm swings back and forth on the left side of the body.



In the center of the screen is the front grill emblem of a Honda car, with a red doll stuck underneath the "H". On the right side of the screen, a hand reaches in, attempting to pull the doll out, then manipulating it, trying to remove it.

Figure 5. Our DDIM inversion and reconstruction process. Grouped in sets of three rows: the first row is the original video, the second row is the latent noise after DDIM Inversion, and the third row is the reconstructed video.



**Task Type:** Motion Recognition

**Question:** What does the woman do just before closing her eyes?

- A. She keeps her eyes open wide.
- B. She smiles widely.
- C. She removes a cotton swab herself.
- D. She furrows her brow.

**Qwen2.5VL-7B:** B. She smiles widely.

**Qwen2.5VL-7B + MotionEnhancer:** D. She furrows her brow.



**Task Type:** Location-related Motion

**Question:** What action does the hand perform first when it appears on screen?

- A. Places a shiitake mushroom on the right side of the pizza
- B. Places green pepper slices on the pizza
- C. Sprinkles cheese crumbs on the pizza
- D. Puts blueberries on the pizza

**Qwen2.5VL-7B:** C. Sprinkles cheese crumbs on the pizza

**Qwen2.5VL-7B + MotionEnhancer:** A. Places a shiitake mushroom on the right side of the pizza



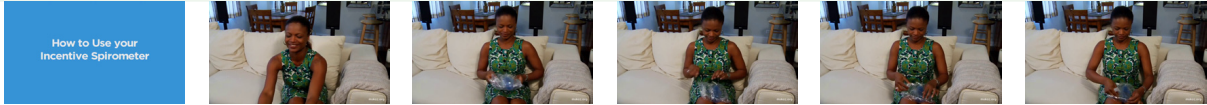
**Task Type:** Camera Motion

**Question:** What sequence describes the actions involving the camera and hand movements?

- A. Hand retracts, then camera pans left
- B. Hand extends, then camera zooms out
- C. Camera moves, then fingers extend and retract
- D. Fingers extend, then hand waves

**Qwen2.5VL-7B:** A. Hand retracts, then camera pans left

**Qwen2.5VL-7B + MotionEnhancer:** C. Camera moves, then fingers extend and retract



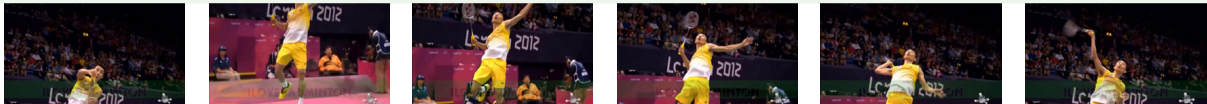
**Task Type:** Motion-related Objects

**Question:** What is the person wearing green clothes wearing on their left hand?

- A. Wristwatch
- B. Wrist guard
- C. Bracelet
- D. Ring

**Qwen2.5VL-7B:** D. Ring

**Qwen2.5VL-7B + MotionEnhancer:** A. Wristwatch



**Task Type:** Action Order

**Question:** Please describe the detailed breakdown of the action in the video.

- A. Jump up, lower left arm, swing right arm, catch the ball.
- B. Catch the ball, drop left arm, jump, swing right arm
- C. Jump up, drop left arm, swing right arm, catch the ball.
- D. Catch the ball, Swing right arm, Lower left arm, Jump

**Qwen2.5VL-7B:** C. Jump up, drop left arm, swing right arm, catch the ball.

**Qwen2.5VL-7B + MotionEnhancer:** A. Jump up, lower left arm, swing right arm, catch the ball.



**Task Type:** Repetition Count

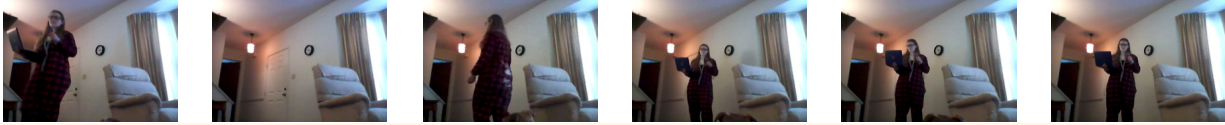
**Question:** Please count the number of repeated actions in the video.

- A. 2
- B. 6
- C. 4
- D. 8

**Qwen2.5VL-7B:** B. 6

**Qwen2.5VL-7B + MotionEnhancer:** A. 2

Figure 6. Qualitative results on MotionBench.



**Task Type:** Action Sequence

**Question:** After picking up the computer and walking to the right, which of the following action sequences does the woman in the red and black plaid shirt perform next?

- A. Walk to the front left → Stop → Look to the right
- B. Look to the right → Stop → Pick up the computer
- C. Stop → Look to the right → Walk to the front left
- D. Walk backward → Stop → Turn left and look forward
- E. Look to the right → Turn left and look forward → Stop

**Qwen2.5VL-7B:** E. Look to the right → Turn left and look forward → Stop

**Qwen2.5VL-7B + MotionEnhancer:** C. Stop → Look to the right → Walk to the front left



**Task Type:** Holistic Action Classification)

**Question:** What is the core behavior demonstrated by the man in white clothing throughout the video?

- A. Repeatedly pouring water and adjusting the position of the cup
- B. Continuously walking back and forth between different locations
- C. Rapidly opening and closing drawers while retrieving items
- D. Moving a chair then sitting down to organize documents
- E. Concentratedly flipping through contents in a folder.

**Qwen2.5VL-7B:** B. Continuously walking back and forth between different locations

**Qwen2.5VL-7B + MotionEnhancer:** D. Moving a chair then sitting down to organize documents



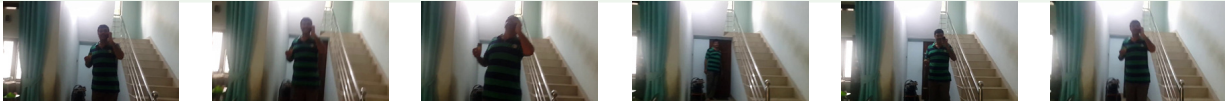
**Task Type:** Single Action Detail

**Question:** When the boy in green clothes performs a rollover maneuver, what is the primary dynamic movement of the body?

- A. Using both hands to push off the mattress and quickly sit up
- B. Swinging legs in the air to make tapping movements
- C. Repeatedly hitting the pillow with the head three times
- D. Keeping the upper body stationary while only moving the legs
- E. The body transitions from lying on the back to a side-lying position.

**Qwen2.5VL-7B:** C. Repeatedly hitting the pillow with the head three times

**Qwen2.5VL-7B + MotionEnhancer:** E. The body transitions from lying on the back to a side-lying position.



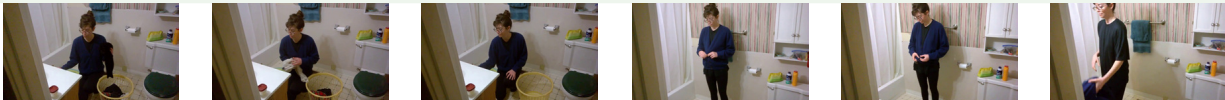
**Task Type:** Multiple Action Details

**Question:** In the video, what did the man in the black and green striped short-sleeve do?

- A. Put the phone in his pocket
- B. Stared at the phone screen and continuously typed
- C. Ran forward, took out the phone to answer a call, and then walked around
- D. Put something in his mouth to eat
- E. Continued to run forward and touched the camera lens.

**Qwen2.5VL-7B:** A. Put the phone in his pocket

**Qwen2.5VL-7B + MotionEnhancer:** C. Ran forward, took out the phone to answer a call, and then walked around



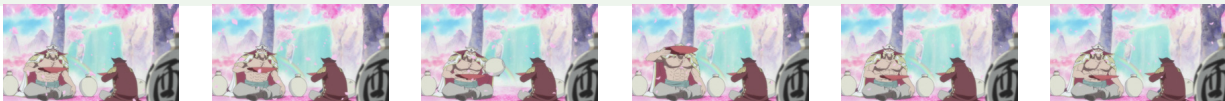
**Task Type:** Camera Motion

**Question:** When the woman wearing glasses and a dark blue shirt stands up, what is the main movement of the camera?

- A. Camera shake
- B. Upward movement
- C. Left and right movement
- D. Rotation
- E. Fast zoom

**Qwen2.5VL-7B:** E. Fast zoom

**Qwen2.5VL-7B + MotionEnhancer:** B. Upward movement



**Task Type:** Non-Subject Motion

**Question:** What are all the non-dominant natural environment movements in the video?

- A. Petals falling
- B. Waterfall pouring
- C. Treetops swaying and waterfall pouring
- D. Petals falling and waterfall pouring
- E. The hem of a character's clothing fluttering

**Qwen2.5VL-7B:** A. Petals falling

**Qwen2.5VL-7B + MotionEnhancer:** D. Petals falling and waterfall pouring

Figure 7. Qualitative results on FAVOR-Bench.

- Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2
- [5] Yipeng Du, Tiehan Fan, Kepan Nan, Rui Xie, Penghao Zhou, Xiang Li, Jian Yang, Zhenheng Yang, and Ying Tai. Motion-sight: Boosting fine-grained motion understanding in multi-modal llms. *arXiv preprint arXiv:2506.01674*, 2025. 6, 8
- [6] Chaoyou Fu, Yuhua Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025. 8
- [7] Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Yangze Li, Zuwei Long, Heting Gao, Ke Li, et al. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction. *arXiv preprint arXiv:2501.01957*, 2025. 9
- [8] Junqi Ge, Ziyi Chen, Jintao Lin, Jinguo Zhu, Xihui Liu, Jifeng Dai, and Xizhou Zhu. V2pe: Improving multi-modal long-context capability of vision-language models with variable visual position encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21070–21084, 2025. 2
- [9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1
- [10] Wenyi Hong, Weihang Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024. 6
- [11] Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihang Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8450–8460, 2025. 6, 7
- [12] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 8
- [13] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 6
- [14] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhua Zhu, Haihan Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024. 6, 9
- [15] Bin Lin, Yang Ye, Bin Zhu, Jiayi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 6
- [16] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023. 9
- [17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 9
- [18] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6038–6047, 2023. 1
- [19] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 1
- [20] 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. 2
- [21] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1
- [22] Chongjun Tu, Lin Zhang, Pengtao Chen, Peng Ye, Xianfang Zeng, Wei Cheng, Gang Yu, and Tao Chen. Favor-bench: A comprehensive benchmark for fine-grained video motion understanding. *arXiv preprint arXiv:2503.14935*, 2025. 7
- [23] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024. 6
- [24] Weiyun Wang, Zhe Chen, Wenhui Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, et al. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024. 2
- [25] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024. 6
- [26] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 1
- [27] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 6
- [28] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 6, 9
- [29] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 6

- [30] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 2, 6, 9