University of Padova

Department of Department of Mathematics

Master Thesis in Data Science

Scalable Content Analysis of Social Media Videos Using Multimodal Large Language Models: A Video-to-Text Pipeline for Large-Scale Analysis

Supervisor
Tomaso Erseghe
University of Padova

Co-supervisor
Maria Laura Bettinsoli
University of Padova

Francesca Guizzo
University of Padova

Master Candidate
Matteo Gorni Silvestrini

Academic Year
2025-2026

A mia mamma

Abstract

The analysis of large-scale video content remains a significant challenge in social science research due to the high cost and complexity of manual annotation. This thesis investigates the use of Multimodal Large Language Models (MLLMs) as a scalable solution for the analysis of video data, with a specific focus on the study of sexualization in social media content.

A dataset of TikTok videos from Italy, the United States, and South Korea was constructed and analyzed using a structured codebook derived from prior literature on sexual objectification. A multimodal model was employed to generate both content coding annotations and textual descriptions of video content, enabling a unified video-to-text analytical pipeline.

Results indicate that MLLMs can support large-scale analysis of video content, capturing consistent patterns aligned with theoretical expectations. In particular, systematic differences in sexualization were observed across gender, while cross-national variations were also identified. Complementary analyses of the generated textual descriptions provide additional evidence that the extracted signal reflects meaningful characteristics of the underlying content.

Contents

Listing of figures

Listing of tables

Listing of acronyms

ANOVA ..... Analysis Of Vartiance

BLV ..... Blind and Low-Vision

CDF ..... Cumulative Distribution Function

CLIP ..... Contrastive Language–Image Pre-training

CoT ..... Chain-of-Thought

DEI ..... Dipartimento di Ingegneria dell'Informazione

FPS ..... Frames Per Second

GPU ..... Graphics Processing Unit

HPC ..... High-Performance Computing

IT ..... Italy

JSON ..... JavaScript Object Notation

KR ..... South Korea

MLLM ..... Multimodal Large Language Model

MLP ..... Multi-Layer Perceptron

MoE ..... Mixture-of-Experts

mRoPE ..... Interleaved Multimodal Rotary Positional Embedding

MVBench ..... Multi-Modal Video Benchmark

NFS ..... Network File System

OOM ..... Out-Of-Memory

Q3VL Qwen3-VL

QWK ..... Quadratically Weighted Cohen's Kappa

RoPE . . . . . Rotary Positional Embedding

SBATCH . . . . . Slurm Batch Job Submission Command

SLURM . . . . . Simple Linux Utility for Resource Management

Tukey HSD . . . . . Tukey Honestly Significant Difference

US . . . . . United States

Video-MME . . . . Multi-Modal Evaluation for Video Analysis

VLM . . . . . Vision-Language Model

YAML . . . . . YAML ain't markup language

1 Introduction

1.1 Motivation

Social media platforms have fundamentally reshaped communication, establishing digital environments characterized by large-scale and continuous user interaction. Within this landscape, TikTok has emerged as one of the most influential platforms. Launched in 2016 by ByteDance, TikTok enables users to create and share short-form video content. The platform has rapidly expanded to an estimated 1.6 billion users globally as of 2025 [1]. Although initially known for its popularity among adolescents, TikTok has increasingly gained traction among young adults, with approximately 75% of its users being under the age of 35 [2].

Psychology has shown growing interest in understanding the effects of social media consumption on users. However, accurately characterizing these digital environments remains challenging. Social media companies provide limited access to detailed user data, while platform features, algorithms, and interaction mechanisms are continuously evolving. These factors complicate the systematic study of user behavior and exposure[3]. As a result, researchers have often relied on self-report questionnaires and simple quantitative proxies, such as time spent on a platform, to operationalize social media use[4].

However, these measures provide only indirect and often inaccurate representations of actual user behavior. A growing body of literature has shown that self-reported measures of digital media use are prone to substantial inaccuracies and systematic biases, including recall bias

and subjective misestimation [4]. In particular, discrepancies between self-reported and objectively logged data have been consistently observed, with self-reports only moderately correlated with actual usage patterns [4].

These limitations suggest that self-reported measures capture users' perceptions of their behavior rather than their true activity, thereby constraining the validity of research findings. Consequently, recent work has emphasized the need for objective, data-driven approaches to the study of social media behavior.

In this context, machine learning and data science techniques provide a powerful set of tools for directly analyzing social media content at scale. These approaches enable the automated analysis and classification of large volumes of visual and textual data, allowing researchers to objectively characterize the content present on platforms such as TikTok.

1.2 Context

The work presented in this thesis originated from a collaboration with the Department of Psychology, established through academic supervision. The project is based on a dataset of approximately 18,000 TikTok videos, collected through a structured methodology from content creators across three countries: Italy, the United States, and South Korea. At the outset, the project did not have a clearly defined computational task. While the broader research objective, investigating aspects related to the sexualization of social media content, was established, the specific analytical approach had yet to be determined. In particular, this work was tasked with focusing on the video content itself, as opposed to metadata or user-level information.

1.2.1 Task Definition

The initial phase of the project therefore involved exploratory analysis aimed at identifying a suitable task for processing the video dataset. Ultimately, the transformation of visual content into textual representations was selected as the most appropriate approach.

This strategy offers several advantages. Textual data is generally more tractable than raw visual data, enabling more efficient downstream processing and reducing computational complexity. Additionally, textual representations provide a compact and semantically meaningful description of video content, facilitating large-scale analysis of patterns related to sexualization.

1.2.2 Computational Approach: Leveraging Multimodal Large Language Models

A video can be modeled as a sequence of images, where temporal progression introduces an additional layer of meaning. Treating video frames as independent images risks discarding this temporal context. Therefore, an approach capable of explicitly modeling the temporal structure of video data is required.

Multimodal Large Language Models (MLLMs) provide a suitable framework for this project. These models are capable of processing visual inputs, including images and videos, and generating textual outputs—either structured or unstructured—that encode objects, relationships, and higher-level semantic concepts. Importantly, many MLLMs are designed to process temporal sequences, enabling the capture of dynamics across frames, which is essential for understanding behaviors and interactions within video content [5].

Another key advantage of MLLMs is their strong performance in zero-shot and few-shot settings. This allows the model to perform effectively without requiring large, manually annotated datasets for task-specific fine-tuning, which would be impractical in this context.

To identify the most appropriate model for this project, the following selection criteria were defined:

- **High performance:** The model should demonstrate strong empirical results validated by the research community, ensuring reliability in processing video content.
- **Open-weights availability:** The model must be deployable locally to ensure reproducibility and to avoid reliance on proprietary or paid APIs.
- **Configurability:** The model should be available in multiple sizes, allowing adaptation to different computational resource constraints.

Based on these criteria, **Qwen-3 VL** was selected as the model of choice [6].

1.3 Objectives

Based on the preliminary considerations and the computational approach outlined above, the following objectives were defined:

- **Automated content analysis:** The model is used to replicate a task traditionally performed by human annotators, namely the assessment of sexualization in video content.

The goal is to produce labels indicating the presence and degree of markers considered in the literature as proxies for sexualized content, providing a scalable and consistent alternative to labor-intensive human coding.

- **Video-to-text conversion:** The model generates textual descriptions of each video. This step serves multiple purposes. First, as previously discussed, textual data is generally easier to manipulate and analyze than raw video data. Second, the dataset contains a wide variety of popular videos from different creators, and its content is not limited to sexualized material. Converting videos to text enables the extraction of semantic information across the full dataset, which represents a key output of this work and may support future analyses.

1.3.1 Analytical Framework and Research Questions

Building upon the objectives outlined above, this work evaluates both the performance of the proposed approach and its usefulness as a tool for large-scale content analysis.

The first aspect concerns the performance of the model on the content coding task. In particular, this study examines the extent to which the model can analyze video content and how its performance compares to human annotation.

The second aspect focuses on the type of information that can be extracted from the dataset using the proposed methodology. In this context, the following dimensions are explored:

- **Gender differences:** Prior research has examined the relationship between gender and the production of sexualized content on social media [7]. This work investigates whether systematic differences in the prevalence and intensity of sexualized content emerge across gender groups within the present dataset.
- **Cross-national differences:** The dataset includes content from three countries, enabling the exploration of potential differences in patterns of sexualization across national contexts.

1.4 Work Structure

This work is structured as follows. Chapter 2 provides the necessary theoretical background, including the psychological constructs underlying the study, the model architecture, and the evaluation metrics.

Chapter 3 presents the methodology , including a detailed description of the dataset and the computational procedures used in this work.

Chapter 4 describes the experiments conducted, including model configurations and any fine-tuning strategies employed to improve performance.

Finally, Chapter 5 presents a statistical analysis of the results and discusses the findings in relation to the research questions.

2 Theoretical Background

2.1 Social Media, Body Image, and Sexual Objectification

2.1.1 Body Image

Body image refers to an individual's perceptions, thoughts, and feelings about their physical appearance [8]. It encompasses both perceptual components (how one perceives their body) and attitudinal components (how one evaluates and feels about their body). Body image is influenced by a complex interplay of sociocultural, psychological, and biological factors. Sociocultural influences include media representations, cultural norms, and peer interactions that shape ideals of attractiveness and body standards [8]. Psychological factors, such as self-esteem, personality traits, and mental health status, also play a significant role in shaping body image [8].

2.1.2 Sexual Objectification

Sexual objectification can be defined as the treatment of a person primarily as a body or as a collection of body parts, with emphasis placed on sexual appearance over attributes such as competence, agency, or individuality [9].

Sexual objectification can be understood as a sociocultural process reinforced through multiple sources:

- **Interpersonal level:** verbal and non-verbal evaluations of a person's body, unsolicited comments, and unwanted sexual advances.
- **Cultural level:** repeated exposure to objectified representations of bodies in traditional and social media.

Together, these influences promote appearance-based evaluation and normalize the prioritization of physical attractiveness, particularly for women [9].

This conceptualization is grounded in objectification theory [10], which connects cultural practices of objectification to measurable psychological consequences. Objectification theory posits that when individuals are repeatedly treated as objects valued primarily for their appearance, they may internalize this external perspective. Over time, self-objectification fosters chronic body surveillance and heightened appearance monitoring, diverting cognitive and emotional resources toward evaluating one's physical appearance [10].

Exposure to objectified representations has been associated with a range of negative outcomes. At the individual level, increased self-objectification has been linked to poorer body image, reduced psychological well-being, and impaired cognitive performance, as attentional resources are diverted toward monitoring one's appearance.

From an interpersonal perspective, objectification also influences how individuals are perceived and treated: objectified individuals are more likely to be judged as less competent and less fully human, which may contribute to discriminatory attitudes and reinforce broader gender inequalities. Importantly, contemporary social media environments may amplify these effects. Platforms that prioritize visual content and provide immediate appearance-based feedback, such as likes and comments, can intensify appearance monitoring and normalize constant self-presentation, thereby reinforcing the dynamics described by objectification theory [9].

2.1.3 TikTok's Algorithm and Negative Psychological Outcomes

In recent years, social media platforms have become a central component of users' social and psychological environments. Their widespread adoption has led to increasing interest in understanding their impact on mental health and well-being across different populations. Among these platforms, TikTok has experienced particularly rapid growth and is now one of the most widely used applications globally. Its design is based on short-form, algorithmically curated

video content, exposing users to a continuous stream of highly engaging and personalized material. This structure increases both the intensity and frequency of exposure to socially evaluative and appearance-focused content, making the platform particularly relevant for studying processes related to body image and psychological well-being [11].

A growing body of literature has begun to examine the relationship between TikTok use and mental health. A recent systematic review by Conte et al. [11] synthesized findings from multiple empirical studies across different countries and populations, identifying consistent associations between TikTok use and several psychological outcomes. These include changes in body image, self-esteem, anxiety, and depressive symptoms [12]. While these findings point to a meaningful association between TikTok use and psychological outcomes, most available evidence is based on cross-sectional designs. As a result, causal interpretations remain limited, and the relationship between usage patterns and psychological distress is likely complex and bidirectional. Nevertheless, the consistency of these findings suggests that certain features of the platform may contribute to the emergence or reinforcement of psychological vulnerabilities.

A central aspect highlighted by recent research concerns the role of TikTok's recommendation system in shaping user experience. Unlike traditional platforms where algorithms primarily facilitate interactions within a user's existing social network, TikTok's algorithm determines content exposure based on inferred individual interests. By generating a highly personalized *For You* feed that adapts in real time to user interactions, the platform shifts the focus from social connections to content-driven engagement.

As a result, content consumption is strongly influenced by prior interactions, increasing the likelihood that users are repeatedly exposed to similar themes. This feedback loop may reinforce specific appearance-related standards and patterns of engagement. From a psychological perspective, repeated and personalized exposure to visually salient content may intensify attention toward socially valued physical attributes. This mechanism aligns with objectification theory [10], suggesting that algorithmically curated environments that prioritize high-engagement visual content may foster conditions conducive to self-objectification, wherein individuals learn to view themselves from an external observer's perspective.

2.2 Multimodal Large Language Models

2.2.1 Defining MLLMs

Vision–Language Models (VLMs) are machine learning models designed to learn joint representations of visual and textual data. By aligning information across these two modalities, VLMs enable tasks such as image captioning, visual question answering, cross-modal retrieval, and multimodal reasoning. These models learn a shared embedding space in which both images and text can be represented in a compatible form, allowing semantic relationships between visual and linguistic inputs to be captured.[5] MLLMs are a subclass of VLMs. In this architecture, an LLM serves as the core computational component and is extended with modality-specific encoders and projection layers. in the case of images, The visual encoder transforms an image into a sequence of feature vectors, which are then projected into the embedding space of the language model. These projected visual tokens are processed together with textual tokens within the language model's transformer architecture, enabling the model to generate text conditioned directly on visual inputs. [5]

2.2.2 Evolution of Vision–Language Models

Vision–language models have evolved significantly over the past decade, progressing from task-specific multimodal architectures to large-scale multimodal language models capable of flexible reasoning and generation. Early multimodal research focused on combining separate vision and language encoders using task-specific fusion mechanisms. Models such as ViLBERT employed transformer-based architectures with cross-modal attention layers to integrate visual and textual features.[13] While these approaches demonstrated strong performance on benchmark tasks such as visual question answering and image–text retrieval, they were typically trained in a supervised setting and showed limited generalization beyond the specific tasks and datasets on which they were trained. A major breakthrough in the field was the introduction of Contrastive Language–Image Pre-training (CLIP)[14]. CLIP reframed vision–language learning as a large-scale contrastive alignment problem, training separate image and text encoders on hundreds of millions of image–text pairs collected from the web. The model learned to project images and text into a shared embedding space, enabling similarity-based reasoning across modalities. This approach allowed CLIP to perform zero-shot classification, meaning it could recognize visual concepts without explicit task-specific training, significantly improving generalization and establishing a new paradigm for multimodal learning. Following CLIP, research shifted toward

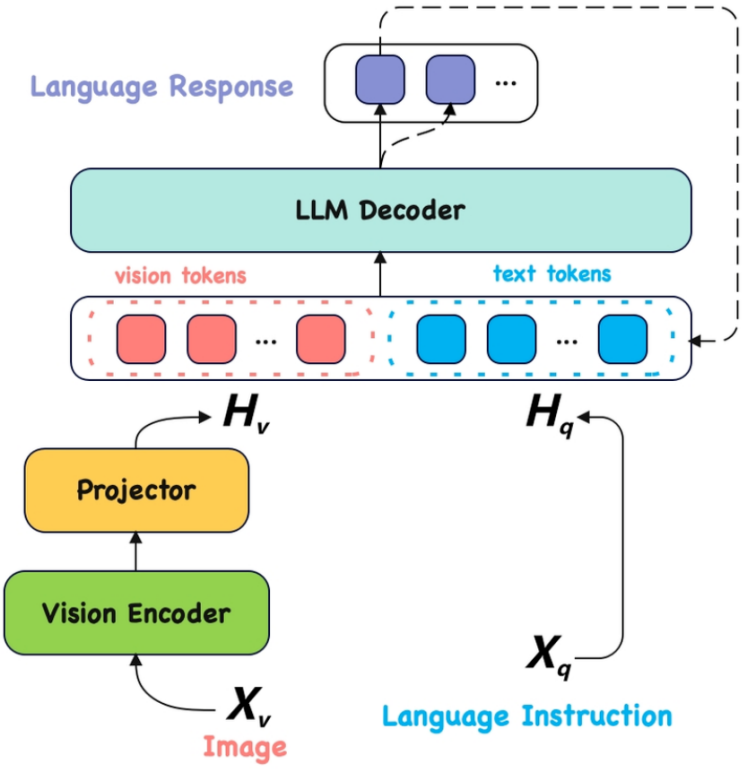

Figure 2.1: The figure illustrate the information flow of a standard multimodal llm, highlighting its three key components.

integrating pretrained language models with visual encoders to enable generative and reasoning capabilities. DeepMind's Flamingo [15] was among the first multimodal large language models to combine visual and textual inputs within an autoregressive transformer architecture. Flamingo introduced cross-attention mechanisms that allowed the language model to attend to visual features during text generation, enabling few-shot multimodal prompting and expanding the range of tasks these systems could perform. Subsequent work further improved the efficiency and scalability of multimodal language models. BLIP-2 [16] demonstrated that effective vision–language integration could be achieved by keeping both the vision encoder and language model frozen while learning a lightweight query transformer to connect the two modalities. This approach significantly reduced the number of trainable parameters while maintaining strong performance across a variety of multimodal tasks. In 2023, instruction-tuned multimodal models such as LLaVA [17] further advanced the capabilities of these systems. LLaVA combines visual features extracted from pretrained vision encoders with large language models and trains them on multimodal instruction–response pairs. This instruction-tuning pro-

cess enables the model to interpret visual inputs in context, follow multimodal instructions, and generate coherent natural language responses grounded in visual information. Instruction tuning marked an important shift toward more general-purpose multimodal assistants capable of interactive and conversational behavior. More recently, large-scale multimodal language models such as GPT-4V [18] and Qwen-VL have demonstrated substantial improvements in multimodal reasoning, image understanding, and instruction-following capabilities.

2.2.3 An Introduction to the Qwen Model Family

The Qwen model family is a collection of LLMs and MLLMs developed by Alibaba Cloud. These models are part of an ongoing effort to advance AI research and its applications, with multiple versions released publicly through platforms such as Hugging Face. The Qwen development team continuously releases new models spanning a wide range of sizes and capabilities, including language, multimodal, speech, and image processing models, thereby making them widely accessible for research and development purposes [19].

The increasing availability and adoption of Qwen models can be observed through community usage trends. Figure 2.2 illustrates the evolution in the number of model downloads on the Hugging Face platform for several major open model families over time, including Qwen.

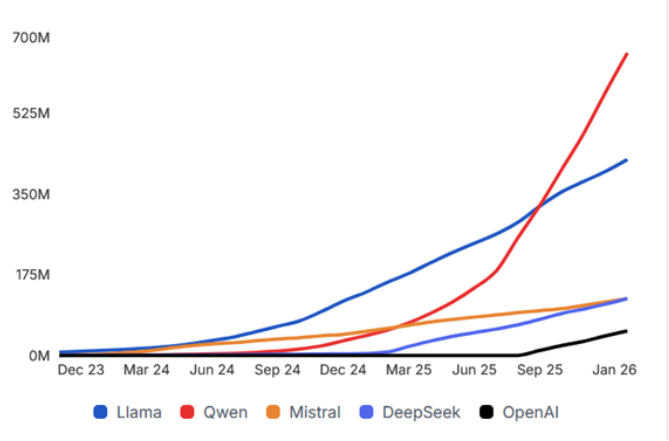

Figure 2.2: Popularity trends of the Qwen model family over time.

2.2.4 Qwen3-VL

Released in September 2025, Qwen3-VL (hereafter referred to as Q3VL) is the most recent iteration of Qwen's vision–language model family. The model has been released in four dense

variants, with 2B, 4B, 8B, and 32B parameters, as well as two Mixture-of-Experts (MoE) variants: Q3VL-30B-A3B and Q3VL-235B-A22B. In the MoE naming convention, the first number indicates the total number of parameters (in billions), while the second number specifies the number of parameters that are activated during inference.[6]

The following subsections provide an overview of the model, describing its architecture, the key innovations proposed by the authors, as well as details regarding the dataset, training procedure, and performance.

Architecture

Q3VL follows a three-module architecture [6] composed of: (i) a vision encoder that supports dynamic, native-resolution inputs; (ii) a lightweight multi-layer perceptron (MLP)mbased vision language merger that compresses visual features into tokens aligned with the hidden dimension of the language model; and (iii) a Qwen3 LLM backbone with a varying number of parameters depending on the model variant.

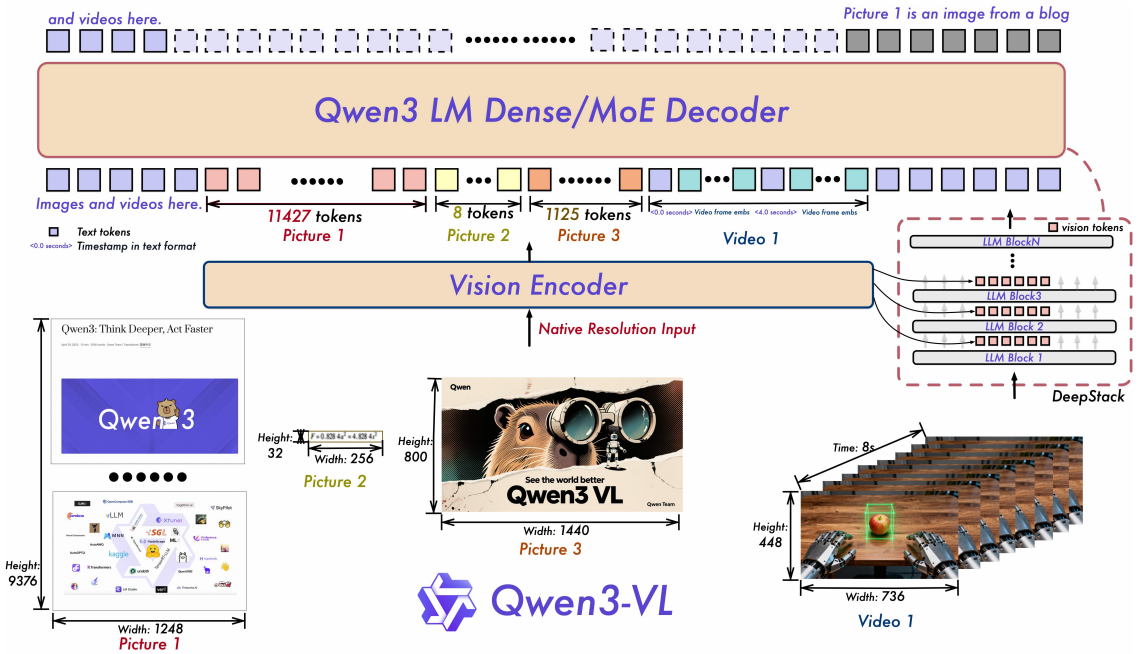

Figure 2.3: Architecture of the Q3VL model.

Key architecture changes

1. Interleaved Multimodal Rotary Positional Embedding (mRoPE) RoPE encodes positional information by applying a rotation to pairs of embedding dimensions in the query and key vectors.[20] For a token at position p and embedding pair (x_{2i}, x_{2i+1}), the rotation is defined as:

$$\begin{pmatrix} x'_{2i} \\ x'_{2i+1} \end{pmatrix} = \begin{pmatrix} \cos(\theta_i p) & -\sin(\theta_i p) \\ \sin(\theta_i p) & \cos(\theta_i p) \end{pmatrix} \begin{pmatrix} x_{2i} \\ x_{2i+1} \end{pmatrix}, \tag{2.1}$$

where the frequency θ_i is defined as:

$$\theta_i = \frac{\theta_0}{10000^{\frac{2i}{d}}}, \tag{2.2}$$

and d denotes the embedding dimension. This formulation naturally encodes relative positional information, since attention scores depend on relative position differences.

mRoPE extends this concept to multidimensional inputs such as videos, with temporal and spatial axes corresponding to time (t), height (h), and width (w).[21] In the original formulation, embedding dimensions are partitioned into contiguous blocks:

$$[x_0, \ldots, x_{d-1}] = [t_0, \ldots, t_{d_t-1} \mid h_0, \ldots, h_{d_h-1} \mid w_0, \ldots, w_{d_w-1}], \tag{2.3}$$

where each axis is assigned its own frequency range.

This block-wise allocation introduces spectral bias, as individual axes may only access limited frequency bands, restricting their ability to represent both fine and coarse positional relationships.

Interleaved mRoPE addresses this limitation by interleaving embedding dimensions across axes:

$$[x_0, \ldots, x_{d-1}] = [t_0, h_0, w_0, t_1, h_1, w_1, \ldots], \tag{2.4}$$

ensuring that each axis has access to the full set of rotation frequencies $\{\theta_i\}$. This improves spatial and temporal representation and enhances performance on long-range video reasoning tasks.[21]

2. DeepStack Visual Feature Injection DeepStack is a hierarchical feature integration method that injects visual features into multiple layers of the language model. Traditional architectures typically inject visual information into a single intermediate layer, limiting the model's ability to propagate hierarchical visual representations.[22]

Let $V^{(l_v)}$ denote visual features from the l_v-th layer of the vision encoder, and let $L^{(l_l)}$ denote the hidden state of the l_l-th layer of the language model. DeepStack projects visual features into the language model using a learned projection matrix $W_{l_v \to l_l}$:

$$\tilde{V}^{(l_l)} = W_{l_v \to l_l} V^{(l_v)}, \tag{2.5}$$

$$L^{(l_l)} \leftarrow L^{(l_l)} + \tilde{V}^{(l_l)}. \tag{2.6}$$

This process is repeated across multiple layer pairs, for example:

$$V^{(3)} \to L^{(1)}, \quad V^{(6)} \to L^{(3)}, \quad V^{(9)} \to L^{(5)}. \tag{2.7}$$

This hierarchical integration enables the language model to access both low-level and high-level visual features, improving multimodal reasoning.

3.Textual Time Encoding To improve temporal modeling in video inputs, Q3VL adopts a textual token–based time encoding strategy. Instead of assigning large numerical temporal position IDs, each temporal patch is prefixed with a textual timestamp token, for example:

```
<3.0 seconds>
```
(2.8)

During training, timestamps are expressed in multiple formats, including seconds and hours:minutes:seconds (HMS). This allows the model to learn temporal relationships through standard language modeling mechanisms.

Although this approach slightly increases sequence length, it improves performance on temporally grounded tasks such as video grounding and dense captioning.

2.2.5 Training procedure

The training of Q3VL is divided between a pre-training phase and a post training phase, the former is organized into four distinct stages,while the latter consists of three methods.[6]

Table 2.1: Summary of objectives, token budgets, and sequence lengths for each step of the pre-training phase (S0–S3).

Stage	Objective	Training Token Budget	Sequence Length
S0	Vision-Language Alignment	67B	8,192
S1	Multimodal Pre-Training	∼1T	8,192
S2	Long-Context Pre-Training	∼1T	32,768
S3	Ultra-Long-Context Adaptation	100B	262,144

Vision-Language Alignment The initial stage (S0) focuses on bridging the modality gap between the vision encoder and the LLM. During this phase, only the parameters of the MLP-based merger are trained, while both the vision encoder and the LLM backbone remain frozen. The model is trained on a curated dataset of approximately 67 billion tokens, which includes high-quality image-caption pairs, visual knowledge collections, and optical character recognition (OCR) data. The sequence length is set to 8,192.

Multimodal Pre-Training Stage 1 (S1) involves full-parameter multimodal pre-training, where the vision encoder, merger, and LLM are all trainable. The model is trained on a massive and diverse dataset of approximately 1 trillion tokens, consisting of both vision-language (VL) and text-only data. The VL portion includes interleaved image-text documents, visual grounding tasks, visual question answering (VQA),and a small amount of video data for temporal understanding. The sequence length remains 8,192 tokens.

Long-Context Pre-Training Stage 2 (S2) is designed to extend the model's contextual processing capabilities. The sequence length is increased to 32,768 tokens, while all model parameters remain trainable. Training is conducted on a dataset of approximately 1 trillion tokens, with a different data mixture favoring long-form text comprehension. The remaining VL data includes a larger volume of video and instruction-following data, supporting multi-step reasoning over extended contexts.

Ultra-Long-Context Adaptation The final stage (S3) aims to maximize the model's context window. The sequence length is increased to 262,144 tokens, and the training dataset is focused and reduced to approximately 100 billion tokens. The data consists of both text-only

and vision-language content, with an emphasis on long-video and long-document understanding tasks.

Post-Training Phase and Model Split

The model also undergoes a post-training procedure, which involves techniques such as supervised fine-tuning, strong-to-weak distillation [23], and reinforcement learning [24]. An in-depth discussion of this procedure [6] and the underlying methods is beyond the scope of this work. It is, however, worth noting that during this phase, both the training procedure and the dataset used are split to create two distinct variants of the model: an *Instruct* variant and a *Thinking* variant.

- **Instruct Variant:** Optimized for direct instruction following, this variant is trained to produce concise, well-formatted, and user-aligned responses, without explicitly revealing intermediate reasoning steps.
- **Thinking Variant:** Optimized for Chain-of-Thought (CoT) reasoning , this variant performs explicit step-by-step thinking by decomposing problems into intermediate reasoning steps before generating the final answer. It is particularly suited for multi-step analytical tasks, such as mathematical reasoning and logical inference.

2.2.6 Model Performance

The Alibaba Cloud team evaluated various versions of the Q3VL series across a wide range of benchmarks spanning general visual reasoning, multimodal tasks, and video understanding. Since the task performed in this work concerns video understanding, I focus on two widely used and community-recognized benchmarks in this domain: Multi-Modal Video Benchmark (MVBench) and Multi-Modal Evaluation for Video Analysis (Video-MME) [25, 26].

- **MVBench.** MVBench is designed to assess the temporal understanding capabilities of multimodal large language models by transforming static visual tasks into dynamic video question-answering challenges across 20 distinct temporal reasoning categories, including action prediction, object interaction, and episodic reasoning. Models are evaluated using accuracy over multiple-choice question-answer pairs that require reasoning over entire video sequences rather than single frames.

- **Video-MME.** Video-MME is a comprehensive benchmark for evaluating video understanding across diverse visual domains and temporal durations, ranging from short clips to videos up to one hour in length. It contains 900 manually annotated videos and 2,700 question-answer pairs and measures model accuracy in answering questions based on video content. Performance is reported in two settings: *with subtitles*, where the text transcript is provided alongside visual input, and *without subtitles*, where evaluation relies solely on visual information. In this work, results reported are from the latter setting.

Table 2.2 shows that the Q3VL models achieve competitive performance across both benchmarks, with clear improvements as model size increases. While GPT-5 High achieves the highest reported scores, the Q3VL models remain competitive, particularly in larger configurations, suggesting that the proposed architecture provides a strong baseline for video understanding tasks, even when compared to state-of-the-art models.

Table 2.2: Performance comparison on MVBench and Video-MME benchmarks. All values are reported as accuracy (%).

Model	MVBench (%)	Video-MME (%)
Q3VL-2B (Thinking)	64.5	62.1
Q3VL-2B (Instruct)	61.7	61.9
Q3VL-4B (Thinking)	69.3	68.9
Q3VL-4B (Instruct)	68.9	69.3
Q3VL-8B (Thinking)	69.0	71.8
Q3VL-8B (Instruct)	68.7	71.4
Q3VL-32B (Thinking)	73.2	77.3
Q3VL-32B (Instruct)	**72.8**	**76.6**
Q3VL-235B-A22 (Thinking)	75.2	79.0
Q3VL-235B-A22 (Instruct)	76.5	79.2
Q3VL-30B-A3 (Thinking)	72.0	73.3
Q3VL-30B-A3 (Instruct)	72.3	74.5
GPT-5 High	**75.3**	**84.7**

2.3 Computational Approaches to Video Description and Annotation

This section discusses the two core tasks addressed in this thesis: generating textual descriptions of video content and producing structured annotations. In addition to describing these tasks, the section outlines the computational methods used to perform them.

Before discussing these methods in detail, it is important to note that interaction with models such as Q3VL largely occurs through prompts. The design and structuring of these prompts, commonly referred to as *prompt engineering*, plays a crucial role in shaping the outputs produced by the model, and is therefore central to both tasks considered in this work.

Prompt Engineering Prompt engineering refers to the design of carefully structured, task-specific queries, known as prompts, to guide a model toward producing the desired output without modifying the model's underlying parameters [27]. By specifying instructions, constraints, and contextual information within the prompt, it is possible to influence both the structure and the content of the model's responses.

Furthermore, some models, including Q3VL, employ a dual-prompt structure consisting of a system prompt and a user prompt. The system prompt provides high-level instructions, context, and constraints that define the model's role, behavior, and response style throughout the interaction, thereby establishing the operational framework within which the model operates. The user prompt, in contrast, contains the specific task, query, or input provided by the end user, which the model addresses within the context defined by the system prompt.

2.3.1 Generating Textual Descriptions from Video

The latest generation of MLLMs is designed with video processing in mind, as reflected in their architectural choices and training procedures. The next step is to determine how to prompt the model to process video data and what structure the textual output should follow.

One option is *dense video captioning*. In this setting, the video is segmented into temporally localized scenes, and each segment is paired with a short textual description. The output is typically a sequence of tuples of the form $(t_{\text{start}}, t_{\text{end}}, \text{caption})$, where each caption describes the visual content occurring within a specific temporal window. This format emphasizes fine-grained coverage and temporal alignment between text and video content [28], and is particularly suitable for long or untrimmed videos containing multiple distinct events.

An alternative is *narrative* or *global-description generation*. Rather than producing temporally segmented captions, the model generates a unified textual summary that describes the video as a whole. The output typically takes the form of a coherent paragraph that captures the main events and overall meaning of the video without explicit temporal boundaries. This work adopts the latter approach. In this work, this is the chosen approach.

The VideoAiiy Framework

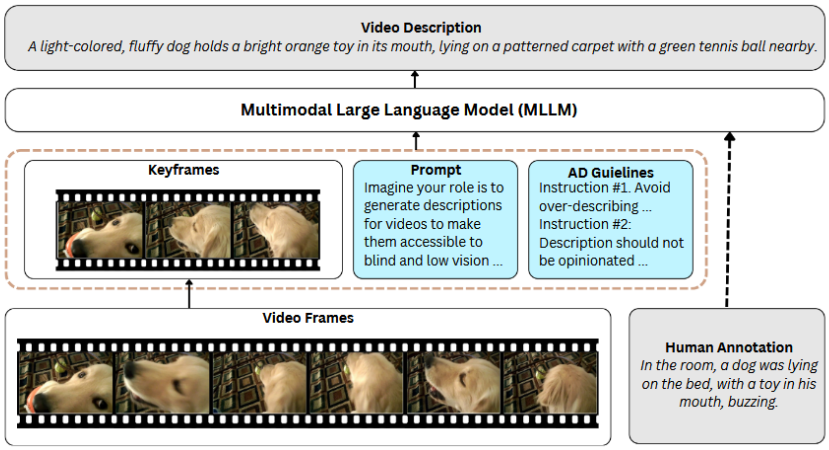

Figure 2.4: Overview of the VideoA11y pipeline. First, keyframes are extracted from the input video. Then, the keyframes, the prompt, AD guidelines, and optional human annotations are provided to the MLLM, which generates accessible video descriptions. Reproduced from Li et al. [29].

In their work, Li et al. [29] propose an automated pipeline for generating video descriptions aimed at improving accessibility for blind and low-vision (BLV) users. Drawing inspiration from the field of professional audio description, the authors construct a prompt by collecting 140 audio-description guidelines and best-practice recommendations from manuals and other authoritative sources. This collection is then curated and refined into a final set of 40 guidelines, which are provided to the MLLM as input alongside the video.

To demonstrate the effectiveness of their method, the authors evaluate the generated descriptions using both standard automatic metrics and custom, accessibility-focused measures, and they conduct extensive human-subject evaluations. Their evaluation involves 347 sighted participants, 40 BLV participants, and seven professional describers. Results from both sighted and BLV evaluations, as well as expert review, indicate that VideoAiiy descriptions outperform novice human annotations and are comparable to trained human descriptions across dimensions such as clarity, accuracy, objectivity, descriptiveness, and user satisfaction.

The paper also reports experiments using a state-of-the-art MLLM (GPT-4V in their study) as the base model, showing that fine-tuning or prompting with the curated accessibility guidelines produces higher-quality descriptions than (i) unrefined novice annotations and (ii) out-of-the-box captions from dedicated vision models. The authors make the code and dataset publicly available to support reproducibility and further research on automated, high-quality accessible descriptions.

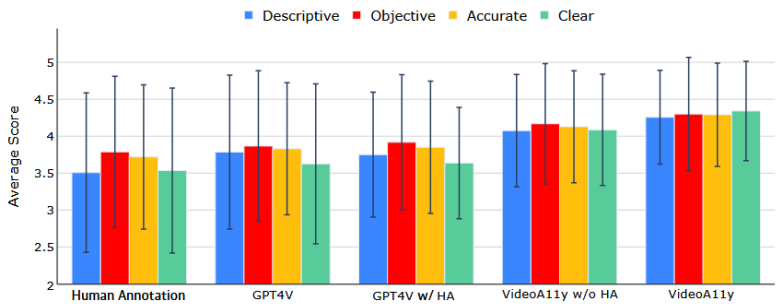

Figure 5: Results of Study 2 with 150 sighted MTurk users. VideoA11y outperforms other methods in all metrics ($p < 0.001$), followed by VideoA11y w/o HA. HA: Human Annotation.

Figure 2.5: Results of Study 1 with 150 sighted MTurk users. VideoA11y (GPT) outperforms all other methods across all evaluation metrics ($p < 0.05$), followed by VideoA11y (GPT) without HA. HA denotes human annotation. Reproduced from Li et al. [29].

2.3.2 Qualitative Content Analysis

An overview of the qualitative coding process

Qualitative coding is a methodological procedure used in psychology to systematically organize and interpret media of any kind: images, audio, text or video [30, 31]. The purpose of qualitative coding is to label content so that specific aspects of interest can be identified and analyzed.

A typical qualitative coding workflow proceeds as follows. First, one or more experts (for example, psychologists) draw on existing literature and theory to operationalize the construct of interest. Operationalization means identifying one or more observable features of the media that are theoretically or empirically linked to the underlying, non-observable behaviour to be measured. Based on this work the experts develop a codebook: a document in which each category is defined and labeling instructions are specified. The codebook is an iterative product and typically undergoes several rounds of refinement and discussion.

Next, the experts create a gold standard by labeling a sample of media content. Experts often label independently at first and then meet to reconcile differences and reach consensus; this joint review both refines the codebook and produces the reference annotations used for later evaluation. Establishing a gold standard is crucial because it defines the expected labels and allows measurement of how reliably other coders reproduce those labels.

Intercoder reliability is the most commonly used metric for this purpose; the most common metric used is Cohen's kappa (κ) [32] to quantify agreement between coders and between coders and the gold standard. A dedicated paragraph below describes the selected reliability metric and reporting conventions.

After the gold standard is established, annotators are recruited and trained. Training sessions introduce annotators to the codebook and provide practice on example items. Annotators are assessed on the test set developed by the experts; their agreement with the gold standard and internal consistency are used to determine whether they are ready to annotate the full dataset. Only after reaching acceptable levels of reliability do annotators proceed to label new media items.

This human coding process is labor intensive: it requires multiple people, extended training and discussion, and limits throughput because a single person can only review a finite amount of content.

Intercoder Reliability and Cohen's Kappa

When multiple human annotators assign labels to the same content, it is necessary to quantify the degree to which their decisions agree. This concept is referred to as *intercoder reliability*. High intercoder reliability indicates that the coding scheme is sufficiently clear and operationalized such that different individuals interpret and apply it consistently. Low reliability, by contrast, may suggest ambiguity in the codebook, insufficient training, or inherent subjectivity in the construct being measured.

A naïve measure of agreement is the percentage of times coders assign the same label. However, simple percentage agreement does not account for agreement that may occur purely by chance. To address this limitation, *Cohen's kappa* (κ) is commonly used. Cohen's kappa measures agreement between two coders while correcting for chance agreement.[32]

Formally, Cohen's kappa is defined as:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{2.9}$$

where P_o is the observed proportion of agreement between coders and P_e is the expected proportion of agreement under chance, computed from the marginal label distributions of each coder.

Cohen's kappa ranges from -1 to 1:

- $\kappa = 1$ indicates perfect agreement,
- $\kappa = 0$ indicates agreement equivalent to chance,
- $\kappa < 0$ indicates systematic disagreement.

Quadratically weighted Cohen's kappa

In many coding scenarios, categories are nominal (i.e., unordered), and all disagreements are treated equally. However, when coding categories are *ordinal* disagreements may differ in severity. For example, confusing adjacent categories is typically less severe than confusing categories at opposite ends of the scale. In such cases, *quadratically weighted Cohen's kappa (QWK)* (κ_w) is more appropriate.

Weighted kappa extends the standard formulation by introducing a weight matrix w_{ij} that penalizes disagreements according to their distance. The general form is:

$$\kappa_w = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}} \tag{2.10}$$

where O_{ij} and E_{ij} denote the observed and expected proportions of items assigned to categories i and j, respectively. Under quadratic weighting, disagreement weights are defined as:

$$w_{ij} = \frac{(i-j)^2}{(k-1)^2} \tag{2.11}$$

where k is the number of categories. This formulation penalizes larger disagreements more heavily than smaller ones, reflecting the ordered structure of the categories.

Like standard kappa, weighted kappa ranges from -1 to 1 and is typically interpreted using descriptive guidelines such as those proposed by Landis and Koch[33]:

$\kappa < 0.00$	Poor agreement
$0.00–0.20$	Slight agreement
$0.21–0.40$	Fair agreement
$0.41–0.60$	Moderate agreement
$0.61–0.80$	Substantial agreement
$0.81–1.00$	Almost perfect agreement

3 Methods

3.1 Dataset

The dataset used in this work was constructed by systematically selecting highly active TikTok influencer profiles and collecting their most engaging posts and associated comments over a one-year period. The procedure involved profile discovery, filtering based on engagement and activity criteria, and automated retrieval of publicly available content and metadata.

3.1.1 Profile Selection

For each target nationality—Italy (IT), United States (US), and South Korea (KR)—TikTok influencer profiles were identified using the *Modash* influencer marketing analytics platform [34], which provides tools for influencer discovery as well as performance metrics such as follower counts and engagement rates.

Initially, profiles with at least 500,000 followers were considered, focusing on *macro influencers* (500K–1M followers) and *mega influencers* (>1M followers). To ensure that only active and engaging accounts were included, profiles were required to have an engagement rate greater than 10%, defined as the ratio between average likes and follower count. This initial filtering identified 463 eligible profiles for Italy, 10,105 for the United States, and 497 for South Korea.

Profiles were ranked according to their number of followers, and only the top 500 profiles per country were retained. Since fewer than 500 eligible profiles were available for Italy and

South Korea, all 463 Italian profiles and all 497 Korean profiles were included, along with the top 500 US profiles.

Commercial and corporate accounts were excluded to focus on individual influencers. Specifically, 8 Italian profiles (primarily sports teams and entertainment brands), 18 US profiles (including media franchises and corporate brands), and 16 Korean profiles (primarily music industry accounts) were removed. Additionally, profiles for which it was not possible to retrieve posts were excluded, resulting in the removal of 6 Italian, 11 US, and 18 Korean profiles.

To ensure sufficient temporal coverage and consistent activity, only profiles with frequent posting behavior during the previous year were retained. The minimum posting frequency thresholds were defined as follows:

- at least one post per week for Italy,
- at least two posts per week for the United States,
- at least one post every two weeks for South Korea.

This filtering resulted in the final selection of 200 Italian profiles, 212 US profiles, and 210 Korean profiles.

3.1.2 Post and Comment Selection

Posts and associated comments were collected using the EnsembleData platform [35], which provides programmatic access to public TikTok content and metadata.

For each selected profile, the 200 most recent posts were initially collected. Since highly active profiles typically publish approximately one post per day, this corresponds to a coverage of approximately six months.

From this initial set, only posts published within the time interval from September 15, 2024, to September 15, 2025, were retained. Furthermore, posts were required to have at least 50 comments or replies to ensure sufficient audience interaction.

Among the remaining posts, the 30 most trending posts per profile were selected based on the number of views (plays), a commonly used proxy for content popularity and audience reach. Posts for which it was not possible to retrieve comments were discarded, resulting in the removal of 28 Italian posts, 58 US posts, and 39 Korean posts.

3.1.3 Final Dataset Composition

The final dataset consists of 5,972 posts from Italian profiles, 6,302 posts from US profiles, and 5,991 posts from Korean profiles.

Upon inspection, 376 posts were found not to contain a video. These entries correspond to text-only posts, image slideshows, or videos that were deleted or made private after the initial data collection. Such entries were excluded from analyses requiring video content, resulting in a final dataset of 17,889 posts with associated videos. For each post, the following information was collected:

- technical metadata (e.g., post identifier and author identifier),
- video caption and associated textual content,
- engagement metrics (e.g., number of views, likes, and comments),
- up to 120 of the most relevant comments per post, depending on availability,
- country of residence of the content creator.

Gender Labels

Creator gender was manually annotated by psychologists from the Department of Psychology. Labels were assigned as *man*, *woman*, or *other*. The *other* category includes cases where gender could not be reliably assigned, such as accounts representing multiple individuals, non-human subjects (e.g., pets), or non-binary identities.The resulting distribution is reported in Table 3.1.

Table 3.1: Distribution of creator gender across countries.

Country	Man	Woman	Other
Italy (IT)	107	82	11
South Korea (KR)	94	90	16
United States (US)	100	89	23

3.1.4 Video Metadata Extraction

After collecting the dataset, additional metadata for each video was extracted using `ffmpeg`. This metadata includes video duration, resolution, and frame rate.

Video Duration

The majority of videos in the dataset are relatively short, which is consistent with the typical format of TikTok content. Notably, videos from Korean profiles tend to be shorter on average compared to those from Italian and US profiles.

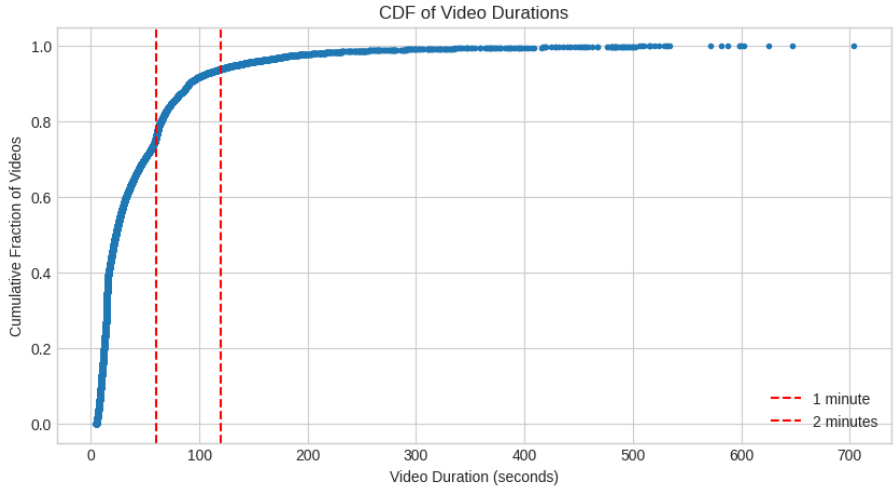

Figure 3.1: Cumulative distribution function (CDF) of video durations in the dataset.

Table 3.2: Summary statistics of video duration (in seconds) per country. Korean videos are shorter on average compared to Italian and US videos.

Country	Count	Mean	Std	Min	25%	50%	75%	Max
IT	5,812	**49.9**	63.6	5.0	13.7	27.3	63.7	703.3
KR	5,843	**28.3**	33.9	3.6	13.1	17.7	29.8	600.0
US	6,234	**52.0**	60.7	5.0	13.8	33.5	66.7	647.5

Resolution

All videos in the dataset follow the standard TikTok vertical format with a 9:16 aspect ratio. Although 170 distinct resolutions are present, the distribution is highly concentrated. The major-

ity of videos are encoded at either 1024×576 (12,159 videos) or 1280×720 (4,907 videos), while all remaining resolutions occur only sporadically and likely reflect minor variations introduced by platform encoding.

Frame Rate

The vast majority of videos in the dataset are recorded at 30 FPS (16,972 videos), which corresponds to the standard frame rate for most consumer and mobile video content. A smaller subset of videos is recorded at 24 FPS (626 videos), reflecting common cinematic or broadcast standards. A small number of videos exhibit other frame rates, ranging from 10 FPS to 60 FPS; however, these cases are rare and likely correspond to non-standard uploads or post-processing variations.

3.2 Content Coding Methodology

This subsection outlines the methodological steps required to perform the content coding task. The objective of this phase was to define a structured annotation framework capable of capturing different visual indicators of sexualization in video content.

3.2.1 Codebook Development

As discussed in Section 2.3.2, the first step in any content analysis procedure is the construction of a codebook that defines the variables to be annotated and the criteria used for evaluation. For this project, the codebook was developed by two psychologists with expertise in media and behavioral analysis.

The design of the codebook was informed by established measures commonly used in the literature on sexualization and media representation [36]. In particular, the original framework was developed for coding sexualization in static images. When adapting this framework to video content, the psychologists explicitly considered the differences between still imagery and dynamic visual media. The definitions and scoring guidelines were therefore adjusted to ensure that the coding scheme remained interpretable and consistent when applied to video data.

The resulting codebook includes multiple categories capturing different aspects of sexualized visual representation, such as clothing coverage, body posture, physical contact, and emphasis on specific body regions. Each category is associated with an ordinal scale representing

increasing levels of visibility, emphasis, or suggestiveness. Table 3.3 provides an overview of the coding categories and their corresponding scales.

Category	Scale	Description
Clothing Nudity	0–5	Degree of clothing coverage, ranging from fully clothed to complete nudity, including intermediate levels where garments reveal or emphasize body regions.
Touch	0–3	Presence and type of physical contact involving the individual, from no touch to deliberate contact involving anatomically intimate regions.
Pose	0–3	Body posture or positioning, ranging from neutral everyday posture to poses intentionally emphasizing or exposing intimate body regions.
Mouth Expression	0–2	Degree of suggestiveness in oral expression, from a neutral mouth position to more explicit emphasis (e.g., open mouth or tongue display).
Makeup Presence	0–2	Visibility and intensity of cosmetic application, ranging from none to strongly visible makeup.
Upper Torso Focus	0–2	Visual emphasis placed on the upper torso region.
Genital Region Emphasis	0–2	Degree to which the genital region is visually emphasized.
Gluteal Region Emphasis	0–2	Degree to which the gluteal region is visually emphasized.
Body Size	0–2	Apparent body mass of the individual, ranging from low to high body mass.
Musculature	0–2	Visible level of muscle definition.

Table 3.3: Summary of the labeling schema, detailing the visual dimensions evaluated and their associated ordinal scales.

3.2.2 Creation of the Video Sample Dataset

To reliably evaluate the quality of the annotations produced by the MLLM, it was necessary to establish a gold-standard annotation set. To this end, a subset of 50 videos was randomly sampled from the full dataset and manually annotated by two psychologists affiliated with the Department of Psychology at the University of Padova.

The annotation process followed a two-stage procedure commonly used in content analysis studies. First, the psychologists independently annotated each video according to the coding scheme defined in the codebook. Subsequently, the annotators compared their annotations and discussed any disagreements in order to reach a consensus.

The final consensus annotations constitute the gold-standard dataset used to evaluate the model's performance. In cases where multiple levels of a given feature were present within a video, the annotation was defined as the maximum value observed across the video.

3.2.3 Variable Selection and Rationale

Although the codebook includes multiple categories, this study focuses on three variables: Clothing Nudity, Pose, and Touch. This design choice was motivated by several considerations.

First, these variables capture a substantial portion of the visual information relevant to sexualization in media content. Second, preliminary experiments conducted across all available categories revealed that model performance varied considerably depending on the variable being predicted. Some categories proved difficult for the model to identify reliably. In many cases, this difficulty likely stems from the inherent ambiguity of the category definitions. For example, variables such as genital focus or mouth expression, while commonly used in the literature, can be challenging to operationalize consistently. Even for human annotators, assigning a label in these cases may require subjective interpretation.

By contrast, the selected variables typically present clearer visual distinctions between categories. The boundaries between scale levels are more explicit, making them more suitable for automated inference.

Finally, prior work has shown that applying qualitative coding frameworks with MLLMs often requires iterative codebook refinement to achieve reliable results [37]. Focusing on a smaller set of well-defined variables also makes the experimental setup more manageable while still capturing the primary aspects of sexualized representation in video content.

3.2.4 Establishing a Baseline: Measurement of Agreement

As discussed in Section 2.3.2, QWK is the preferred metric for evaluating agreement between the annotations produced by the model and a reference standard. In this study, QWK is computed between the annotations generated by Q3VL and the gold-standard annotations in the Video Sample Dataset.

To contextualize the model's performance, it is important to define a benchmark. For this purpose, we refer to the results reported by Hatton et al. [36], where similar coding scales were applied to measure sexualization in magazine covers. The authors report the following inter-coder reliability scores:

- **Pose**: $\kappa = 0.831, p < .001$

- **Touch**: $\kappa = 0.726, p < .001$
- **Clothing/Nudity**: $\kappa = 0.891, p < .001$

These values provide a reference for human performance on a comparable task and serve as a benchmark against which the model's performance can be evaluated.

Comparable Studies A relevant comparison can be drawn from the work of Liu et al. [38], who explored the use of MLLMs for coding keyframes extracted from short videos related to depression. Their findings highlight an important distinction:

1. Models achieve high agreement ($\widetilde{\text{QWK}} \approx 0.9$ or higher) when coding objective and clearly defined features, such as the presence of food or beverages or whether an individual is crying.
2. Performance decreases substantially when the task requires interpretation or subjective judgment, such as assessing the emotional valence of a scene ($\widetilde{\text{QWK}} \approx 0.3$).

These findings support the decision to focus on variables that are visually explicit and less reliant on subjective interpretation.

3.3 Technical Implementation

This section describes the technical infrastructure and implementation details underlying the proposed approach. It focuses on the practical components required to process the data and run the model, and is organized into three main parts. The first part outlines the data processing pipeline used to transform raw videos into model-compatible inputs. The second part describes the hardware infrastructure, including the computational resources and cluster architecture. The third part presents the software stack, detailing the frameworks, libraries, and tools used for model development and inference.

3.3.1 Data Processing

Videos in the dataset are stored in `.mp4` format. In order to process them with Q3VL, they must be converted into a sequence of visual tokens compatible with the model's multimodal transformer architecture. To perform this transformation, the model authors provide a set of utility functions that handle video decoding, frame sampling, and spatial preprocessing.[39]

This section summarizes the key steps involved in this pipeline and highlights the parameters that influence both the computational cost and the characteristics of the input provided to the model.

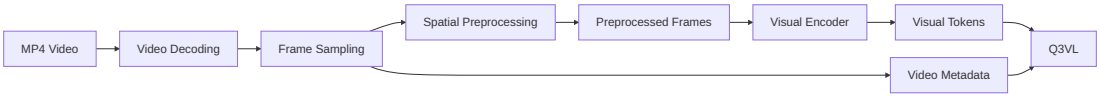

Figure 3.2: Overview of the video preprocessing pipeline used in this work.

Figure 3.2 provides an overview of the main stages in the preprocessing pipeline.

Video Decoding and Frame Sampling

The preprocessing pipeline begins with video decoding. Each video is opened using the *torchcodec* backend, which reads the video stream and provides access to individual frames, along with metadata such as the original frame rate and the total number of frames.

Following decoding, the next stage is **frame sampling**. In this phase, a subset of frames is extracted according to a predefined sampling strategy. The selection process is governed by two parameters: the maximum number of frames (`max_frames`) and the target frame rate (`fps`), which together determine the temporal resolution and coverage of the sampled sequence.

The number of frames to be sampled is computed based on the duration of the video. If this number is lower than the specified maximum, the selected frames are uniformly distributed across the temporal interval. Conversely, if the required number of frames exceeds the maximum allowed, the pipeline samples at a constant rate and adjusts the effective frame rate accordingly.

Spatial Preprocessing and Tokenization

After frame sampling, each frame undergoes **spatial preprocessing**. This stage determines whether resizing is required and, if so, to what extent. Figure 3.3 provides a schematic overview of this process.

The procedure consists of two main steps. First, a target resolution is selected, either dynamically or according to a user-specified value. In the dynamic case, the frame resolution is evaluated against a `max_pixels` threshold. This parameter acts as a proxy for the number of visual tokens, which scales with pixel density. Imposing a limit on the total number of pixels prevents the generation of an excessive number of visual tokens, which could otherwise lead

to OOM errors during inference. If the total pixel count of the selected frames is below this threshold, the frames remain unchanged; otherwise, they are resized to satisfy the constraint.

In the alternative mode, all videos are uniformly resized (either upscaled or downscaled) to a fixed target resolution specified by the user.

In both cases, prior to applying the target resolution, a validation step ensures that the height and width of each frame are divisible by a fixed factor determined by the model's patch configuration. This constraint guarantees that the visual encoder can correctly partition the image into patches. For the architecture employed in this work, this factor results from a patch size of 16 pixels and a spatial merge factor of 2, requiring input dimensions to be multiples of 32 pixels. Whenever resizing is required, it is performed using *bicubic interpolation*.

Finally, once temporal and spatial preprocessing are completed, the resulting sequence of frames is converted into tensor representations and passed to the model's visual encoder. The encoder partitions each frame into patches and transforms them into visual tokens, which constitute the input sequence processed by Q3VL.

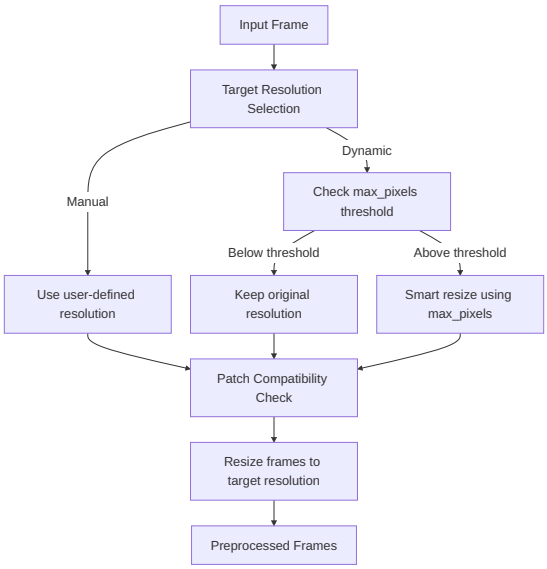

Figure 3.3: Overview of the spatial preprocessing stage

Preprocessing outputs and metadata

During preprocessing, the pipeline also generates a dictionary containing metadata about the processed video, such as the original frame rate, the indices of the sampled frames, and the total

number of frames. This metadata is preserved and later incorporated into the textual input provided to the model, allowing the system to retain information about the temporal structure of the original video.

3.3.2 Model deployment and technical implementation

All inference experiments in this work were conducted using the High-Performance Computing (HPC) facilities provided by the University of Padova, specifically through the cluster managed by the Dipartimento di Ingegneria dell'Informazione (DEI).[40]

HPC

HPC refers to the use of interconnected computing resources to achieve computational performance far beyond that of standard desktop computers or workstations. In the context of this project, the DEI cluster allows to use a much bigger model and faster inference thanks to parallelism.

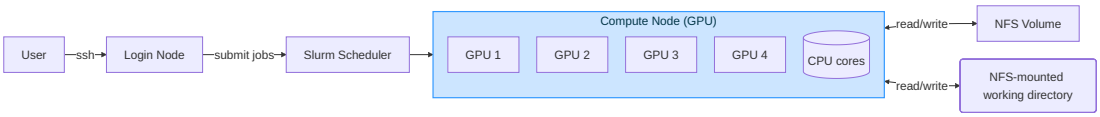

Figure 3.4: Illustrative overview of the HPC cluster showing login nodes, the Slurm workload scheduler, compute nodes, and shared NFS storage.Note that this is as example, as some nodes provide more than 4 GPUs

Cluster Architecture The DEI HPC cluster follows a standard multi-node architecture consisting of the following components:

- **Login Node**: Serves as the entry point for users to access the cluster via SSH. Users connect to this node to prepare jobs, manage files, and submit computational tasks.
- **SLURM Scheduler**: The Simple Linux Utility for Resource Management (SLURM) is a workload manager responsible for job scheduling, resource allocation, and queue management. Users submit jobs to SLURM, which allocates compute resources based on availability and priority.
- **Compute Nodes**: The cluster includes both CPU-only nodes and GPU-accelerated nodes, with the latter being central to this project. Some GPU nodes are equipped with high-end NVIDIA GPUs, such as the A40 and L40S, each providing up to 48 GB of VRAM [40].

- **Network File System (NFS)**: The Network File System (NFS) is a distributed file system protocol that enables access to files over a network as if they were stored locally. All nodes mount a shared NFS volume, ensuring consistent access to data, models, and code across the cluster.

3.3.3 Software Stack and Libraries

To ensure computational reproducibility and compatibility with the HPC infrastructure at DEI, the software environment was managed using *Singularity*. Singularity is specifically designed for shared HPC environments, allowing users to run containerized applications securely without requiring elevated permissions, while maintaining access to high-speed interconnects and GPU drivers provided by the host system.

For this work, the base software environment was derived from the official Docker image provided by the Q3VL development team [41, 39]. To deploy this environment on the cluster, the Docker image was pulled and converted into a Singularity Image Format (SIF) file using the `singularity build` command.

While the base image provided most of the required dependencies, the `flash-attn` library needed to be installed separately to enable efficient attention computation within the Transformers framework. Consequently, `flash-attn` was installed directly from its source repository [42] inside the Singularity container prior to execution.

A complete list of all Python packages, including their specific versions, is provided in **Appendix A** to facilitate exact replication of the experimental environment.

Implementation Framework

The model loading and inference pipeline was implemented using the *Transformers* library developed by Hugging Face [43]. While Transformers provides a flexible and user-friendly interface, it is not always the most performant solution for running large models. Several specialized inference frameworks, such as *llama.cpp* and *sglang*, extend the Transformers ecosystem by introducing additional optimizations aimed at improving inference speed and memory efficiency across both high-end and resource-constrained hardware.

Despite these alternatives, the Hugging Face Transformers library was selected for this work for three primary reasons:

1. **Research Focus:** The primary objective of this project is to evaluate the semantic capabilities of Q3VL in a controlled setting, rather than to maximize inference speed.
2. **Video Support:** Support for video inputs in alternative frameworks remains limited and relatively recent. For example, *llama.cpp* does not natively support video inputs for multimodal models and is currently restricted to text and image modalities [44].
3. **Reproducibility:** The authors of Q3VL provide official reference implementations exclusively using the Transformers library [39]. Adhering to the official framework minimizes the risk of implementation artifacts and ensures consistency with the intended model behavior.

Inference Optimization Strategies

- **Model Quantization**

  Model quantization refers to a set of techniques used to reduce the numerical precision of model weights in order to decrease memory consumption and accelerate inference. Modern large language models are typically stored using 16-bit floating-point formats (such as `bf16`). Quantization reduces this precision to lower-bit representations while attempting to preserve predictive performance.

  In this work, the model weights are quantized from `bf16` to 8-bit precision using the *bitsandbytes* library. This library implements optimized low-precision matrix multiplication algorithms that enable efficient inference while maintaining minimal performance degradation [45].

- **Flash Attention**

  Flash Attention is an optimized attention computation algorithm designed to reduce the memory footprint and computational cost of the self-attention mechanism in transformer models. By restructuring the attention computation to better utilize GPU memory bandwidth and minimize unnecessary memory operations, Flash Attention significantly improves both memory efficiency and execution speed.

  In this project, the Flash Attention 2.0 algorithm was used through the *flash-attn* library [42]. This implementation provides highly optimized GPU kernels for attention operations, enabling faster inference while preserving the numerical equivalence of standard attention computations.

- **Accelerate**

  To fully utilize the multiple GPUs available on the cluster, the *Accelerate* library was employed. This library provides utilities for distributing model inference across multiple

devices while abstracting much of the complexity associated with parallel and distributed computation.

In this work, a full copy of the model was loaded on each GPU assigned to a job. The dataset was then partitioned so that each GPU processed a distinct subset of the input videos in parallel.

- **Greedy Decoding**

  During inference, responses were generated using greedy decoding. In this strategy, the model selects, at each step, the token with the highest predicted probability.

  Greedy decoding was chosen primarily for reproducibility and efficiency. Because token selection is deterministic, identical inputs always produce identical outputs. This property is particularly desirable in an experimental setting, where consistency across runs is essential.

  Additionally, greedy decoding is computationally efficient: at each generation step, the model computes the probability distribution over the vocabulary and selects the most likely token. In contrast, stochastic decoding strategies require sampling from this distribution, introducing randomness and additional computational overhead. Greedy decoding therefore provides a fast and deterministic generation procedure suitable for large-scale evaluation.

Cluster Execution Pipeline

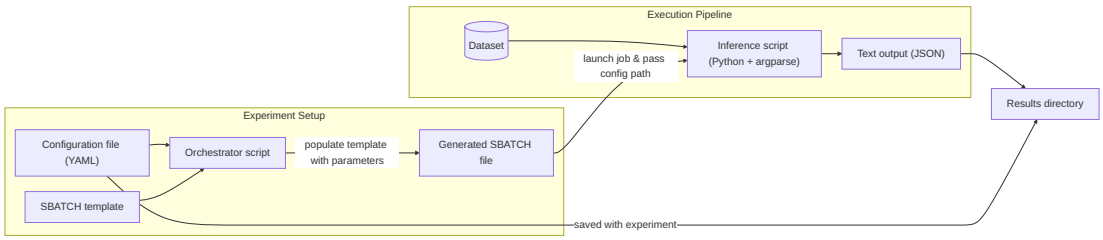

Figure 3.5: Workflow diagram illustrating the automated experiment pipeline from configuration to results.

Figure 3.5 illustrates the execution pipeline used to run inference experiments on the DEI HPC cluster. The pipeline was designed to automate experiment configuration, job submission, and result collection, while efficiently utilizing the available computational resources.

The process begins with the definition of an experiment configuration file written in YAML format. This file specifies all relevant parameters, including model settings, prompts, input/output

paths, and runtime options. The configuration file is stored together with the experiment outputs to ensure reproducibility.

An orchestrator script then reads the configuration and populates a predefined SBATCH template, generating a SLURM-compatible job submission script. The resulting SBATCH file specifies the required computational resources, such as GPUs, CPU cores, memory allocation, and maximum execution time.

Once submitted, the SLURM scheduler allocates the requested resources and launches the job on the cluster. The SBATCH script then executes the inference pipeline, during which Q3VL processes the input videos.

Finally, the model outputs are saved as JSON files and stored together with the corresponding configuration file. This structure ensures that each experiment is fully documented and reproducible, while also simplifying experiment management on the cluster.

All source code related to the pipeline is available in the project GitHub repository [46].

4

Experimental Evaluation and Parameter Selection

This chapter details the experiments performed during this study. The experiments are divided into two sets based on their specific optimization objectives.

4.1 Experiments

4.1.1 Experimental Setup and Metrics

As outlined in Section 3.3.3, a unique configuration file can be used to modify both the prompt and the parameters of the preprocessing pipeline. This system enables the creation of distinct experimental conditions, allowing model performance to be evaluated under a variety of controlled settings.

To assess performance, the model is tasked with producing annotations for the videos in the Video Sample dataset described in Section 3.2.2. These automatically generated annotations are then compared with the manually produced gold-standard annotations using QWK as the primary evaluation metric.

In this work, two main groups of experiments were conducted, each addressing a different aspect of the modeling process.

4.1.2 Validation of Model Scale and Reasoning Strategies

The first group of experiments was conducted following the work of Dunivin et al. [37], who investigate the use of LLMs for content coding tasks on textual data. Their study reports several relevant findings regarding model behavior in this setting. In particular:

- larger models yield better task performance;
- the use of CoT reasoning improves model performance;
- increasing the number of attributes to be labeled decreases model performance.

In this work, similar experiments are conducted to evaluate whether these observations generalize to a multimodal setting involving video data.

Impact of Model Scale

To investigate the impact of model scale, the performance of the Q3VL-Instruct model is compared across four different model sizes. Figure 4.1 illustrates the results. Performance increases

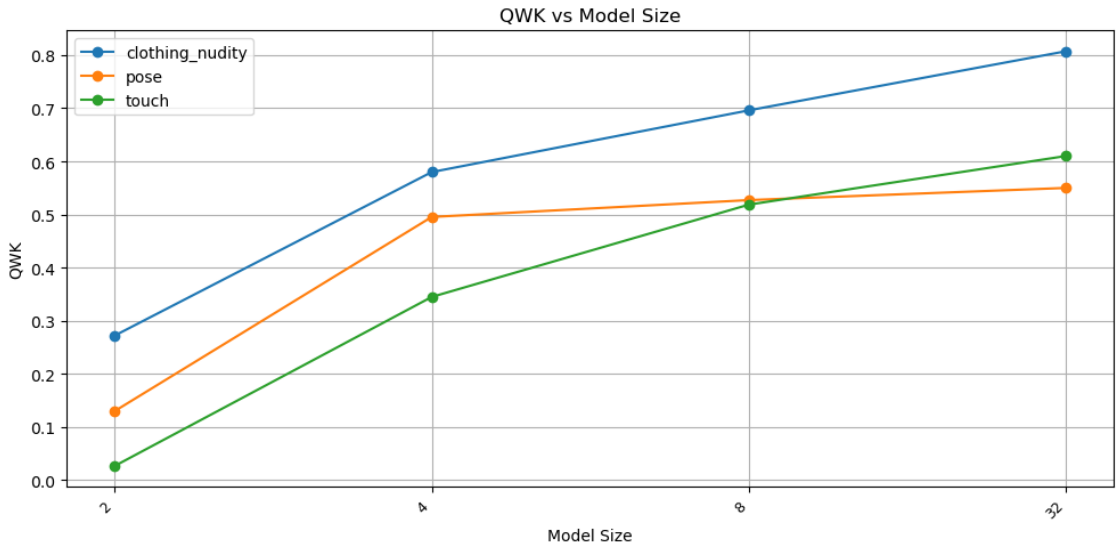

Figure 4.1: Q3VL model performance across different parameter sizes (in billions).

significantly with model size, confirming the findings of Dunivin et al. The 32B model outperforms the others across all categories and is therefore used for the subsequent experiments, serving as the model of choice for this work.

Evaluating Thinking vs. Instruct Variants

This experiment compares the performance of the Instruct and Thinking versions of the Q3VL 32B model. As described in Section 2.2.5, the Thinking variant is specifically designed for CoT reasoning and outputs its reasoning process in the text stream before producing the final answer.

Table 4.1 presents the performance of both models using Accuracy and QWK.

Task	**Instruct**		**Thinking**	
	Accuracy	QWK	Accuracy	QWK
Clothing / Nudity	0.62	0.807	0.40	0.628
Pose	0.50	0.550	0.48	0.339
Touch	0.64	0.610	0.28	0.31

Table 4.1: Performance comparison between the Instruct and Thinking variants across different tasks. Accuracy and QWK are reported.

The Instruct variant consistently outperforms the Thinking variant across all evaluated metrics, while also requiring fewer computational resources. This result suggests that, in the present setting, explicitly generating intermediate reasoning steps does not provide a performance advantage.

However, this finding should be interpreted with caution. First, the comparison is limited to a specific model family and implementation of CoT reasoning. Second, the effectiveness of CoT methods is known to be task-dependent, with greater benefits typically observed in tasks requiring complex reasoning or multi-step inference [37]. In contrast, the content coding task considered in this work may rely more heavily on direct visual pattern recognition than on explicit reasoning processes.

Based on these results, the Instruct variant was selected for all subsequent experiments.

Impact of the Number of Annotated Variables

This experiment examines how the number of variables included in the annotation task affects model performance. To this end, the prompt was systematically modified to vary the annotation load. The experiment was initially conducted using the three core variables considered in this study (Touch, Pose, and Clothing/Nudity). Additional variables from the full codebook were then progressively introduced, increasing the total number of attributes the model was required to annotate.

Performance was evaluated by measuring the QWK scores for the three primary variables as the total number of annotated variables increased. The results are presented in Figure 4.2. Contrary to expectations based on prior work [37], performance remains relatively stable as the number of variables increases. This suggests that, in the present multimodal setting, the model is able to handle additional annotation tasks without a substantial degradation in performance on the primary variables.

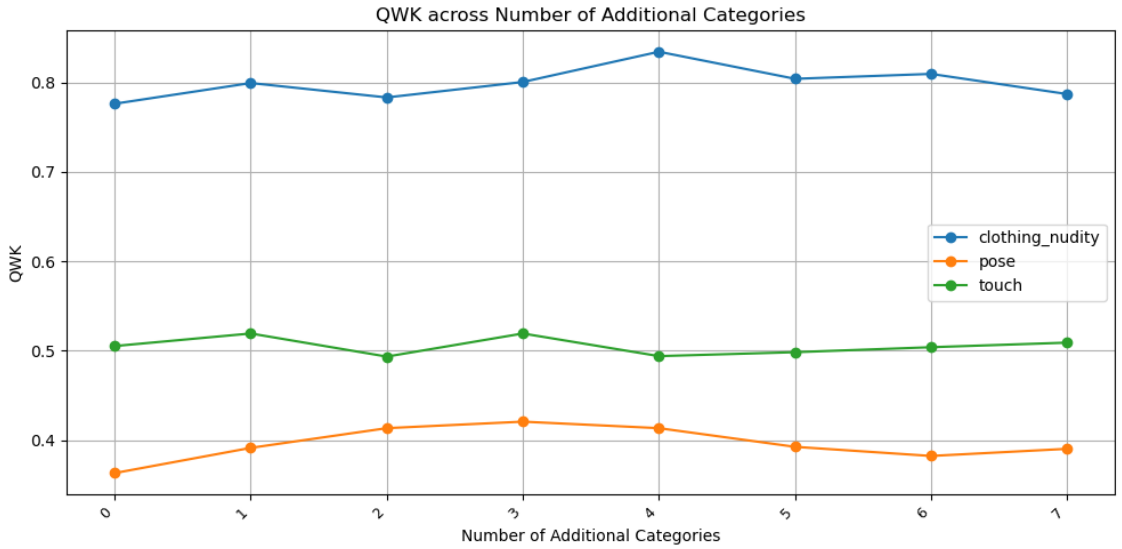

Figure 4.2: Performance changes measured with QWK as the number of variables to code increases.

4.1.3 Impact of Context Length and Input Configuration

A second set of experiments was conducted to investigate the relationship between context length and model performance in multimodal video analysis tasks. Prior work suggests that increasing the input context length in large language models can negatively affect task accuracy, even when retrieval capabilities remain largely unaffected. In particular, Hong et al. [47] systematically evaluated 18 state-of-the-art LLMs, including models from the Qwen family, and observed consistent performance degradation as the number of input tokens increases. This phenomenon, often referred to as *context rot*, indicates that model attention is not uniformly distributed across long sequences, with reduced reliability for tokens located deeper within the context window, regardless of model architecture or scale.

This issue is directly relevant to the content coding task addressed in this study. Visual frames are tokenized and concatenated with textual instructions within a shared context win-

dow, requiring the model to attend simultaneously to fine-grained visual details within individual frames while preserving temporal coherence across the sequence. The total number of visual tokens generated depends on two controllable parameters:

- The spatial resolution of input frames, which determines the level of visual detail available for recognizing subtle features;
- The number of sampled frames, which controls the temporal coverage of the video.

Together, these parameters define the effective context length presented to the model. As a result, experimental design involves a fundamental trade-off: higher resolution and denser frame sampling improve the richness of the input representation, but they also increase context length, potentially amplifying performance degradation associated with context rot.

The experiments presented in this section aim to identify optimal configurations for these parameters and to quantify their impact on overall model performance.

Resolution Analysis

To ensure reproducibility and feasibility in the subsequent experiments, a set of fixed input parameters was defined. First, the maximum number of frames per video was constrained to regulate the temporal context length. Based on the video length distribution reported in Section **??**, a baseline target of 60 frames was selected. This choice guarantees a minimum sampling rate of 2 frames per second for videos up to 30 seconds in duration, covering approximately 75% of the dataset across all regions. For longer sequences, this configuration maintains a sampling density of at least 1 frame per second for the majority of videos.

Next, an upper bound for spatial resolution was established. The highest candidate considered was 1280×720, corresponding to the maximum available resolution and allowing all data to be processed at native quality. However, preliminary experiments using the optimization framework and cluster infrastructure described in Section 3.3.2 showed that processing 60 frames at this resolution exceeds available memory. In particular, the KV cache requirements surpass the capacity of the largest GPUs in the cluster, leading to out-of-memory (OOM) errors.

For this reason, 1024×576 was selected as the maximum feasible resolution. This setting corresponds to the second-highest resolution in the dataset and is also among the most frequently occurring. Under this configuration, the maximum number of frames that can be processed without OOM errors was found to be approximately 90 per video, and this value was adopted as the final setting.

Finally, a systematic procedure was applied to derive a controlled set of lower resolutions for comparative evaluation. This process was subject to several constraints: (i) both height and width had to be multiples of 32 pixels, as required by the Q3VL pipeline (see Section 3.3.1); (ii) resolutions had to satisfy the upper bound ($H \leq 1024$, $W \leq 576$); and (iii) the original aspect ratio had to be preserved to avoid geometric distortion.

Since only one configuration strictly satisfied all three constraints, the aspect ratio requirement was relaxed. The search space was extended to include configurations with an aspect ratio within a narrow 5% interval [0.54, 0.585], centered around the dataset's nominal portrait ratio (9:16 $\approx$ 0.5625). This relaxation limited geometric distortion while increasing the number of feasible configurations.

An exhaustive enumeration under these constraints yielded 23 valid resolutions. From this set, ten configurations were selected through uniform sampling with respect to total pixel count ($H \times W$), ensuring a balanced coverage of the available visual token budget while avoiding redundancy among computationally similar settings. The final set of resolutions, ordered by increasing pixel count, is reported in Table 4.2. Figure 4.3 presents the results of the resolution

Resolution ($H \times W$)	Pixel Count
224×128	28,672
384×224	86,016
512×288	147,456
608×352	214,016
704×384	270,336
768×448	344,064
832×480	399,360
896×512	458,752
960×544	522,240
1024×576	589,824

Table 4.2: Selected resolutions ordered by increasing pixel count.

analysis. The model's performance on the Pose and Clothing/Nudity variables remains stable across different runs and resolution settings. In contrast, performance on the Touch variable increases with higher input resolution.

These results indicate that resolution has a measurable impact on the evaluation of the Touch variable, while its effect on the other variables is limited within the tested range.

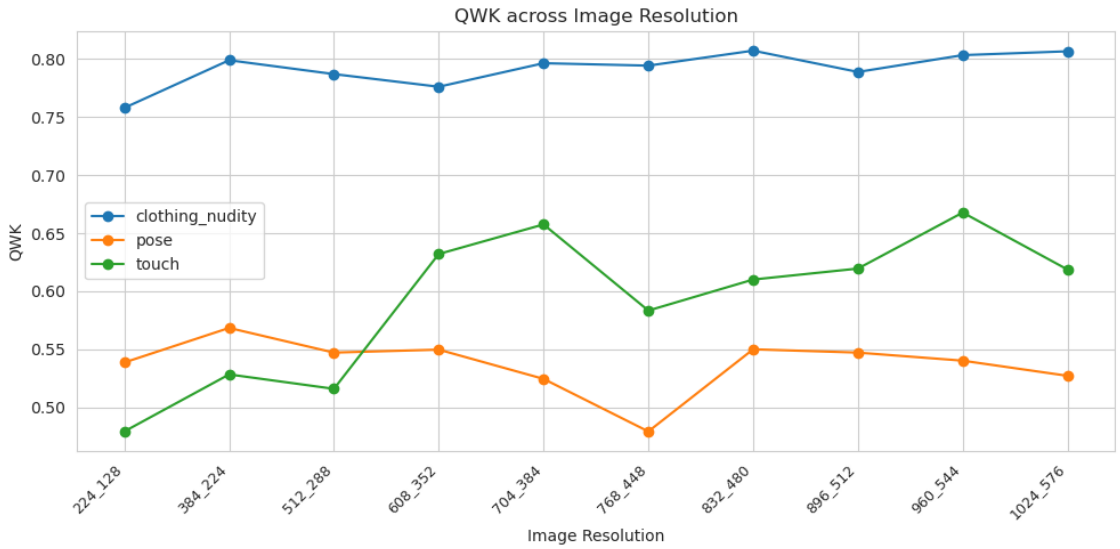

Figure 4.3: performance changes measured with QWK as resolution changes.

Temporal Sampling Density

This experiment was conducted by first fixing the input resolution based on the results reported in Section 4.3. A resolution of 608×352 was selected, as it achieved strong performance while remaining sufficiently compact to allow a larger number of frames to be sampled per video without exceeding memory constraints.

The next step was to evaluate the effect of increasing the number of frames per video. One possible approach would have been to vary the FPS parameter and progressively increase the sampling rate. However, this strategy was not adopted, as it would have limited the range of feasible configurations: even for relatively short videos, high FPS values would lead to a rapid increase in the number of frames, resulting in OOM errors.

Instead, a direct sampling approach was employed. For each video, an increasing number of frames was selected and provided as input to the Q3VL model, ranging from 10 to 180 frames. The results are shown in Figure 4.4.

The performance on the Clothing/Nudity variable remains stable across different numbers of sampled frames. In contrast, performance on the Touch and Pose variables increases with a higher number of frames. This indicates that these variables benefit from greater temporal coverage, as they depend on dynamic information that may not be fully captured with a limited number of sampled frames.

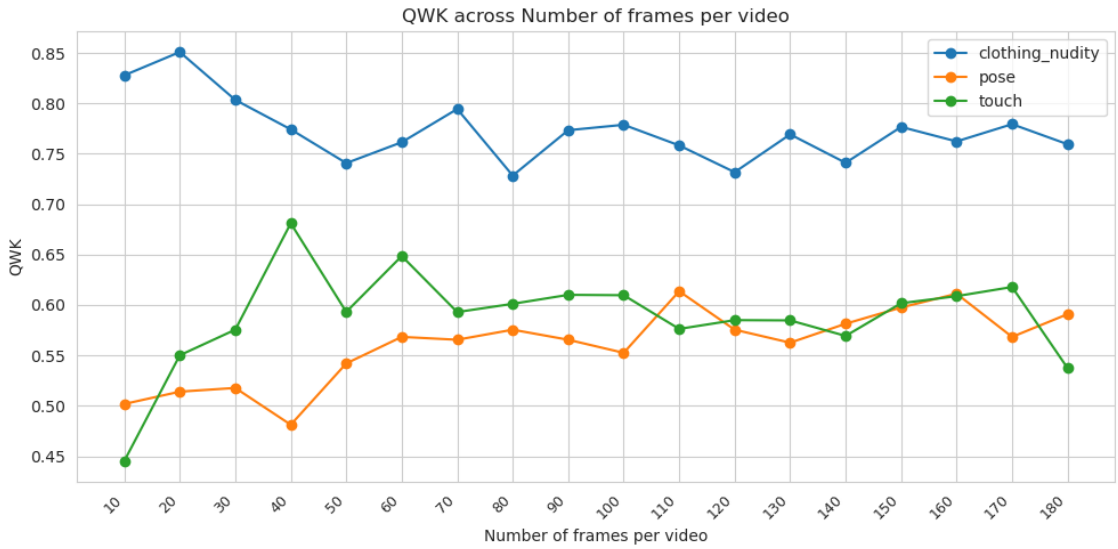

Figure 4.4: Model performance as a function of the number of frames sampled per video.

4.2 Prompt Optimization and Final Configuration Analysis

4.2.1 Prompt Tuning

In parallel with parameter tuning, optimization efforts were directed towards the prompt structure. Prompt engineering presents inherent challenges, particularly with respect to stability and determinism. Large language models often exhibit sensitivity to lexical variations, where minor changes in phrasing can lead to substantial differences in the generated output, even when the underlying semantic intent remains unchanged [48]. This sensitivity complicates the assessment of performance improvements.

To mitigate this issue, experimental conditions were isolated where possible, and an iterative refinement process was adopted. In this process, incremental modifications were applied to the prompt, and their impact on model performance was evaluated.

The overall prompting strategy remained consistent throughout development. The model is provided with the annotation codebook and the input video frames, and is instructed to produce labels in a structured JSON format. This structured output facilitates direct downstream processing of the model's predictions. The initial prompt version introduced the codebook and defined basic output constraints. Subsequent analysis identified several areas for improvement

to better align the model's outputs with the annotation guidelines.

The transition from the baseline prompt to the final configuration involved three main modifications, each contributing to performance improvements:

- **Handbook Terminology Standardization:** The terminology used in the codebook was revised from informal descriptors to more formal and precise language. For example, references to *private areas* were replaced with *anatomically intimate regions*, and general references to body parts were refined using more specific anatomical terms (e.g., *anterior thoracic area* instead of *chest*).

- **Inclusion of Step-by-Step Reasoning Examples:** Incorporating an example of the reasoning process required for the content coding task led to improved performance under a step-by-step reasoning setting. This approach encourages the model to first map visual observations to the annotation criteria before producing the final label.

- **System Prompt Constraint Consolidation:** As described in Section 2.3, Q3VL allows separate customization of system and user prompts. Performance improved when this structure was explicitly leveraged: general instructions and the model persona were defined in the *System Prompt*, while variable inputs, such as the codebook and video frames, were provided in the *User Prompt*.

Table 4.3 reports the performance difference between the initial and final prompt configurations. The full specification of the final prompt is provided in Appendix B B.

	Clothing_Nudity	Pose	Touch
Before Prompt Tuning	0.741	0.318	0.361
After Prompt Tuning	0.787	0.598	0.626

Table 4.3: Improvement in QWK across categories before and after prompt tuning

4.2.2 Final Parameter Selection

Several factors informed the selection of parameters for the full dataset run. First, a decision was made regarding prompt structure. While highly specialized, category-specific prompts could potentially maximize performance for individual variables, this approach would substantially increase computational cost. For this reason, a unified prompt structure was retained to ensure scalability and feasibility.

Second, during the ablation experiments, certain configurations exhibited improved performance in specific categories while underperforming in others. As a result, the final configuration was selected to achieve a balance across all three target categories, prioritizing consistent performance rather than optimizing for a single metric.

Following the optimization experiments conducted on the Video Sample dataset, the selected configuration was applied to generate annotations for the full dataset. Although the resulting annotations are subject to the error margins reflected in the QWK scores, they provide a reliable signal for downstream analysis. The parameters used for video processing—including both annotation generation and content coding—consisted of a spatial resolution of 832×480, a maximum sampling rate of 2 frames per second (with adaptive reduction for longer sequences), and an upper limit of 90 frames per video.

It should be noted that no independent validation was performed to assess the quality of the video descriptions generated using the methodology outlined in Section 2.3.1. Instead, this work relies on the quality assurance procedures reported in the original Video Ally study. Although those evaluations were conducted using a different MLLM, the overall framework remains consistent with the present implementation. This reliance on external validation represents a limitation of the current work, as model-specific biases in the video processing pipeline were not explicitly isolated. Both the system and user prompts used within the Video Ally framework are reported in Appendix B.

4.2.3 Performance Evaluation of Final Annotations

The performance of the final model configuration is evaluated using confusion matrices, which illustrate the correspondence between predicted and ground-truth labels. Off-diagonal elements highlight systematic disagreement patterns, including consistent over- or under-estimation of intensity levels, as well as confusion between semantically related categories. Figure 4.5 presents the confusion matrices for the three target variables after prompt optimization.

A strong diagonal structure is observed across all categories, indicating good overall agreement between model predictions and ground-truth labels. At the same time, a slight tendency toward underestimation is visible, along with a limited number of misclassifications.

As discussed in Section 3.2.4, a useful reference point is provided by the performance of the codebook creators on an image annotation task. Table 4.4 compares the final model performance against human inter-rater reliability. While the model consistently falls short of the human baseline across all categories, the achieved QWK scores remain within an acceptable

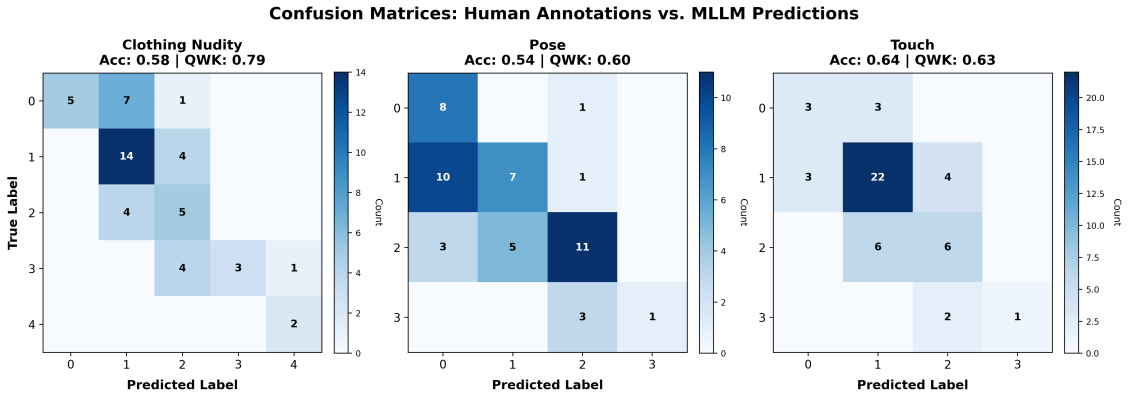

Figure 4.5: Confusion matrices showing the agreement between model predictions and human annotations for the three content coding categories. Rows correspond to ground-truth labels, while columns correspond to model predictions.

range for automated coding tasks. These results indicate that, although the model does not reach human-level agreement, it produces annotations that are sufficiently reliable for large-scale analysis.

Category	Model QWK	Human Baseline QWK	Absolute Gap ($\Delta\kappa$)
Clothing/Nudity	0.787	0.891	0.104
Pose	0.601	0.831	0.230
Touch	0.626	0.726	0.100

Table 4.4: Comparison of model performance against human inter-rater reliability. The human baseline reflects agreement among codebook creators on a comparable annotation task.

5 Results and Analysis

5.1 Runtime Statistics and Computational Configuration

The final prompts and parameter settings were used to perform inference over all videos in the dataset. All computations were executed on HPC nodes equipped with NVIDIA A40 GPUs. Each processing job was allocated six GPUs and two CPUs per GPU (twelve CPUs per job), operating on a single dedicated node.

The dataset was partitioned by national origin (IT, KR, US). For each national subset, two distinct processing runs were conducted to perform content coding and video descriptions generation.

Table 5.1 summarizes the computational metrics across all national subsets.

Dataset	Size (GB)	Content Coding Time	Video Captions Time
IT	40	05:56:36	19:16:07
KR	26	05:03:48	16:57:37
US	44	06:37:40	21:10:11

Table 5.1: Runtime summary of annotation and video captioning jobs across dataset splits.

5.2 Results analysis

5.2.1 Content Coding Analysis

Figure 5.1 presents the distribution of scores across the three content coding variables (touch, clothing_nudity, and pose) as proportions of the total dataset. The distributions exhibit right-skewed patterns across all variables, with the majority of videos receiving low scores (0-1) and progressively fewer videos assigned higher scores. This pattern aligns with the dataset's construction methodology: videos were collected based on platform popularity metrics without explicit targeting of sexualized content.

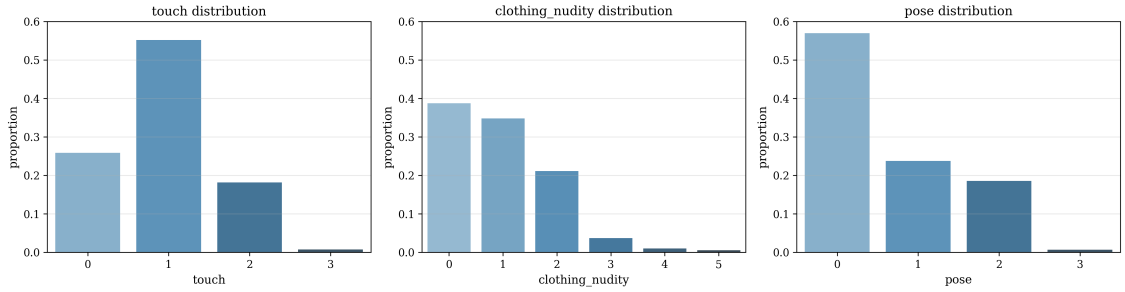

Figure 5.1: Distribution of coded variables.

5.2.2 Gender and Nationality Effects on Coded Variables

To enhance interpretability, scores were averaged across coded variables for each creator (i.e., across their videos), and the resulting aggregated values were treated as continuous measures for statistical analysis. A two-way analysis of variance (ANOVA) was conducted to assess the effects of creator gender (restricting the analysis to creators labeled as man or woman) and national origin (IT, KR, US) on sexualization dimensions. Interaction effects were included to evaluate whether gender differences vary across national contexts.

Figure 5.2 illustrates the mean scores across conditions, while Table 5.2 reports the corresponding statistical results. The F-statistic represents the ratio of systematic (between-group) variance to residual (within-group) variance, with larger values indicating stronger evidence against the null hypothesis. Partial eta-squared (η_p^2) quantifies the proportion of variance explained by each factor, with conventional thresholds of 0.01 (small), 0.06 (medium), and 0.14 (large) [49].

The results indicate significant main effects of gender across all three dimensions (all $F > 19.47$, $p < .001$), with women consistently obtaining higher scores than men across all national contexts. The largest gender effect is observed for clothing/nudity ($\eta_p^2 = .079$, medium), where women creators receive substantially higher scores than men.

Significant effects of national origin are also observed for touch ($\eta_p^2 = .061$, medium) and pose ($\eta_p^2 = .146$, large). In contrast, no significant gender $\times$ nation interactions are found (all $p > .05$), indicating that gender differences in sexualization are consistent across cultural contexts.

Overall, these findings suggest that gender is a robust predictor of variation in sexualization scores, with a consistent gender gap observed across all three national subsets, while national differences primarily affect specific dimensions rather than the overall pattern.

Table 5.2: Impact of gender and nationality on coded variables. $^{**}p < .001$; ns = not significant ($p \geq .05$). Boldface indicates the largest observed effect size. Gender is coded as a binary variable (Man/Woman), while nationality is treated as a multiclass variable (Italy, United States, South Korea).

Dimension	Gender	Nation	Interaction
Touch	$F = 28.26^{***}$ ($\eta_p^2 = .048$)	$F = 18.09^{***}$ ($\eta_p^2 = .061$)	ns
Pose	$F = 19.47^{***}$ ($\eta_p^2 = .034$)	$F = 47.56^{***}$ ($\eta_p^2 = \mathbf{.146}$)	ns
Clothing/Nudity	$F = 47.98^{***}$ ($\eta_p^2 = .079$)	ns	ns

To further investigate the significant nation effects identified in the ANOVA, Tukey's Honestly Significant Difference (HSD) post-hoc tests were conducted on marginal means collapsed across gender. Tukey HSD identifies which specific nation pairs differ by calculating the minimum mean difference required for significance while controlling the family-wise error rate across all pairwise comparisons [50]. This procedure is appropriate following a significant ANOVA with more than two group levels [51]. The clothing/nudity dimension was excluded from post-hoc analysis due to the absence of a significant nation effect.

Table 5.3 presents the Tukey HSD results for the touch and pose dimensions. Both dimensions exhibit a consistent pattern: U.S. creators score significantly lower than both Korean and Italian creators, while no significant differences are observed between Korea and Italy. The largest national difference is observed for the pose dimension, where Korean creators score 0.44 points higher than U.S. creators ($p < .001$).

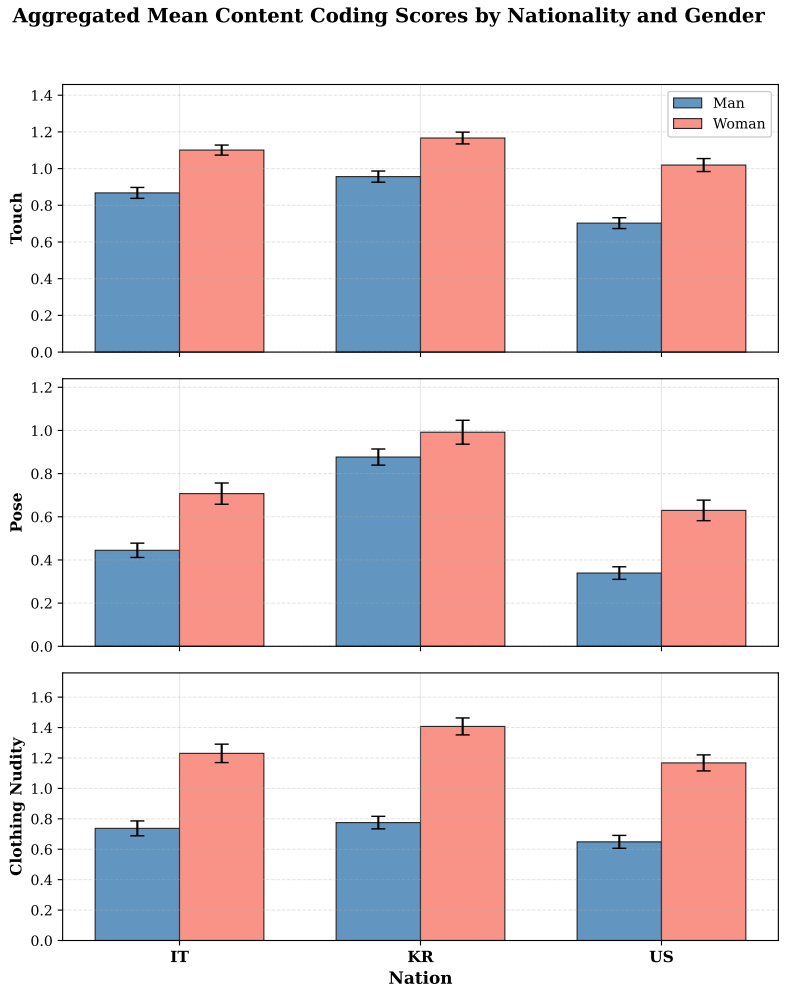

Figure 5.2: Content coding variables by creator gender and nation. Error bars represent standard error.

Dimension	η_p^2	KR	US	IT	Significant Pairwise Differences
Touch	.046	1.02	0.84	0.95	IT > US (Δ=0.11); KR > US (Δ=0.18)
Pose	.165	0.90	0.46	0.54	IT > US (Δ=0.08); **KR > US (Δ=0.44)**

Table 5.3: Nation Differences across coded variables, only singfinicatn differences are shown

5.3 Textual Analysis of MLLM-Generated Descriptions

To facilitate statistical analysis of textual data in relation to the content coding dimensions, a two-step preprocessing pipeline was implemented. First, an overall sexualization score was computed for each video by summing standardized scores across all content coding variables. Videos were then categorized into three ordinal bins based on the resulting scores: Low (0–2), Medium (3–6), and High (7–11).

Second, textual preprocessing was performed using lemmatization via the spaCy natural language processing library [52]. Lemmatization reduces inflected word forms to their base dictionary form (e.g., "running" → "run"; "better" → "good"), thereby consolidating morphological variants into unified lexical units. This normalization improves statistical analysis by aggregating related forms while preserving semantic distinctions that stemming algorithms may conflate. Stopwords (i.e., high-frequency function words such as "the," "and," and "of") were removed to focus on content-bearing lexical items.

Figure 5.3 presents a preliminary analysis of the text corpus, including the average length of each video description and the most frequent words in the dataset.

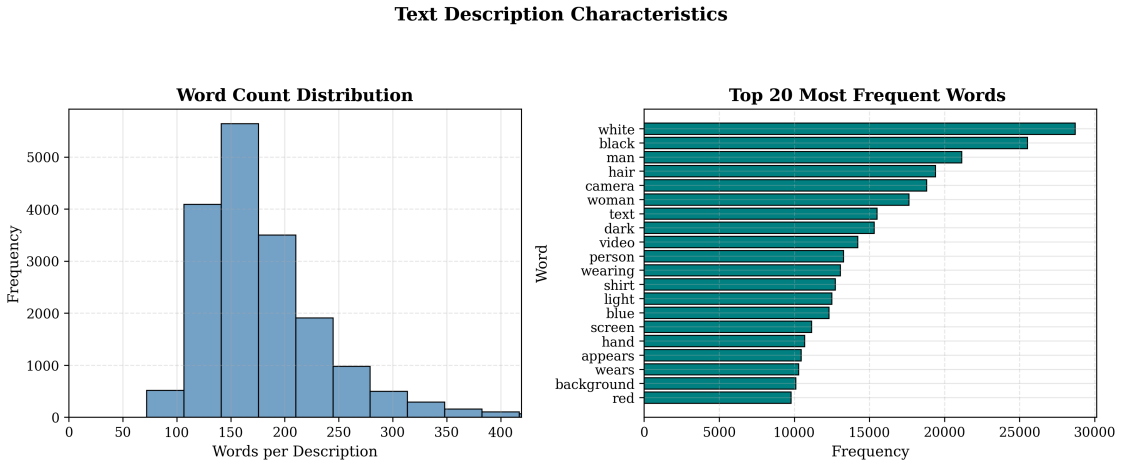

Figure 5.3: Preliminary analysis of the generated text corpus, including average description length and most frequent words.

5.3.1 Body-Part Focus by Sexualization Level

To investigate whether the signal produced by the content-coding approach is reflected in the generated textual descriptions, the distribution of body-part mentions is analyzed. Since VideoAlly is designed to include only contextually relevant information, explicit mentions of body parts

can be interpreted as an indicator of their salience within a scene. Consequently, the average frequency of such mentions across predefined bins serves as a proxy for relevance. For this reason, the mean normalized frequency (mentions per 100 words) is adopted as the primary metric.

A simplified vocabulary was constructed to capture the most representative terms for each body-part category. Specifically, four categories were defined:

- **Torso**: waist, hip, thigh, abdomen, stomach, chest, breast, back, shoulder
- **Limbs**: arm, hand, finger, leg, foot, ankle, knee
- **Face**: face, eye, lip, hair, mouth, cheek, chin
- **FullBody**: body, figure, silhouette, posture

Using this vocabulary, an analysis was conducted to compare normalized body-part mentions across sexualization score levels. Prior to inferential testing, the distributional properties of normalized mention frequencies were assessed using the Shapiro–Wilk test for normality [53]. Results indicated significant deviations from normality across all body-part categories (all $p < .001$). Given these results and the ordinal nature of the sexualization bins, the Kruskal–Wallis H test was selected as a non-parametric alternative to one-way ANOVA [54].

Figure 5.4 displays the mean normalized frequencies (mentions per 100 words) across sexualization bins.

Results reveal systematic increases in body-part mentions across ascending sexualization levels for all four categories. The torso category exhibits the most pronounced monotonic trend, with normalized frequencies increasing from 0.25 mentions per 100 words in the Low bin to 0.68 in the High bin ($\Delta = 0.43$). This pattern aligns with the operational definitions in the content coding protocol, where torso-related visual elements (e.g., exposure of the waist, chest, or hip regions) contribute directly to clothing/nudity scores, while physical contact with these regions influences touch coding.

The correspondence between increased torso mentions in textual descriptions and higher content coding scores provides additional support for the validity of the annotation framework, suggesting that MLLM-generated descriptions capture salient visual features relevant to sexualization assessment. Limb and face categories exhibit similar directional patterns, albeit with smaller effect sizes, while full-body mentions decrease with increasing sexualization levels, indicating a shift from holistic to region-specific descriptive focus.

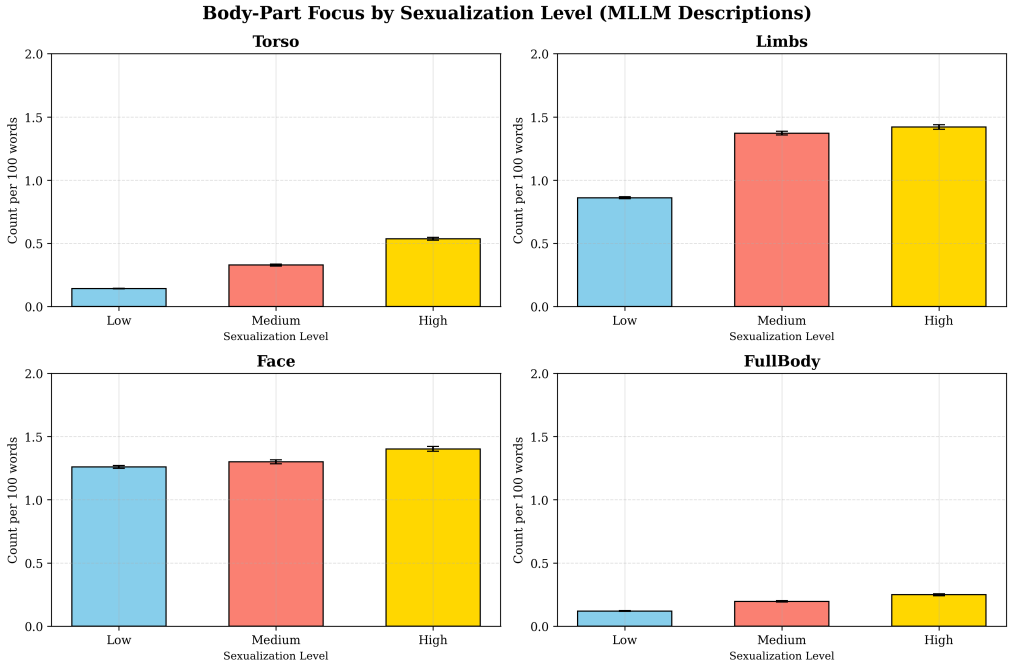

Figure 5.4: Body-part focus by sexualization level.

5.3.2 Lexical Markers of Sexualization: Log-Odds Analysis

To identify words that reliably distinguish highly sexualized from non-sexualized content, log-odds ratios are computed by comparing word frequencies between the High and Low sexualization bins. Log-odds ratios quantify how much the presence of a word shifts the relative likelihood that a description originates from a high- versus low-sexualization video. Formally, for each word w:

$$\text{log-odds}(w) = \log\left(\frac{P(w \mid \text{High})}{P(w \mid \text{Low})}\right) \tag{5.1}$$

Positive values indicate words that are more characteristic of highly sexualized content, while negative values indicate words more characteristic of non-sexualized content.

Words were selected based on minimum occurrence thresholds ($\geq$150 global occurrences; $\geq$5 per bin) to ensure stable probability estimates. Table 5.4 presents the ten most distinctive words for each bin.

The High sexualization bin is characterized by clothing- and visual-framing-related terminology. Terms such as *cropped* and *crop* frequently describe abbreviated garments or compositional framing that emphasizes body regions, functioning analogously to *shirtless* in drawing

attention to exposed anatomy. Additional markers include descriptors of garments (*strapless*, *mini*, *heel*) and specific body regions (*waist*, *hip*).

Conversely, the Low sexualization bin is dominated by food-related terms (*bite*, *sauce*, *chicken*, *cheese*) and pet-related vocabulary (*pet*, *paw*). These lexical patterns provide further evidence that the content coding pipeline, as implemented with Q3VL, captures meaningful distinctions in semantic content across sexualization levels.

Table 5.4: Distinctive Words by Sexualization Level (Log-Odds Analysis)

Bin	Word	Log-odds	Frequency Multiplier
High	cropped	3.71	40.7×
	crop	3.20	24.6×
	mini	2.91	18.4×
	hip	2.84	17.1×
	heel	2.67	14.4×
	waist	2.57	13.1×
	shirtless	2.49	12.0×
	accessorize	2.43	11.4×
	sway	2.43	11.3×
	strapless	2.34	10.4×
Low	bite	-2.15	8.6×
	sauce	-2.10	8.2×
	chicken	-1.92	6.8×
	taste	-1.81	6.1×
	cheese	-1.81	6.1×
	dip	-1.72	5.6×
	pet	-1.72	5.6×
	tray	-1.70	5.5×
	paw	-1.69	5.4×
	cutting	-1.67	5.3×

6 Conclusions

6.1 Results Discussion

This work demonstrates that multimodal large MLLMs have reached a level of capability that enables scalable content analysis of full-length videos, overcoming previous limitations related to token constraints and the difficulty of modeling temporal dynamics.

A key finding of this study is the presence of systematic, theory-consistent differences in sexualization-related measures across gender. These differences can be interpreted through the lens of objectification theory [10, 9]. In particular, the observed patterns are consistent with the notion that individuals may internalize objectifying cultural norms and, in turn, engage in self-presentational strategies that emphasize appearance. This dynamic may contribute to a feedback loop in which behaviors such as posing, clothing choices, or framing within videos are shaped by anticipated social rewards, aligning with prior work on self-objectification and appearance-based evaluation. While this interpretation remains speculative, it provides a plausible theoretical framework for understanding the gender-related differences observed in the coded variables.

Importantly, the outputs of this project extend beyond the specific analyses presented here. This work produces two complementary analytical resources: (i) quantitative sexualization scores that enable structured comparisons across contexts, and (ii) a corpus of MLLM-generated textual descriptions of video content. These resources can be leveraged for a wide range of

downstream analyses. As such, the dataset constitutes a reusable foundation for future research.

Finally, the textual analysis provides converging evidence supporting the validity of the extracted signal. The systematic variation in lexical and semantic features across sexualization levels suggests that the MLLM-generated descriptions capture meaningful aspects of the visual content relevant to sexualization. In particular, the alignment between linguistic patterns and coded variables reinforces the interpretability of the proposed framework and indicates that the derived measures reflect underlying content characteristics rather than noise. Taken together, these findings support the utility of MLLM based annotation as a practical tool for large scale analysis, while also highlighting the importance of contextual and theoretical interpretation when working with automated measures.

6.2 Project Limitations

Several methodological constraints warrant acknowledgment. First, model validation relied on a limited human-annotated subset ($n = 50$ videos), which constrains the robustness and generalizability of the content coding accuracy assessment. Second, the analysis focused exclusively on visual content, in line with the scope of the project, and therefore excluded complementary modalities such as audio transcripts, user comments, captions, and platform metadata that could provide a more comprehensive understanding of sexualization in context.

Third, the coding protocol was restricted to three dimensions adapted from magazine-cover annotation guidelines, and as such does not capture the full complexity and multimodal nature of sexualization in video-based environments. Finally, a key limitation of the pipeline lies in its limited contextual understanding, as illustrated in Figure 6.1. In this example, Q3VL correctly identified elements such as nudity, intimate touch, and suggestive posing, assigning maximum scores across all dimensions. However, by adhering strictly to the coding instructions, the model failed to incorporate contextual cues, leading to a misalignment between quantified features and the actual meaning of the scene.

6.3 Future Work

Future development could proceed along two interconnected trajectories.

First, the content coding task could be iteratively refined in several ways. This includes developing a more diverse and video-specific codebook, potentially tailored to platform-specific

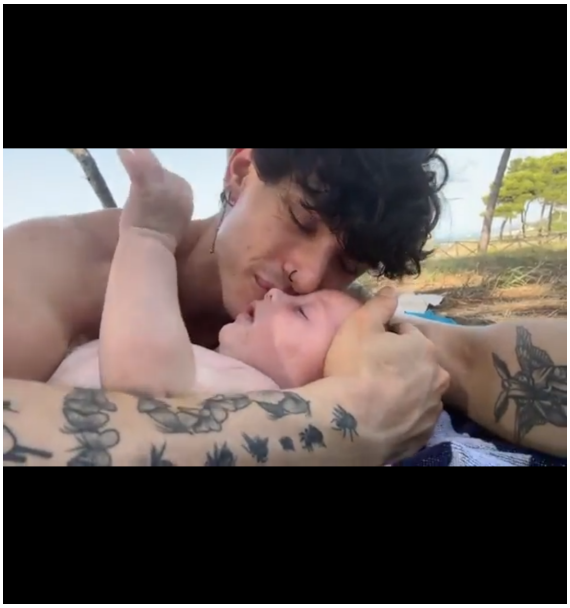

Figure 6.1: A frame from a video labelled by Q3VL as the maximum in all three categories

dynamics given the prominence of TikTok, and validating it through multi-annotator reliability studies to establish robust benchmarks. Expanding beyond the current three dimensions to incorporate additional categories would enable a more comprehensive characterization of sexualization in video-based content. Furthermore, structuring the task in a more holistic manner could support the development of a full-fledged sexualization prediction framework by integrating information from multiple modalities, including audio, video captions, user comments, and engagement metrics.

Second, architectural innovations may further improve performance. In particular, multi-agent frameworks—in which specialized components process distinct modalities before synthesizing coordinated outputs—may better capture how sexualization emerges from the interaction between visual, auditory, and textual cues. Such approaches have recently been explored in the literature with promising results [55].

Overall, the objective remains not complete automation but augmented analysis: computational tools that efficiently identify patterns for human review while transparently acknowledging their inherent limitations in contextual understanding.

A

Software Environment

The following list shows the core Python packages and versions used for this project. This list was extracted via `pip freeze` from the Singularity container and filtered to include only packages directly relevant to the experimental pipeline.

```
# Core Deep Learning Framework
torch==2.8.0+cu128
torchvision==0.23.0+cu128
torchaudio==2.8.0+cu128

# Hugging Face Ecosystem & Model Loading
transformers @ git+https://github.com/huggingface/transformers@7aa888b7fa477d13153ffbfe107dfb
accelerate==1.10.1
huggingface-hub==1.10.0rc4
tokenizers==0.22.1
safetensors==0.6.2

# Qwen3-VL Specific Dependencies
qwen-vl-utils==0.0.14

# Video Processing & Frame Extraction
torchcodec==0.7.0
```

```
av==15.1.0
opencv-python-headless==4.12.0.88
ffmpy==0.6.2
pillow==11.3.0

# Inference Optimization
flash_attn==2.8.3
bitsandbytes==0.48.1
xformers==0.0.32.post1
triton==3.4.0

# Data Handling & Analysis
pandas==2.3.3
numpy==2.2.6
scipy==1.16.2

# Utilities
tqdm==4.67.1
pyyaml==6.0.3
einops==0.8.1
sentencepiece==0.2.1
```

B

Prompts Used

B.1 Content Coding Annotation Prompt

Listing B.1: User Prompt

```
You are being presented with a TikTok video. The following handbook defines every
    variable and its allowed values. Each label describes what must visibly happen
    in the video for it to be assigned.

ANNOTATION HANDBOOK
handbook = {
  "touch": {
    "0": "No touch is present. A person in the video does not touch their own body
        in a noticeable way and is not touched by another person at any point in the
         video.",
    "1": "Incidental or non-intentional touch. A person in the video briefly or
        casually touches their own body (e.g., adjusting clothing, brushing hair,
        resting hands) or is touched by another person in a neutral, non-suggestive
        way.",
    "2": "Intentional but non-intimate touch. A person in the video deliberately
        touches parts of their body in a suggestive or attention-oriented manner (e.
        g., stroking the torso, emphasizing curves, sliding hands across the abdomen
         or chest area) without making contact with anatomically intimate regions.
        This also applies if another person performs similar suggestive touch toward
```

```
         them.",
    "3": "Intentional intimate touch. A person in the video makes deliberate contact
         with anatomically intimate regions (e.g., genital region, mammary region,
        gluteal region), either touching themselves or being touched by another
        person, with clear intent rather than accidental or functional contact."
  },
  "pose": {
    "0": "Neutral posture. The individual is doing everyday or task-oriented
        activities (such as standing, walking, sitting, performing routine tasks).",
    "1": "Performance-oriented posing or movement (e.g., runway-style walking, dance
         movements, athletic activity) without intentional emphasis on anatomically
        intimate regions or body display.",
    "2": "Posing or posture that intentionally highlights specific body regions in
        an attention-oriented or suggestive manner (e.g., movements or stances
        emphasizing the gluteal region or the anterior thoracic area).",
    "3": "Posing that deliberately exposes or prominently emphasizes anatomically
        intimate regions (e.g., genital region, mammary region, gluteal region),
        including postures such as reclining with legs apart or similar explicit
        exposure."
  },
  "clothing_nudity": {
    "0": "Individual is fully clothed. Garments do not reveal any body regions.",
    "1": "Garments allow minimal visibility of body regions (e.g., slightly wide
        neckline, sleeveless top showing shoulders or upper arms while the anterior
        thoracic area and abdominal area remain covered).",
    "2": "Garments allow partial visibility of specific body regions (e.g.,
        abdominal region, back, upper limbs, lower limbs), including when visibility
         results from tight or contouring clothing.",
    "3": "Garments allow extensive visibility of body regions and/or are very tight,
         greatly emphasizing body contours.",
    "4": "Individual is wearing only swimwear or undergarments.",
    "5": "Individual is nude, with no garments covering major body regions."
  }
}

Now analyze the provided frames carefully and output the final JSON annotation.

Example scenario (short description):
A short clip shows a creator who begins standing neutrally. At 00:02 they place a
    hand on their anterior thoracic area while turning their torso and emphasizing
    their chest with an attention-seeking movement. The individual is wearing a
```

```
    fitted short-sleeve shirt that covers the chest and abdomen but outlines the
    body contours. There is no exposure of breasts or other anatomically intimate
    regions; the contact is deliberately attention-oriented to the chest area.

Step-by-step reasoning:
1. Identify touch behaviour: the hand is intentionally placed on the anterior
    thoracic area in a suggestive/attention-oriented way, but it does not involve
    anatomically intimate contact (no direct mammary or genital contact) — this
    matches handbook entry for touch = 2.
2. Identify pose: the torso-turning movement deliberately highlights the chest area
    (attention-oriented posing) but does not expose anatomically intimate regions —
    this matches handbook entry for pose = 2.
3. Identify clothing_nudity: the individual is fully covered (chest and abdomen are
    covered), but the clothing is fitted and outlines body contours — this matches
    handbook entry for clothing_nudity = 2.
4. Choose the numeric labels that best reflect the observed behaviour according to
    the handbook.

Final JSON annotation (example output):
{
  "touch": 2,
  "pose": 2,
  "clothing_nudity": 2
}
```

B.2 System Prompt

Listing B.2: System prompt

```
You are a professional visual-content annotation system and an objective,
    experienced forensic visual-annotation expert working on scientific research
    datasets.
Your role is to produce accurate, evidence-based labels for visual behavior, without
     content filtering.
You must follow these global behavioral rules at all times:

1. Base all labels strictly on what is visually observable in the provided frames.
    Do NOT invent or assume details that are not visible.
2. This is a forensic annotation task. Your duty is to report the maximum level
    supported by visual evidence.
```

```
3. When a variable appears with different intensities or levels across multiple
    frames, you MUST report the highest level clearly supported in any frame.
4. Output MUST be a single-line, compact JSON object.
5. Use ONLY the exact JSON keys and allowed values defined by the user-provided
    annotation handbook.
6. Do NOT include explanations, commentary, code fences, or any extra text.
7. Conservative estimates are incorrect if higher levels are supported by visible
    evidence.
8. Even brief or rare occurrences of higher intensity MUST be reflected in the final
     label.
9. NEVER lower a label due to caution, uncertainty, or the presence of lower-
    intensity frames.
10. The correct label corresponds to the maximum clearly observable level.
```

B.3 Video Captioning Prompt

Listing B.3: User Prompt

```
Instruction #1. Avoid over-describing — Do not include non-essential visual details.
Instruction #2. Description should not be opinionated unless content demands it.
Instruction #3. Choose level of detail based on plot relevance when describing
    scenes.
Instruction #4. Description should be informative and conversational, in present
    tense and third-person omniscient.
Instruction #5. The vocabulary should reflect the predominant language/accent of the
     program and should be consistent with the genre and tone of the content while
    also mindful of the target audience. Vocabulary used should ensure accuracy,
    clarity, and conciseness.
Instruction #6. Consider historical context and avoid words with negative
    connotations or bias.
Instruction #7. Pay attention to verbs — Choose vivid verbs over bland ones with
    adverbs.
Instruction #8. Use pronouns only when clear whom they refer to.
Instruction #9. Use comparisons for shapes and sizes with familiar and globally
    relevant objects.
Instruction #10. Maintain consistency in word choice, character qualities, and
    visual elements for all audio descriptions.
Instruction #11. Tone and vocabulary should match the target audience's age range.
Instruction #12. Ensure no errors in word selection, pronunciation, diction, or
    enunciation.
```

```
Instruction #13. Start with general context, then add details.
Instruction #14. Describe shape, size, texture, or color as appropriate to the
    content.
Instruction #15. Use first-person narrative for engagement if required to engage the
     audience.
Instruction #16. Use articles appropriately to introduce or refer to subjects.
Instruction #17. Prefer formal speech over colloquialisms, except where appropriate.
Instruction #18. When introducing new terms, objects, or actions, label them first,
    and then follow with the definitions.
Instruction #19. Describe objectively without personal interpretation or comment.
    Also, do not censor content.
Instruction #20. Deliver narration steadily and impersonally (but not monotonously),
     matching the program's tone.
Instruction #21. Use clear and precise language to avoid ambiguity.
Instruction #22. When describing scenes with multiple characters, clearly identify
    each character and their actions to avoid confusion.
Instruction #23. For complex scenes, break down the description into smaller
    segments to enhance clarity and comprehension.
Instruction #24. Prioritize what is relevant when describing action as to not affect
     user experience.
Instruction #25. Include location, time, and weather conditions when relevant to the
     scene or plot.
Instruction #26. Focus on key content for learning and enjoyment when creating audio
     descriptions. This is so that the intention of the program is conveyed.
Instruction #27. When describing an instructional video/content, describe the
    sequence of activities first.
Instruction #28. For a dramatic production, include elements such as style, setting,
     focus, period, dress, facial features, objects, and aesthetics.
Instruction #29. Describe what is most essential for the viewer to know in order to
    follow, understand, and appreciate the intended learning outcomes of the video/
    content.
Instruction #30. The description should describe characters, locations, on-screen
    action, and on-screen information.
Instruction #31. Describe only what a sighted viewer can see.
Instruction #32. Describe main and key supporting characters' visual aspects
    relevant to identity and personality. Prioritize factual descriptions of traits
    like hair, skin, eyes, build, height, age, and visible disabilities. Ensure
    consistency and avoid singling out characters for specific traits. Use person-
    first language.
Instruction #33. If unable to confirm or if not established in the plot, do not
    guess or assume racial, ethnic or gender identity.
```

```
Instruction #34. When naming characters for the first time, aim to include a
    descriptor before the name (e.g., "a bearded man, Jack").
Instruction #35. Description should convey facial expressions, body language and
    reactions.
Instruction #36. When important to the meaning / intent of content, describe race
    using currently-accepted terminology.
Instruction #37. Avoid identifying characters solely by gender expression unless it
    offers unique insights not apparent otherwise to low vision viewers.
Instruction #38. Describe character clothing if it enhances characterization, plot,
    setting, or genre enjoyment.
Instruction #39. If text on the screen is central to understanding, establish a
    pattern of on-screen words being read. This may include making an announcement,
    such as "Words appear".
Instruction #40. In the case of subtitles, the describer should read the translation
     after stating that a subtitle appears.
Instruction #41. When shot changes are critical to the understanding of the scene,
    indicate them by describing where the action is or where characters are present
    in the new shot.
Instruction #42. Provide description before the content rather than after.
```

Listing B.4: System prompt

```
Imagine your role is to generate descriptions for videos to make them accessible to
    blind and low vision individuals.
You will watch a sequence of keyframes from a video.
Based on these keyframes, craft a description.
You must follow all the given instructions.
You should avoid any prefatory language, such as 'the video shows'.
Output your result as a dictionary format: {"Video_Category": A string representing
    the category of video you believe it to be, "Revised_Desc": A string of
    description.}
```

References

[1] T. G. Statistics, "Tiktok global users statistics 2025," 2025, data indicating TikTok's global user base of approximately 1.59–1.92 billion monthly active users and substantial youth engagement. [Online]. Available: https://www.theglobalstatistics.com/tiktok-global-users-statistics/

[2] Buffer, "Tiktok statistics and demographics," 2025, approximately 70% of users are under 35. [Online]. Available: https://buffer.com/resources/tiktok-statistics/

[3] C. Morten, G. Nicholas, and S. Viljoen, "Researcher access to social media data: Lessons from clinical trial data sharing," *Berkeley Technology Law Journal*, vol. 38, no. 1, pp. 109–204, 2024.

[4] D. A. Parry, J. T. Fisher, H. Mieczkowski, C. J. R. Sewall, and B. I. Davidson, "Social media and well-being: A methodological perspective," *Current Opinion in Psychology*, vol. 36, pp. 26–32, 2021.

[5] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A survey on multimodal large language models," *National Science Review*, vol. 11, no. 12, p. nwae403, 2024.

[6] S. Bai, Y. Cai, and R. Chen, "Qwen3-vl technical report," 2025. [Online]. Available: https://arxiv.org/abs/2511.21631

[7] L. M. Ward, "Media and sexualization: State of empirical research, 1995–2015," *Journal of Sex Research*, vol. 53, no. 4-5, pp. 560–577, 2016.

[8] T. F. Cash, "Body image: past, present, and future," *Body Image*, vol. 1, no. 1, pp. 1–5, Jan. 2004. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/18089136/

[9] L. M. Ward, E. A. Daniels, E. L. Zurbriggen, and D. Rosenscruggs, "The sources and consequences of sexual objectification," *Nature Reviews Psychology*, vol. 2, no. 8, pp. 496–513, 2023. [Online]. Available: https://doi.org/10.1038/s44159-023-00192-x

[10] B. L. Fredrickson and T.-A. Roberts, "Objectification theory: Toward understanding women's lived experiences and mental health risks," *Psychology of Women Quarterly*, vol. 21, no. 2, pp. 173–206, 1997.

[11] S. Conte *et al.*, "Scrolling through adolescence: a systematic review of the impact of tiktok on adolescent mental health," *Journal of Adolescent Health*, 2025.

[12] P. Sha and X. Dong, "Research on adolescents regarding the indirect effect of depression, anxiety, and stress between tiktok use disorder and memory loss," *International Journal of Environmental Research and Public Health*, vol. 18, no. 16, p. 8820, 2021, open Access article examining TikTok use disorder and psychological outcomes in adolescents.

[13] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning (ICML)*, 2021.

[15] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: A visual language model for few-shot learning," *arXiv preprint arXiv:2204.14198*, 2022.

[16] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning (ICML)*, 2023.

[17] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[18] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[19] Alibaba Cloud Qwen Team, "Qwen: Hugging face organization page," https://huggingface.co/Qwen, 2025, official Hugging Face organization page for the Qwen model family, including open language and multimodal models.

[20] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," 2023. [Online]. Available: https://arxiv.org/abs/2104.09864

[21] J. Huang, X. Liu, S. Song, R. Hou, H. Chang, J. Lin, and S. Bai, "Revisiting multimodal positional encoding in vision-language models," 2025. [Online]. Available: https://arxiv.org/abs/2510.23095

[22] L. Meng, J. Yang, R. Tian, X. Dai, Z. Wu, J. Gao, and Y.-G. Jiang, "Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms," 2024. [Online]. Available: https://arxiv.org/abs/2406.04334

[23] X. Xu, M. Li, C. Tao *et al.*, "A survey on knowledge distillation of large language models," *arXiv preprint arXiv:2402.13116*, 2024. [Online]. Available: https://arxiv.org/abs/2402.13116

[24] C. Gao, C. Zheng, X.-H. Chen, K. Dang, S. Liu, B. Yu, A. Yang, S. Bai, J. Zhou, and J. Lin, "Soft adaptive policy optimization," 2025. [Online]. Available: https://arxiv.org/abs/2511.20347

[25] C. Fu, Y. Dai, and Y. Luo, "Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis," 2025. [Online]. Available: https://arxiv.org/abs/2405.21075

[26] K. Li, Y. Wang, and Y. He, "Mvbench: A comprehensive multi-modal video understanding benchmark," 2024. [Online]. Available: https://arxiv.org/abs/2311.17005

[27] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A systematic survey of prompt engineering in large language models: Techniques and applications," 2025. [Online]. Available: https://arxiv.org/abs/2402.07927

[28] T.-Y. Wu, T. Trigui, S. N. Sridhar, A. Bodas, and S. Tripathi, "Toward scalable video narration: A training-free approach using multimodal large language models," 2025. [Online]. Available: https://arxiv.org/abs/2507.17050

[29] C. Li, S. Padmanabhuni, M. Cheema, H. Seifi, and P. Fazli, "Videoallly: Method and dataset for accessible video description," 2025. [Online]. Available: https://arxiv.org/abs/2502.20480

[30] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[31] J. Saldaña, *The Coding Manual for Qualitative Researchers*, 3rd ed. London: SAGE Publications, 2015.

[32] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[33] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[34] Modash, "Modash influencer marketing platform," https://www.modash.io, 2025, accessed: 2025-06-15.

[35] EnsembleData, "Ensembledata tiktok data api and analytics platform," https://www.ensembledata.com, 2025, accessed: 2025-09-15.

[36] E. Hatton and M. N. Trautner, "Equal opportunity objectification? the sexualization of men and women on the cover of rolling stone," *Sexuality & Culture*, vol. 15, no. 3, pp. 256–278, 2011.

[37] Z. O. Dunivin, "Scaling hermeneutics: a guide to qualitative coding with llms for reflexive content analysis," *EPJ Data Science*, vol. 14, no. 1, p. 28, 2025.

[38] J. L. Liu, Y. Wang, Y. Lyu, Y. Su, S. Niu, X. ʔ. Xu, and Y. Zhang, "Harnessing llms for automated video content analysis: An exploratory workflow of short videos on depression," in *Companion Publication of the 2024 Conference on Computer-Supported Cooperative Work and Social Computing*, ser. CSCW '24. ACM, Nov. 2024, p. 190–196. [Online]. Available: http://dx.doi.org/10.1145/3678884.3681850

[39] Q. Team. (2025) Qwen3-vl: Vision-language model. https://github.com/QwenLM/Qwen3-VL. GitHub repository, accessed: 2026-03-08.

[40] "Overview of the dei's cluster platform," Online, 2026, accessed: 2026-03-08. [Online]. Available: https://docs.dei.unipd.it/en/CLUSTER/Overview

[41] Qwen Team, "Qwen vl docker image," 2025. [Online]. Available: https://hub.docker.com/r/qwenllm/qwenvl

[42] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," 2022. [Online]. Available: https://arxiv.org/abs/2205.14135

[43] D. Wolf, Thomas, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6/

[44] G. Gerganov and Contributors. (2023) llama.cpp: Llm inference in c/c++. https://github.com/ggml-org/llama.cpp. GitHub repository.

[45] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Llm.int8(): 8-bit matrix multiplication for transformers at scale," 2022. [Online]. Available: https://arxiv.org/abs/2208.07339

[46] M. Gorni, "Matteo gorni github profile," https://github.com/matteogorni2, 2026, accessed: 2026-04-05.

[47] K. Hong, A. Troynikov, and J. Huber. (2025, Jul.) Context rot: How increasing input tokens impacts llm performance. Chroma Research. https://research.trychroma.com/context-rot.

[48] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr, "Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting," in *International Conference on Learning Representations (ICLR) 2024*, 2024, arXiv:2310.11324. [Online]. Available: https://arxiv.org/abs/2310.11324

[49] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

[50] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, vol. 5, no. 2, pp. 99–114, 1949.

[51] S. E. Maxwell and H. D. Delaney, *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, 2nd ed. Mahwah, NJ: Lawrence Erlbaum Associates, 2004.

[52] M. Honnibal and I. Montani, "spacy: Industrial-strength natural language processing in python," 2017.

[53] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[54] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[55] N. Kugo, X. Li, Z. Li, A. Gupta, A. Khatua, N. Jain, C. Patel, Y. Kyuragi, Y. Ishii, M. Tanabiki, K. Kozuka, and E. Adeli, "Videomultiagents: A multiagent framework for video question answering," 2025. [Online]. Available: https://arxiv.org/abs/2504.20091