Scientific Journals
An international open-access platform

Hanlin Letters

Article

OPEN ACCESS

Human-Centered Interfaces for Describing Complex Video Edits with Natural Language and Example Images

Nguyen Thai Duong,[1] Le Thi Bao Ngan[2]

[1] Department of Information Systems, Thu Dau Mot University, Đường 30 Tháng 4, Phường Phú Thọ, Thủ Dầu Một, Vietnam;
[2] Department of Software Engineering, Quang Nam University, Đường Trần Hưng Đạo 102, Phường Tân Thạnh, Tam Kỳ, Vietnam;

Abstract

Digital video editing is increasingly shaped by machine learning models that can transform content through high-level specifications, yet most interfaces remain grounded in low-level timelines, keyframes, and parameter panels. There is growing interest in describing edits using natural language, sometimes complemented by example images that express desired style, composition, or local modifications. However, the design of human-centered interfaces that allow users to specify complex, multi-step video edits through such multimodal descriptions remains only partially understood. This paper examines how natural language and example images can be combined into interfaces that allow users to describe detailed spatiotemporal changes while maintaining control, predictability, and the ability to iteratively refine results. The paper introduces a problem formulation in which user intent is modeled as a structured edit plan over a video, develops a multimodal representation that connects language, images, and video content, and sketches algorithmic mappings from interface-level descriptions to concrete edit operations. The discussion emphasizes ambiguity management, edit transparency, and support for exploratory workflows. Potential evaluation strategies grounded in usability, expressivity, and edit quality are outlined, with a focus on how different interaction patterns affect user understanding of the underlying automated system. The paper aims to provide a technically oriented perspective on the joint design of models and interfaces for human-centered, language-driven video editing supported by example imagery, with attention to both algorithmic structure and human factors.

1. Introduction

Video editing tools have historically been designed around the metaphor of a timeline populated with clips, tracks, and keyframes [1]. The user expresses intent by manipulating layers, numerical sliders, curves, and masks that specify how visual properties change over time. This representation gives experienced editors detailed control but requires substantial expertise, and it often forces users to translate high-level conceptual goals into many low-level operations. At the same time, advances in generative models and learned video representations have demonstrated that much of the underlying transformation can be driven by higher level semantic signals. These developments motivate the question of how human-centered interfaces might allow video edits to be described more directly using natural language and example images.

Natural language promises an accessible way to specify intent, using phrases like “make the sky more dramatic near sunset” or “blur the background when the interviewer walks into the frame”. Example images can express visual properties that are difficult to describe in words, such as specific color grading, texture, or compositional style [2]. In principle, combining language with example images provides a rich channel through which users can describe complex video edits without learning many parameters. In practice, however, text-only prompting often leads to ambiguous results, and example-based style transfer can be difficult to localize or control. Interfaces must reconcile the flexibility and ambiguity of human descriptions with the need for precise, reproducible editing operations.

Complex video edits involve multiple objects, temporal structure, and conditional behavior. A user may wish to apply different color treatments to foreground and

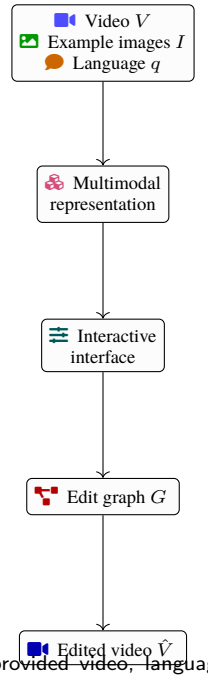

Figure 1: Overview of the end-to-end system: user-provided video, language, and example images are encoded in a multimodal representation that supports interaction, construction of an edit graph, and generation of the edited video.

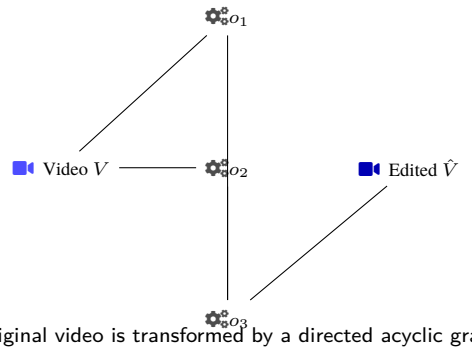

Figure 2: Edit-graph formulation: the original video is transformed by a directed acyclic graph of masked operations o_1, o_2, o_3; their composition yields the edited video and exposes structure suitable for inspection and refinement.

background, alter motion trajectories for specific objects, or introduce effects that depend on events occurring at particular times. Such edits can be decomposed into sequences or graphs of operations that act on spatial regions and time intervals. When these operations are controlled by a learned model that interprets natural language and example images, the interface must make the model's actions understandable and revisable [3]. Otherwise, users may be surprised by nonlocal changes, unintended alterations of important content, or brittle behavior when their descriptions deviate from training data.

Human-centered design frames this challenge as a joint problem of representation, interaction, and algorithmic mapping. At the level of representation, the system requires a shared space in which video content, language descriptions, and example images can be related to each other. At the interaction level, the interface should help users externalize their mental model of the desired edit, inspect and adjust how the system interprets their inputs, and manage complexity over time. At the algorithmic level, the model must map multimodal descriptions to concrete edit plans that are explainable, stable under small variations in input, and responsive to user corrections.

This paper develops a technical perspective on these three layers for the specific case of complex video edits described by natural language and example images. A formal problem statement describes video as a spatiotemporal signal, user intent as a structured edit plan, and the

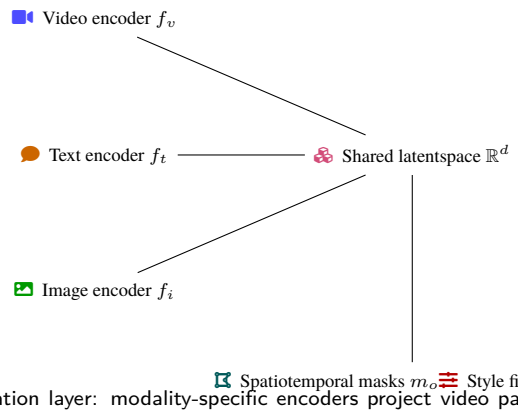

Figure 3: Multimodal representation layer: modality-specific encoders project video patches, textual phrases, and example images into a shared latent space, from which masks and style fields are derived to control localized spatiotemporal edit operations.

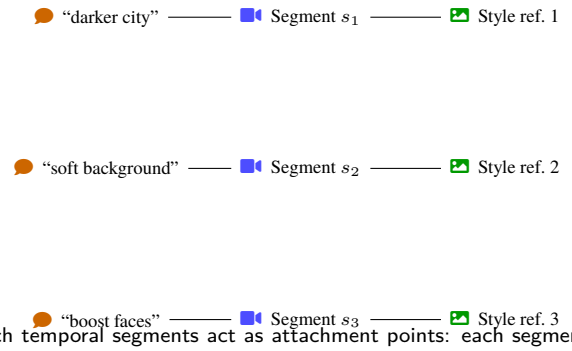

Figure 4: Interface concept in which temporal segments act as attachment points: each segment aggregates natural-language cues and example images, which the system generalizes into spatiotemporal edits while preserving the user-visible association between descriptions and footage.

interface as a mediator between human descriptions and edit graphs. Multimodal representations are introduced that embed video segments, textual phrases, and example images into a shared latent space, enabling both semantic matching and control of low-level operations. The paper then sketches algorithmic approaches for transforming interface-level specifications into sequences of parameterized operations, including probabilistic models over edit graphs and optimization-based propagation of edits across time. Throughout, the discussion integrates human-centered considerations such as transparency of model behavior, support for ambiguity resolution, and alignment with editors' mental models of their workflows.

The remainder of the paper is structured as follows. The background and problem formulation section characterizes conventional video editing workflows and states the core problem of mapping human descriptions to edit graphs. The multimodal representation section describes how video, language, and example images can be encoded into shared spaces that support both interpretation and control. The interface design section focuses on interaction techniques that combine text and images to express complex edits while exposing the system's internal decisions [4]. The algorithmic mapping section presents mathematical models and optimization procedures that connect these interface-level descriptions to underlying edit operations. An evaluation and discussion section outlines possible methodologies to study such systems and reflects on trade-offs in expressivity, predictability, and user effort. The conclusion summarizes the main ideas and highlights directions for further work that connect model development with interface design.

2. Background and Problem Formulation

Contemporary video editing practice is built on a rich ecosystem of tools that expose fine-grained control over spatial and temporal aspects of footage. In many tools, edits are represented as layers stacked along a timeline, where each layer holds either visual content or an effect. Parameters such as opacity, color balance, or transform matrices are animated over time through keyframes, and

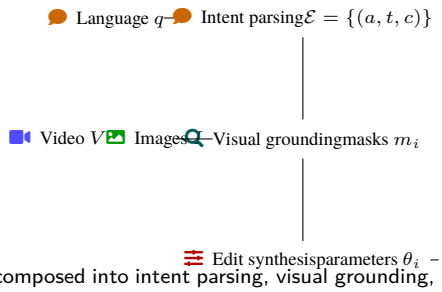

Figure 5: Algorithmic mapping decomposed into intent parsing, visual grounding, and edit synthesis: language is converted to editing acts, aligned to video regions using multimodal features, and finally translated into parameterized operations that populate the edit graph.

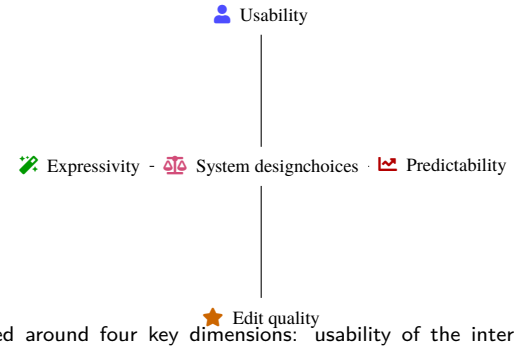

Figure 6: Evaluation space structured around four key dimensions: usability of the interface, expressivity of the edit language, perceptual quality of resulting videos, and predictability of system behavior given user descriptions.

spatially varying operations rely on masks, mattes, or rotoscoped shapes. This representation exposes an explicit structure over edits, in which each operation is associated with a region of space, an interval of time, and a set of numeric parameters [5]. Expert editors internalize this structure and develop mental models that relate conceptual goals, like emphasizing a subject, to concrete manipulations of tracks, masks, and effects.

As learned models for images and video have matured, they have begun to automate parts of this pipeline. Segmentation models provide object-level masks with minimal manual drawing. Tracking models propagate masks across frames. Color grading suggestion systems propose initial parameter settings based on learned style priors. Generative models can synthesize new content or modify existing content according to high-level prompts. However, these systems are often integrated as isolated components, requiring the user to navigate between a conventional low-level parameter space and a separate mode in which natural language or example images are used [6] [7]. The boundary between high-level intent and low-level state remains largely implicit, and it is challenging for users to predict how textual or image prompts will be translated into concrete timeline operations.

To reason about human-centered interfaces for complex, language-driven video edits, it is helpful to formulate the underlying problem in mathematical terms. Let a video be represented as a tensor

$$V \in \mathbb{R}^{T \times H \times W \times C}, \tag{1}$$

where T is the number of frames, H and W are spatial dimensions, and C is the number of channels. The user articulates intent through a natural language description q and a set of example images $I = \{I_1, \dots, I_K\}$, where each I_k is an image in $\mathbb{R}^{H' \times W' \times C}$. The system is tasked with producing a set of edit operations that, when applied to V, yield an edited video $\hat{V}$.

We can model an edit plan as a directed acyclic graph $G = (O, E)$ of operations. Each node $o \in O$ corresponds to an operation such as color adjustment, spatial warping, or compositing. Each directed edge in E specifies that the output of one operation is used as input to another [8]. Each operation o has associated parameters θ_o, as well as spatiotemporal support given by a mask function $m_o : \{1, \dots, T\} \times \{1, \dots, H\} \times \{1, \dots, W\} \to [0, 1]$.

Table 1: Core symbols introduced in the problem formulation for language-driven video editing.

Symbol	Semantic role	Type / domain	Spatiotemporal scope
V	Input video tensor	$\mathbb{R}^{T\times H\times W\times C}$	Full video
q	Natural language description of desired edits	Token sequence	Global or segment-level
$I = \{I_1, \ldots, I_K\}$	Example images specifying style or local appearance	$\mathbb{R}^{H'\times W'\times C}$	Global or region-level
$G = (O, E)$	Edit plan as a directed acyclic graph of operations	Graph structure	Full video
m_o	Spatiotemporal mask for operation o	$[0,1]^{T\times H\times W}$	Local regions
θ_o	Parameters controlling operation o	Vector in $\mathbb{R}^d$	Per-node, possibly time-varying

The edited video is given by evaluating this graph on the original video, yielding

$$\hat{V} = F(G, V), \tag{2}$$

where F composes the operations in topological order, subject to their masks and parameterizations.

From a human-centered perspective, the graph G should be interpretable to the user, at least at a coarse level. Nodes might be grouped into higher-level constructs like "foreground enhancement" or "background defocus", and masks may be visualized as overlays on the video. The interface serves as a mediator between the user's high-level description (q, I) and the graph G, supporting both automated inference of graph structure and user-driven modification. This suggests modeling the system as computing a distribution over possible edit graphs given the video, language, and images,

$$p(G \mid V, q, I), \tag{3}$$

from which one or more candidate graphs are selected and presented to the user for further refinement [9].

User satisfaction can be represented as a random variable S that depends on the edited video $\hat{V}$ and the user's internal intent. Because intent is not observed directly, the system can only approximate it through (q, I). A human-centered objective is to choose G that maximizes the expected satisfaction

$$\mathbb{E}[S \mid V, q, I], \tag{4}$$

subject to constraints on complexity, transparency, and editability. Complexity can be formalized, for example, as a function of the number of operations, the number of masks, and the degree of temporal variation in parameters, yielding a scalar

$$C(G) = \alpha_1|O| + \alpha_2|E| + \alpha_3 R(G), \tag{5}$$

where $R(G)$ measures temporal regularity of parameters and masks, and $\alpha_1, \alpha_2, \alpha_3$ are weights.

The problem can thus be framed as a constrained optimization over edit graphs,

$$G^\star = \arg\max_G \ \mathbb{E}[S \mid V, q, I] \quad \text{s.t.} \quad C(G) \le \gamma, \tag{6}$$

where γ encodes a limit on complexity appropriate for the user's expertise and the interface's capacity. In practice, $\mathbb{E}[S \mid V, q, I]$ is not directly accessible and must be approximated by surrogate measures derived from multimodal alignment scores, heuristic penalties for undesired changes, and feedback signals collected from the user over time.

This formulation highlights tension between expressivity and control [10]. A highly expressive model may propose rich and intricate graphs that match textual descriptions in abstract semantic space but are difficult for users to inspect or modify. A more constrained model might produce simpler graphs that align more closely with familiar editing concepts but may not fully capture complex instructions. Human-centered interfaces can mediate this tension by allowing users to shape both the structure of G and the criteria used to select among alternative graphs. Natural language and example images become not only de-

Table 2: Main components of the multimodal representation connecting video, text, and example images.

Component	Input	Output representation	Typical role
Video encoder f_v	Spatiotemporal patches $V_{R_{t,x,y}}$	$z^v_{t,x,y} \in \mathbb{R}^{d_v}$	Local visual semantics and appearance
Text encoder f_t	Tokens or phrases q_j	$z^t_j \in \mathbb{R}^{d_t}$	Intent phrases and references to entities
Image encoder f_i	Example image patches $I_{k,l}$	$z^i_{k,l} \in \mathbb{R}^{d_i}$	Style and composition cues from examples
Projections W_v, W_t, W_i	Modality-specific features	$\tilde{z}^v, \tilde{z}^t, \tilde{z}^i \in \mathbb{R}^d$	Shared multimodal embedding space
Temporal encoder f_τ	Sequences $(\tilde{z}^v_{1,x,y}, \ldots, \tilde{z}^v_{T,x,y})$	$r_{x,y} \in \mathbb{R}^d$	Dynamics and event-level context

scriptive inputs but also control signals that specify, for instance, how strongly style from an example image should influence a given region, or which parts of a description are nonnegotiable.

In this perspective, the design of interfaces for complex, language-driven video editing reduces to shaping the space of possible edit graphs, defining how descriptions map into that space, and providing mechanisms for users to navigate and refine points within it. The next sections elaborate the representational and interface choices that support such navigation and discuss algorithmic strategies for realizing the mapping from descriptions to editable graph structures.

3. Multimodal Representations for Language-Guided Video Editing

Realizing a mapping from user descriptions to editable video transformations requires a shared representational space that connects three modalities: the source video, natural language, and example images [11]. This space must support at least two distinct roles. First, it must support semantic matching, so that phrases like "the person in the red jacket" can be aligned with specific regions and time intervals in the video. Second, it must support control over low-level operations, enabling style vectors derived from example images to parameterize color, texture, or motion transformations. The representation must therefore capture both high-level semantics and low-level visual details in a structured way.

Let f_v denote a video encoder that maps spatiotemporal patches into feature vectors. For a frame index t and spatial coordinates (x, y), define a local receptive field $R_{t,x,y}$, and let

$$z^v_{t,x,y} = f_v(V_{R_{t,x,y}}) \in \mathbb{R}^{d_v}. \tag{7}$$

Similarly, a text encoder f_t maps the user description q and any auxiliary textual annotations into token-level embeddings,

$$z^t_j = f_t(q_j) \in \mathbb{R}^{d_t}, \tag{8}$$

where q_j is the j-th token or phrase. An image encoder f_i maps example images into patch-level or global style embeddings,

$$z^i_{k,l} = f_i(I_{k,l}) \in \mathbb{R}^{d_i}, \tag{9}$$

where $I_{k,l}$ denotes a patch or region within example image I_k.

To place these features into a shared space, linear projections can be used:

$$\tilde{z}^v_{t,x,y} = W_v z^v_{t,x,y}, \tag{10}$$

$$\tilde{z}^t_j = W_t z^t_j, \tag{11}$$

$$\tilde{z}^i_{k,l} = W_i z^i_{k,l}, \tag{12}$$

Table 3: Similarity scores and energy terms used for multimodal alignment and regularization.

Quantity	Definition	Functional role
Text–video score $s_{j,t,x,y}$	$s_{j,t,x,y} = \frac{\tilde{z}_j^t \cdot \tilde{z}_{t,x,y}^v}{\|\tilde{z}_j^t\| \, \|\tilde{z}_{t,x,y}^v\|}$	Anchors phrases to spatiotemporal regions
Style assignment $\pi_k(t,x,y)$	Softmax over $\tilde{z}_{t,x,y}^v \cdot \tilde{z}_k^i$	Weights influence of each example image
Local style control $\psi_{t,x,y}$	$\psi_{t,x,y} = \sum_k \pi_k(t,x,y)\phi_k$	Aggregated style parameters per location
Alignment energy E_{align}	$-\sum_{j,t,x,y} w_{j,t,x,y} s_{j,t,x,y}$	Encourages alignment of text and video
Smoothness energy E_{smooth}	$\lambda \sum_{(p,q)\in\mathcal{N}} \|\psi_p - \psi_q\|^2$	Enforces spatial and temporal coherence
Total energy $E(\Theta)$	$E_{\text{align}} + E_{\text{smooth}}$	Defines distribution over assignments

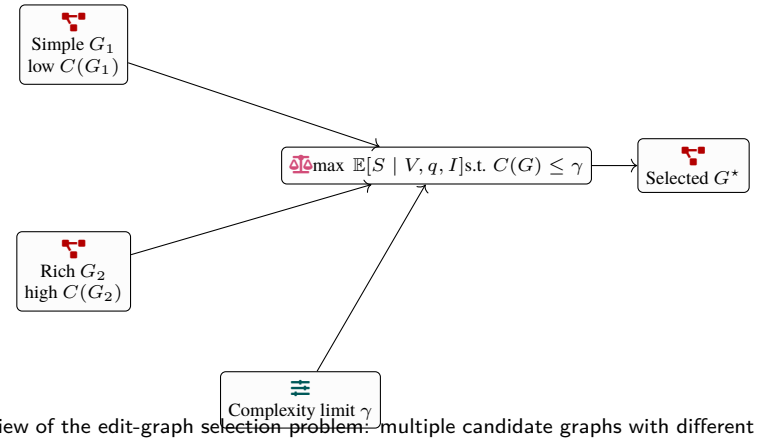

Figure 7: Optimization view of the edit-graph selection problem: multiple candidate graphs with different complexities are evaluated against an expected satisfaction objective under a user- or interface-specific complexity limit, yielding a selected graph $G^\star$ that balances expressivity and control.

where W_v, W_t, W_i are matrices in $\mathbb{R}^{d\times d_v}$, $\mathbb{R}^{d\times d_t}$, and $\mathbb{R}^{d\times d_i}$, respectively. The shared space $\mathbb{R}^d$ allows inner products or cosine similarities to be interpreted across modalities. For example, a correspondence score between a textual token j and a video location (t,x,y) can be defined as

$$s_{j,t,x,y} = \frac{\tilde{z}_j^t \cdot \tilde{z}_{t,x,y}^v}{\|\tilde{z}_j^t\| \, \|\tilde{z}_{t,x,y}^v\|}. \tag{13}$$

Such scores can seed spatiotemporal masks that localize objects or regions referenced in the description [12].

Example images often serve to specify style rather than object identity. A typical approach is to derive a style vector u_k for each example image by aggregating activations in a convolutional feature space. Let g be a function that computes Gram matrices or other statistical summaries, resulting in

$$u_k = g(I_k) \in \mathbb{R}^m. \tag{14}$$

These style vectors can be embedded into the same control space as edit parameters through an affine mapping,

$$\phi_k = Au_k + b, \tag{15}$$

with $A \in \mathbb{R}^{r\times m}$ and $b \in \mathbb{R}^r$. The vector ϕ_k can then be used to parameterize operations such as color transforms or texture synthesis.

Table 4: Representative attachment points for natural language within the video editing interface.

Attachment level	Example user description	Scope in the video	System behavior
Global project	"Give the whole film a cooler, evening tone."	All scenes and shots	Derives global color grading parameters
Timeline segment	"Make this street scene more cinematic and slightly darker."	Selected interval on timeline	Applies style to frames within marked segment
Spatial region on frame	"Boost contrast on the main character only."	Masks around detected subject	Builds foreground enhancement subgraph
Event-conditioned region	"Blur the background when the interviewer walks in."	Frames where condition holds	Adds conditional effect triggered by detections
Grouped macro edit	"Save this look as my interview preset."	Reusable across projects	Packages subgraph as named macro

Table 5: Illustrative editing acts produced by intent parsing from natural language descriptions.

Act ID	Action a_i	Target t_i	Condition c_i
e_1	Color grade	Global scene appearance	None
e_2	Blur / defocus	Background region	Subject visible
e_3	Brightness adjustment	Faces and skin tones	"Keep faces bright"
e_4	Motion emphasis	Moving object (e.g., car)	When turning or accelerating
e_5	Compositing / overlay	Foreground subject vs. new background	When a specified event occurs

The relation between styles and video regions can be modeled probabilistically. For each region (t, x, y), define a categorical distribution over example images,

$$\pi_k(t,x,y) = \frac{\exp\big(\beta\, \tilde{z}^v_{t,x,y} \cdot \tilde{z}^i_k\big)}{\sum_{k'} \exp\big(\beta\, \tilde{z}^v_{t,x,y} \cdot \tilde{z}^i_{k'}\big)}, \tag{16}$$

where β controls sharpness and $\tilde{z}^i_k$ is a global embedding of image I_k. A soft assignment of styles to regions can then be constructed as

$$\psi_{t,x,y} = \sum_k \pi_k(t,x,y)\, \phi_k. \tag{17}$$

The vector $\psi_{t,x,y}$ summarizes how the set of example images should influence the transformation applied to location (t, x, y).

Beyond per-location features, temporal dynamics play an important role [13]. For each region, the trajectory of features over time can capture motion patterns and evolving context. Let

$$h_{x,y} = \big(\tilde{z}^v_{1,x,y}, \ldots, \tilde{z}^v_{T,x,y}\big) \tag{18}$$

denote the temporal sequence of embeddings. Applying a recurrent or attention-based encoder f_τ yields a temporal summary,

$$r_{x,y} = f_\tau(h_{x,y}) \in \mathbb{R}^d. \tag{19}$$

This representation can help interpret temporal modifiers in language, such as "when the car turns left" or "just before the cut". A compatibility score between a temporal

Table 6: Loss terms used in edit synthesis for parameter estimation over spatiotemporal operations.

Loss term	Mathematical form (schematic)	Purpose in optimization
Style loss L_{style}	$\sum_{(t,x,y)\in S_i} \|F(\hat{V}_{t,x,y}) - U_k\|^2$	Matches edited regions to example styles
Smoothness loss L_{smooth}	$\mu \sum_{(p,q)\in\mathcal{N}_i} \|\theta_{i,p} - \theta_{i,q}\|^2$	Promotes spatial and temporal coherence
Consistency loss L_{consist}	Task-specific penalties (e.g., exposure, skin)	Preserves important content properties
Global objective $L(\Theta)$	$\sum_{i=1}^{N} L_i(\theta_i)$	Jointly optimizes all operations
Gradient $\nabla_\theta L$	Combined derivative of all terms	Enables gradient-based parameter updates

Table 7: Key evaluation dimensions for human-centered, language-driven video editing systems.

Dimension	Example metrics	Typical data source
Usability and workload	Task completion time, number of steps, subjective workload scores	Controlled user studies
Expressivity of edits	Fraction of tasks successfully completed, coverage of edit taxonomy	Scenario-based evaluations
Edit quality	Aesthetic ratings, artifact scores (flicker, boundaries, exposure)	Expert or crowd assessments
Predictability and control	Gap between expected and actual results, trust ratings	User predictions and surveys
Robustness to failure	Time to recover, number of corrections, abandonment rate	Interaction logs and stress tests

phrase embedding $\tilde{z}_j^t$ and region trajectory $r_{x,y}$ can be computed using a bilinear form,

$$c_{j,x,y} = \tilde{z}_j^{t\top} B\, r_{x,y}, \tag{20}$$

where $B \in \mathbb{R}^{d\times d}$ is a learned matrix.

Multimodal alignment can be cast as minimizing an energy functional that measures discrepancy between textual descriptions and the induced structure over video and styles. Consider an energy

$$E(\Theta) = E_{\text{align}}(\Theta) + E_{\text{smooth}}(\Theta), \tag{21}$$

where Θ collects all alignment parameters. The term E_{align} penalizes mismatches between textual phrases and selected regions or styles, while E_{smooth} enforces spatial and temporal coherence of assignments. In a simple formulation, [14]

$$E_{\text{align}}(\Theta) = -\sum_{j} \sum_{t,x,y} w_{j,t,x,y}\, s_{j,t,x,y}, \tag{22}$$

where $w_{j,t,x,y}$ are assignment weights. A discrete constraint such as

$$\sum_{t,x,y} w_{j,t,x,y} = 1 \tag{23}$$

for each phrase j would ensure that each phrase is anchored to at least one location, though in practice soft assignments are more appropriate.

The smoothness term can be defined as a quadratic form over neighboring assignments:

$$E_{\text{smooth}}(\Theta) = \lambda \sum_{(p,q)\in\mathcal{N}} \|\psi_p - \psi_q\|^2, \tag{24}$$

Table 8: Selected trade-offs in the joint design of models and interfaces for video editing.

Axis	High end	Low end	Implications
Model expressivity	Rich, complex edit graphs	Simple, constrained recipes	Balance between coverage and inspectability
Interface transparency	Detailed masks and graph views	Minimal exposure of internal state	Affects predictability and learning curve
User effort per edit	Many fine-grained descriptions and corrections	Single global prompts	Trade-off between precision and speed
Automation level	System proposes full edits	System offers local suggestions only	Impacts sense of control and authorship
Personalization strength	Strong adaptation to user history	Generic, one-size-fits-all behavior	Influences consistency and transferability

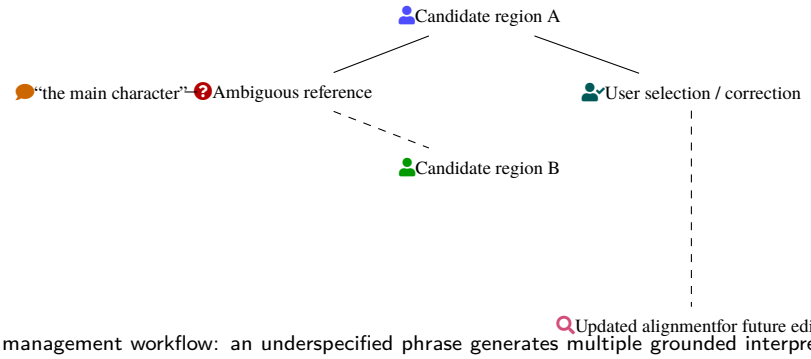

Figure 8: Ambiguity management workflow: an underspecified phrase generates multiple grounded interpretations in the video; the interface exposes these candidates, collects lightweight user feedback, and feeds updated alignment information back into subsequent language-to-edit mappings.

where $\mathcal{N}$ indexes neighboring space-time locations and λ controls regularization strength. This term favors slowly varying style and operation parameters, reflecting the expectation that many edits change gradually rather than abruptly.

A probabilistic interpretation connects this energy to a distribution

$$p(\Theta \mid V, q, I) \propto \exp(-E(\Theta)), \tag{25}$$

supports sampling or approximate inference to propose candidate assignments. These assignments inform both the construction of spatiotemporal masks and the parameterization of operations in the edit graph [15].

The representation layer thus mediates between raw inputs and higher-level edit structures. It provides the quantities needed for the interface to visualize which regions are associated with which phrases, how example images influence different parts of the video, and how style assignments vary over time. At the same time, it supplies the variables over which algorithms operate when constructing and optimizing edit plans. The next section turns to the design of human-centered interfaces that expose these representations and support collaborative interpretation of user intent.

4. Human-Centered Interface Design with Natural Language and Example Images

Human-centered interfaces for language-driven video editing must address several intertwined challenges. Users need to express complex intentions without learning a new programming language, understand how their descriptions

are being interpreted, and iteratively refine both descriptions and results [16]. For many users, natural language is the most accessible channel for specifying goals, while example images help communicate visual properties that are cumbersome to describe in words. The interface must orchestrate these modalities so that users can see how different parts of their input map to specific edits, and how those edits modify space and time within the video.

One central design choice concerns how and where natural language is attached to video structure. A straightforward approach places a text field at the global level, allowing users to describe changes to the video as a whole. However, complex edits often apply to specific segments or events. An interface can therefore integrate language directly into the timeline, allowing the user to attach descriptions to regions corresponding to shots, scenes, or user-defined intervals. When a user highlights a temporal segment and enters a description such as "make the city more cinematic and slightly darker", the system can treat this as an instruction whose scope is restricted to that interval [17]. Visual cues, such as labels above the timeline, can help establish this connection between text and time.

Spatial specificity can be handled by allowing users to pair language with selections on representative frames. The interface might present a key frame for each segment and allow users to draw rough strokes, bounding boxes, or lasso shapes while annotating them with phrases like "subject", "background skyline", or "reflection in the window". Example images can be attached to these annotations, indicating desired styles for the corresponding regions. In this way, the interface collects a structured description: a set of labeled regions and segments, each enriched with both textual and visual examples. The underlying system then uses the multimodal representation to generalize from these cues to the full spatiotemporal volume of the video.

Because learned models inevitably introduce uncertainty, the interface must help users understand and manage ambiguity [18]. When the system interprets a phrase like "the main character" in a crowded scene, it may produce multiple candidate region assignments. Instead of hiding this uncertainty, the interface can visualize alternative interpretations, for example by overlaying color-coded masks corresponding to different potential foreground subjects. Users can then select the interpretation that matches their intent or refine their description to break ties. For example, they might extend the text to "the woman in the red jacket closest to the camera". Through such feedback, the system can update internal alignment scores and refine future proposals.

Example images play multiple roles. They not only specify style but also provide concrete visual references that can be aligned with regions in the video [19]. An interface may present a gallery area where users drag example images and drop them onto specific timeline segments or spatial regions. When an example image is dropped onto a region labeled "background skyline", the system can interpret this as a request to transfer style from that example to the corresponding region across time. The interface can display previews showing how the style is expected to propagate through neighboring frames, exposing the temporal smoothness assumptions encoded in the underlying representation. If users find that style transfer bleeds into undesired areas or times, they can adjust the spatial or temporal scope through handles on the timeline or overlays on the frame.

Support for iteration is another central aspect of human-centered design. Complex edits may require multiple cycles of description, preview, and adjustment. The interface can treat previous descriptions and example assignments as part of a history that users can revise [20]. For instance, after viewing a preview, a user might realize that "slightly darker" was interpreted more strongly than intended. Rather than adjusting numerical sliders, they can refine the text to "only a little darker, keep faces bright" and optionally supply an example image illustrating acceptable brightness. The system should respond by updating parameters associated with relevant operations, ideally in a way that preserves other aspects of the edit graph structure that the user has already validated.

To make the mapping from descriptions to operations understandable, the interface can provide explanations that connect natural language to specific nodes in the edit graph. When a user selects a textual annotation, the interface could highlight the operations and masks influenced by that phrase. Conversely, selecting an operation or region could reveal which textual and visual inputs contributed to its creation. Such bidirectional linking supports a mental model in which language and example images are not opaque triggers but rather components in a causal chain that leads to concrete changes in video appearance [21].

Managing complexity requires mechanisms for abstraction and grouping. As users build more elaborate edits, the number of operations and annotations may grow, risking overwhelming visual clutter. Interfaces can help users create higher-level groups that encapsulate related operations and their controlling descriptions. For example, all operations that implement "evening mood in outdoor scenes" can be grouped into a single macro, with a

human-readable label. Internally, this macro corresponds to a subgraph within the edit graph, but the user interacts with it at a conceptual level. Natural language can be used to create such groups, such as “combine these changes into a reusable preset for daytime interviews”. Example images associated with groups can act as style anchors reused across projects [22].

Another design consideration is the balance between direct manipulation and indirect control. Some users will prefer a workflow dominated by text and example images, while others may desire the ability to inspect and adjust exact parameter values. A human-centered interface can support a hybrid approach. Once the model has proposed an edit, corresponding sliders, curves, and masks can become visible, giving users the option to refine parameters directly. Importantly, changes in either modality should remain synchronized: adjusting a slider should update the textual summary of the effect, and modifying text should adjust sliders in a way that remains coherent. This two-way linkage requires a mapping from parameter changes back into natural language, at least in a coarse descriptive form.

Finally, interfaces can leverage lightweight forms of user feedback to improve future interpretations [23]. Each time a user accepts, rejects, or significantly revises a proposed edit, the system can treat this as a signal about the correctness of its internal mapping between descriptions and operations. Over time, a user-specific model can be learned that predicts preferences, such as typical strength of effects when the user writes “slightly” versus “strongly”, or common compositions implied by phrases like “cinematic look”. The interface can expose controls over personalization, allowing users to reset or export their preference models. Conceptually, this personalization adjusts the distributions over edit graphs conditioned on language and images, shifting probability mass toward patterns that better match the user’s editing style.

In sum, human-centered interface design for complex video editing with natural language and example images revolves around making the internal structure of edits visible and malleable, managing ambiguity collaboratively, and supporting iterative refinement. The next section focuses on the algorithmic side of this process, formalizing how interface-level descriptions can be translated into structured, parameterized edit operations.

5. Algorithms for Mapping Descriptions to Spatiotemporal Edits

Given a video V, a natural language description q, and a set of example images I, the system must construct an edit graph G and associated parameters that, when applied to V, yield an edited video $\hat{V}$ consistent with user intent. This section outlines an algorithmic perspective on this mapping, combining ideas from structured prediction, probabilistic modeling, and numerical optimization [24].

A useful conceptual decomposition separates three stages: intent parsing, visual grounding, and edit synthesis. Intent parsing maps the free-form language q into an intermediate symbolic or semi-structured representation. Visual grounding aligns this representation with spatiotemporal regions of the video and with example images. Edit synthesis then constructs and parameterizes a graph of operations consistent with this alignment and the constraints of the editing model.

Intent parsing can be approximated by a model that predicts a set of editing acts, each consisting of an action, a target, and optional conditions. For example, the phrase “soften the background whenever the interviewer enters the frame” might yield an act with action “blur”, target “background”, and condition “interviewer visible”. Formally, one can define a set $\mathcal{A}$ of possible actions, a set $\mathcal{T}$ of target types, and a set $\mathcal{C}$ of conditions. The parsing model outputs a set [25]

$$\mathcal{E} = \{(a_i, t_i, c_i)\}_{i=1}^{N}, \tag{26}$$

where $a_i \in \mathcal{A}$, $t_i \in \mathcal{T}$, and $c_i \in \mathcal{C} \cup \{\varnothing\}$. This can be framed as a sequence labeling or span classification task over tokens in q, followed by aggregation of contiguous labeled spans. The probability of a particular set of acts might be expressed as

$$p(\mathcal{E} \mid q) = \prod_{i=1}^{N} p(a_i, t_i, c_i \mid q). \tag{27}$$

Visual grounding assigns these acts to specific video locations using the multimodal representation defined earlier. For each act i, the target type t_i implies a distribution over potential masks. If t_i refers to an object, grounding involves object detection and tracking; if it refers to a region like “background”, grounding may involve segmentation layers. Let $\mathcal{M}$ denote the space of masks. The grounding step yields a set of masks $\{m_i\}$, where each $m_i \in \mathcal{M}$

is a function on space-time. A probabilistic formulation models

$$p(m_i \mid V, q, I, a_i, t_i, c_i), \tag{28}$$

which may depend both on visual evidence and on conditions. For conditions involving events, such as “whenever the interviewer enters”, temporal intervals can be derived from detection of the relevant entity, resulting in a support set over t.

Edit synthesis constructs operations o_i corresponding to acts and assigns parameters θ_i. The edited video can be represented as [26]

$$\hat{V}_t = V_t + \sum_{i=1}^{N} m_{i,t} \odot \Delta_{i,t}, \tag{29}$$

where $m_{i,t}$ is the mask of act i at time t, $\odot$ denotes elementwise multiplication, and $\Delta_{i,t}$ is the change induced by operation i. In a simple linear model, one might have

$$\Delta_{i,t} = P_{i,t}\, V_t + b_{i,t}, \tag{30}$$

where $P_{i,t}$ is a matrix defining a local linear transformation and $b_{i,t}$ is an offset. These parameters can vary over time but are usually regularized for smoothness.

The mapping from example images to parameters can be formalized via style vectors. Suppose each act i is associated with a style control vector $\psi_{i,t,x,y}$, derived as in the previous section. Parameters can be modeled as functions of style:

$$\theta_{i,t,x,y} = h_{a_i}(\psi_{i,t,x,y}), \tag{31}$$

where h_{a_i} is an action-specific function. For color grading actions, h_{a_i} might map style vectors to parameters of a parametric tone mapping curve. For blur actions, it might map to kernel widths and shapes.

Parameter estimation can be posed as a numerical optimization problem. For each act i, one can define a loss functional measuring deviation from desired style relative to example images, subject to smoothness and plausibility constraints [27]. Let S_i denote the set of space-time points covered by m_i. One can define

$$L_i(\theta_i) = L_{\text{style}}(\theta_i) + L_{\text{smooth}}(\theta_i) + L_{\text{consist}}(\theta_i), \tag{32}$$

where L_{style} measures mismatch between the edited video and example styles, L_{smooth} penalizes rapid temporal or spatial changes in parameters, and L_{consist} penalizes inconsistency with constraints such as preserving skin tones or maintaining exposure within certain bounds.

An example style loss can be expressed in a feature space. Let F extract features from the edited video and U_k denote features from example image I_k. For a region associated with example k,

$$L_{\text{style}}(\theta_i) = \sum_{(t,x,y)\in S_i} \left\| F(\hat{V}_{t,x,y}) - U_k \right\|^2. \tag{33}$$

Smoothness can be enforced by penalizing differences between neighboring parameter values,

$$L_{\text{smooth}}(\theta_i) = \mu \sum_{(p,q)\in \mathcal{N}_i} \|\theta_{i,p} - \theta_{i,q}\|^2, \tag{34}$$

with $\mathcal{N}_i$ denoting neighbors within the region covered by act i and μ a regularization weight.

The global objective across all acts is

$$L(\Theta) = \sum_{i=1}^{N} L_i(\theta_i), \tag{35}$$

with Θ collecting all parameters. Minimization of $L(\Theta)$ can be performed by gradient-based methods [28]. The gradient of L with respect to a parameter vector $\theta_{i,t,x,y}$ is

$$\nabla_{\theta_{i,t,x,y}} L = \nabla_{\theta_{i,t,x,y}} L_{\text{style}} + \nabla_{\theta_{i,t,x,y}} L_{\text{smooth}} + \nabla_{\theta_{i,t,x,y}} L_{\text{consist}}, \tag{36}$$

which can be computed via backpropagation through the rendering pipeline from parameters to pixels.

The discrete structure of the edit graph G introduces combinatorial complexity. Each act could, in principle, be implemented by multiple possible sequences of operations. A graph search perspective considers the space $\mathcal{G}$ of possible graphs consistent with the set of acts $\mathcal{E}$. A scoring function

$$R(G; V, q, I) = \mathbb{E}[S \mid G, V, q, I] - \lambda C(G) \tag{37}$$

combines predicted user satisfaction with a complexity penalty. Exact maximization over $\mathcal{G}$ is typically infeasible, so approximate strategies such as beam search or stochastic sampling are required.

One strategy is to build graphs incrementally. Starting from the raw video, the algorithm selects an act i and proposes an operation or set of operations that implement it, according to a learned policy [29]. The state at step k consists of the partial graph G_k. A policy

$$\pi(o \mid G_k, V, q, I) \tag{38}$$

selects the next operation o, which is added to the graph to form G_{k+1}. This process continues until a termination condition is met. The policy can be trained using reinforcement learning with a reward signal linked to predicted satisfaction and penalized complexity. The return of a graph trajectory can be written as

$$J(\pi) = \mathbb{E}[R(G_K; V, q, I)], \tag{39}$$

where the expectation is over trajectories induced by policy π.

From a probabilistic perspective, the combination of intent parsing, visual grounding, and graph construction can be framed as inference in a structured model [30]. Let H denote latent variables encoding alignments and graph choices. The posterior distribution

$$p(H, G \mid V, q, I) \tag{40}$$

encodes uncertainty about both alignments and graph structure. Approximate inference methods such as variational inference or Markov chain Monte Carlo can, in principle, explore this space, but computational constraints require simplifications. For example, one can factorize the posterior into a grounding component and a graph component, solving grounding first and then treating it as fixed when constructing the graph.

Human-centered considerations influence these algorithmic choices. For instance, a model that commits early to single-point estimates of masks or graph structure may be efficient but will struggle to expose alternative interpretations to users [31]. A model that maintains multiple hypotheses can better support user selection among alternatives, at the cost of increased complexity. Algorithms must therefore be designed to produce a small number of diverse, high-scoring candidate graphs, each with associated visualizations that the interface can present as options.

Finally, the mapping from descriptions to edits can adapt over time based on user feedback. Let $y_n \in \{0, 1\}$ indicate whether the user accepted the n-th proposed edit, and let x_n denote features summarizing the description, video context, and graph structure. A simple statistical model for acceptance could be a logistic regression

$$p(y_n = 1 \mid x_n) = \sigma(w^\top x_n), \tag{41}$$

where σ is the logistic function and w is a parameter vector. Online updates to w based on observed feedback provide a mechanism for personalization and gradual alignment between the system's predictions and the user's preferences. More complex models can incorporate temporal dependencies across editing sessions, but even a basic model can bias the search over graphs toward regions of the space that have historically produced satisfactory edits for the user [32].

In combination, these algorithmic components define a mapping from user-facing descriptions to internal edit structures that can be optimized, evaluated, and explained. The next section discusses how such systems might be evaluated and what trade-offs emerge when balancing accuracy, transparency, and user effort.

6. Evaluation and Discussion

Evaluating human-centered interfaces for complex video editing with natural language and example images requires attention to multiple dimensions: usability, expressivity, quality of edits, and predictability of system behavior. Unlike purely algorithmic tasks where performance can be measured against objective ground truth labels, editing systems must be assessed in relation to users' subjective goals and workflows. This section sketches possible evaluation approaches and reflects on their implications.

One dimension concerns how effectively users can translate conceptual editing goals into interface actions. Usability studies can investigate metrics such as task completion time, number of interaction steps, and subjective workload ratings when performing editing tasks with different interface conditions [33]. For instance, one condition might involve a baseline timeline-and-slider interface, another might add natural language descriptions, and a third might combine language with example images and the proposed multimodal mapping. Participants could be asked to reproduce target edits described in textual briefs or to achieve self-defined goals on provided footage. The comparison would reveal whether the additional expressivity afforded by language and images reduces or increases cognitive load.

Expressivity relates to the range of edits users can describe and successfully realize using the interface. One way to approximate this is to construct a taxonomy of edit types, such as global color transforms, localized adjustments, temporal rearrangements, and compositing operations, and then sample tasks representative of each category. Users can attempt these tasks using the system, and the degree of success can be judged by expert evaluators or by the users themselves. The fraction of tasks for which users feel they achieved an acceptable result offers a coarse measure of expressivity [34]. More detailed analyses can characterize which types of edits are well supported by

language and example images and which still require low-level controls.

The quality of edited videos can be assessed along aesthetic and technical dimensions. Aesthetic quality includes perceived cohesiveness of style, appropriateness of effects, and absence of distracting artifacts. Technical quality encompasses issues such as temporal flicker, boundary artifacts around masked regions, and preservation of important details like faces. Observers, either experts or crowd workers, can rate edited videos on these criteria without necessarily knowing which interface produced them. Statistical models can relate these ratings to properties of the underlying edit graphs, such as complexity, number of operations, and degree of temporal smoothing, offering insight into how algorithmic choices influence perceived quality.

Predictability is particularly important in systems involving learned models and high-level inputs [35]. Users benefit when they can form accurate expectations about how the system will interpret their descriptions and respond to changes. To probe predictability, studies can ask users to anticipate results before viewing them. For example, after entering a description and selecting example images, users can be asked to sketch or verbally describe what they expect the edited video to look like. The discrepancy between expected and actual edits, as judged by third parties, provides a measure of predictability. Interfaces that expose intermediate representations, such as masks and style assignments, may improve predictability by helping users understand the chain of transformations.

From a statistical perspective, data gathered from such studies can be analyzed using models that account for repeated measures and individual differences. Suppose each trial n involves a participant u_n, a task t_n, an interface condition c_n, and produces an outcome measure r_n, such as a rating of success on a numerical scale. One might fit a mixed-effects model of the form [36]

$$r_n = \alpha + \beta_{c_n} + b_{u_n} + \epsilon_n, \tag{42}$$

where α is a global intercept, β_c captures fixed effects of interface condition c, b_{u_n} is a participant-specific random effect, and ϵ_n is residual noise. Estimates of β_c and their uncertainties indicate whether differences between interface designs are likely to generalize beyond the sampled participants.

Another line of evaluation involves log analysis of real-world usage, when available. Interaction logs can record sequences of descriptions, example image selections, preview requests, and accept or undo operations. Descriptive statistics on these logs can reveal patterns such as how often users rely on language versus low-level controls, how many iterations typical edits require, and where in the workflow users tend to abandon tasks. From a probabilistic modeling perspective, these sequences can be viewed as trajectories in a state space, and models similar to Markov decision processes can be applied to identify common paths and bottlenecks.

Limitations and trade-offs emerge naturally from these evaluations. For example, interfaces that encourage users to specify many fine-grained descriptions and example associations may offer high expressivity but impose greater cognitive and interaction costs. Conversely, systems that aggressively compress descriptions into simple recipe-like graphs may be easier to operate but less capable of capturing nuanced intent [37]. Transparency features, such as exposing edit graphs and alignment maps, can improve understanding but risk overwhelming users with technical detail. Evaluation can help quantify these trade-offs by varying the degree of transparency and measuring its effects on performance and satisfaction.

An important discussion point is the treatment of failure cases. Learned models will inevitably misinterpret some descriptions or propagate style in undesired ways. Human-centered interfaces should be judged not only by their performance when things go well but also by how they support recovery when things go wrong. For example, if the system misidentifies the subject in a scene, the interface should make this misinterpretation visible and easy to correct, perhaps by allowing users to reassign labels or adjust masks with minimal effort. Evaluation tasks can intentionally include videos and descriptions likely to induce failure, observing how users detect and respond to these issues [38].

Another aspect is generalization across users with different levels of expertise. Professional editors may be accustomed to fine-grained control and may prefer interfaces that expose detailed graph structures and parameter curves, while novices may find such representations intimidating. One approach is to design adaptive interfaces that reveal more detail as users demonstrate interest or proficiency, guided by models that estimate user expertise based on interaction patterns. Evaluation can consider how well such adaptive strategies serve different user groups, for example by measuring learning curves over multiple sessions.

The integration of algorithms and interface design also raises questions about system boundaries. Some editing tasks may be better handled entirely by automated suggestions followed by minimal user confirmation, while

others require more deliberate, collaborative interaction. Evaluation can compare workflows in which the system proposes complete edits from high-level descriptions against workflows in which users construct edits incrementally, with the system providing localized assistance [39]. Outcomes may differ not only in efficiency and quality but also in users' sense of control and authorship.

Finally, reflections on ethical and societal implications are relevant. Systems that lower barriers to sophisticated video editing may enable more people to express themselves creatively but may also facilitate misrepresentation or manipulation of video content. While these concerns extend beyond the scope of interface design alone, they intersect with decisions about transparency and traceability. For example, interfaces might support provenance metadata indicating which edits were driven by automated interpretations of natural language and example images. Evaluation could include user understanding of such metadata and its impact on trust.

Overall, evaluation and discussion suggest that success for human-centered, language-driven video editing systems should be measured not only by technical performance but also by how they integrate into users' workflows, how predictably they behave, and how they support learning, exploration, and responsible use [40]. The conclusion summarizes the core ideas and outlines potential directions for further work.

7. Conclusion

This paper has explored a human-centered perspective on interfaces that allow complex video edits to be described using natural language and example images. Starting from a characterization of contemporary video editing practice, the discussion framed the challenge as a mapping from high-level, multimodal descriptions to structured edit graphs acting on spatiotemporal video tensors. This mapping was formalized in terms of a distribution over edit graphs conditioned on video, language, and images, with user satisfaction modeled as an underlying objective moderated by constraints on complexity, transparency, and editability.

At the representational level, the paper described how video segments, textual phrases, and example images can be embedded into a shared space that supports semantic alignment and style transfer. Linear projections and similarity measures link tokens to spatiotemporal regions, while style vectors derived from example images parameterize operations such as color adjustments and texture modifications. Temporal encodings capture evolving context, enabling modifiers like "when" or "before" to influence the scope of edits [41]. Regularization terms encourage spatial and temporal coherence, reflecting expectations about smooth variation in real video edits.

Human-centered interface design was presented as a key mediator between these representations and end users. Interfaces can attach natural language descriptions to timeline segments and spatial regions, pair them with example images, and expose alternative interpretations of ambiguous phrases. By making internal structures such as masks and style assignments visible and editable, the interface allows users to understand and refine how their descriptions are translated into operations. Support for iterative workflows, bidirectional linking between language and operations, and mechanisms for abstraction and grouping all contribute to managing the complexity of real editing tasks.

On the algorithmic side, the mapping from descriptions to edits was decomposed into intent parsing, visual grounding, and edit synthesis. Intent parsing identifies editing acts from natural language, visual grounding aligns these acts with video regions and example images, and edit synthesis constructs and parameterizes operations that implement them [42]. The process was expressed using probabilistic models over alignments and edit graphs, along with optimization-based parameter estimation leveraging multivariate calculus and numerical analysis. Structured prediction and reinforcement learning perspectives highlighted the combinatorial nature of graph construction and the role of approximate search. Statistical models incorporating user feedback suggested pathways for personalization and adaptation over time.

Evaluation considerations emphasized that such systems should be judged along multiple dimensions, including usability, expressivity, edit quality, and predictability. Potential methodologies included controlled user studies comparing interface conditions, expert or crowd-based assessment of edited videos, and log analysis of real-world usage. Mixed-effects models and other statistical tools can help disentangle effects of interface design, user expertise, and task type. The importance of handling failure cases, supporting recovery, and accommodating diverse user populations was underscored, along with the relevance of transparency and provenance in light of broader concerns about video manipulation [43].

The ideas presented here suggest several avenues for further work. One direction involves tighter coupling between model architectures and interface affordances, designing learned representations that are oriented from the outset toward interpretability in the editing context.

Another direction involves richer forms of user feedback, such as sketch-based corrections or demonstrations, which can complement natural language and example images. Longitudinal studies can shed light on how users' mental models of these systems evolve over time and how interfaces might adapt in response. Finally, collaboration between researchers in machine learning, human-computer interaction, and video production practice appears important for grounding technical advances in realistic workflows.

By treating natural language and example images as central elements in both algorithmic design and interaction design, the paper has argued that complex video editing can become more accessible while retaining a structured representation that supports control and understanding. Achieving this balance requires careful integration of mathematical models, multimodal representations, and human-centered interfaces, with attention to how each design choice shapes the way users express intent, interpret system behavior, and ultimately construct the edited videos they have in mind [44].

References

[1] V. D. Huszár and V. K. Adhikarla, "Live spoofing detection for automatic human activity recognition applications.," *Sensors (Basel, Switzerland)*, vol. 21, pp. 7339–, 11 2021.

[2] M. Terras, "Another suitcase, another student hall — where are we going to? what ach/allc 2001 can tell us about the current direction of humanities computing," *Literary and Linguistic Computing*, vol. 16, pp. 485–491, 11 2001.

[3] Y. Yi, J. Dai, C. Wang, J. Hou, H. Zhang, L. Yunlong, and G. Jin, "An effective framework using spatial correlation and extreme learning machine for moving cast shadow detection," *Applied Sciences*, vol. 9, pp. 5042–, 11 2019.

[4] R. Sawaki, D. Sato, H. Nakayama, Y. Nakagawa, and Y. Shimada, "Zf-automl: An easy machine-learning-based method to detect anomalies in fluorescent-labelled zebrafish," *Inventions*, vol. 4, pp. 72–, 12 2019.

[5] A. A. Egorova and A. P. Ryjov, "Generative intelligence systems for image synthesis: Scenarios for their use and related tasks," *Moscow University Computational Mathematics and Cybernetics*, vol. 48, pp. 45–58, 4 2024.

[6] A. Kaushal, S. Kumar, and R. Kumar, "A review on deepfake generation and detection: bibliometric analysis," *Multimedia Tools and Applications*, vol. 83, pp. 87579–87619, 3 2024.

[7] S. S. Harsha, D. Agarwal, A. Revanur, and S. Agrawal, "Digital video editing based on a target digital image," Aug. 21 2025. US Patent App. 18/583,067.

[8] K. A. L. Hernandez, T. M. Rienmüller, D. Baumgartner, and C. Baumgartner, "Deep learning in spatiotemporal cardiac imaging: A review of methodologies and clinical usability.," *Computers in biology and medicine*, vol. 130, pp. 104200–104200, 12 2020.

[9] J. Kim, I. Choi, and M. Lee, "Context aware video caption generation with consecutive differentiable neural computer," *Electronics*, vol. 9, pp. 1162–, 7 2020.

[10] N. Jain, V. Bansal, D. Virmani, V. Gupta, L. Salas-Morera, and L. García-Hernández, "An enhanced deep convolutional neural network for classifying indian classical dance forms," *Applied Sciences*, vol. 11, pp. 6253–, 7 2021.

[11] G. C. Lencioni, R. V. de Sousa, E. J. de Souza Sardinha, R. R. Corrêa, and A. J. Zanella, "Pain assessment in horses using automatic facial expression recognition through deep learning-based modeling.," *PloS one*, vol. 16, pp. e0258672–, 10 2021.

[12] I. P. Maulana, Y. Y. Putra, and I. Rahmawati, "The floating learning media software based on "adobe flash" material nets-build nets build space in class v:," *Cendekiawan*, vol. 3, pp. 1–13, 6 2021.

[13] F. Zhang, J. Zhao, Y. Zhang, and S. Zollmann, "A survey on 360° images and videos in mixed reality: Algorithms and applications," *Journal of Computer Science and Technology*, vol. 38, pp. 473–491, 5 2023.

[14] Y. Banadaki, N. Razaviarab, H. Fekrmandi, G. Li, P. Mensah, S. Bai, and S. Sharifi, "Automated quality and process control for additive manufacturing using deep convolutional neural networks," *Recent Progress in Materials*, vol. 4, pp. 1–19, 2 2022.

[15] J. Guo, P. Sun, and S.-B. Tsai, "A study on the optimization simulation of big data video image

keyframes in motion models," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–12, 3 2022.

[16] F. Martin-Rico, F. Gomez-Donoso, F. Escalona, J. Garcia-Rodriguez, and M. Cazorla, "Semantic visual recognition in a cognitive architecture for social robots," *Integrated Computer-Aided Engineering*, vol. 27, pp. 301–316, 5 2020.

[17] A. Wang, X. Cao, L. Lu, X. Zhou, and X. Sun, "Design of efficient human head statistics system in the large-angle overlooking scene," *Electronics*, vol. 10, pp. 1851–, 7 2021.

[18] J. Maycock, D. Dornbusch, C. Elbrechter, R. Haschke, T. Schack, and H. Ritter, "Approaching manual intelligence," *KI - Künstliche Intelligenz*, vol. 24, pp. 287–294, 8 2010.

[19] F. D. Jordan, M. Kutter, J.-M. Comby, F. Brozzi, and E. Kurtys, "Open and remotely accessible neuroplatform for research in wetware computing.," *Frontiers in artificial intelligence*, vol. 7, pp. 1376042–, 5 2024.

[20] V. Hasija and E. G. Takhounts, "Deep learning model for predicting head kinematics from crash videos," 11 2021.

[21] null null and null null, "Peech signal control software is an important part of our lives.," *Zenodo (CERN European Organization for Nuclear Research)*, 4 2022.

[22] J. Luisi, A. Narayanaswamy, Z. Galbreath, and B. Roysam, "The farsight trace editor: An open source tool for 3-d inspection and efficient pattern analysis aided editing of automated neuronal reconstructions," *Neuroinformatics*, vol. 9, pp. 305–315, 4 2011.

[23] W. Paier, A. Hilsmann, and P. Eisert, "Interactive facial animation with deep neural networks," *IET Computer Vision*, vol. 14, pp. 359–369, 8 2020.

[24] D. Aleksandar, P. Ivan, and N. Mirjana, *Digital Performing Arts - Participatory Practices in a Digital Age*. 3 2023.

[25] L. Zhou, R. Wang, and L. Zhang, "Accurate robot arm attitude estimation based on multi-view images and super-resolution keypoint detection networks.," *Sensors (Basel, Switzerland)*, vol. 24, pp. 305–305, 1 2024.

[26] I. Kwon, J. Li, and M. Prasad, "Lightweight video super-resolution for compressed video," *Electronics*, vol. 12, pp. 660–660, 1 2023.

[27] S. S. Agaian and S. Jassim, "Mobile multimedia/image processing, security, and applications 2016," in *SPIE Proceedings*, vol. 9869, pp. 986901–, SPIE, 9 2016.

[28] U. B. A, "Blinds personal assistant application for android," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, pp. 138–144, 1 2019.

[29] Y. Graham, G. Awad, and A. F. Smeaton, "Evaluation of automatic video captioning using direct assessment," *PloS one*, vol. 13, pp. e0202789–, 9 2018.

[30] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: recent advances and future trends," *Frontiers of Computer Science*, vol. 10, pp. 19–36, 6 2015.

[31] S. Jiao, G. Li, G. Zhang, J. Zhou, and J. Li, "Multimodal fall detection for solitary individuals based on audio-video decision fusion processing.," *Heliyon*, vol. 10, pp. e29596–e29596, 4 2024.

[32] S.-M. Hu and L. Kobbelt, "Preface," *Journal of Computer Science and Technology*, vol. 30, pp. 437–438, 5 2015.

[33] D. A. Souza, P. Dias, D. A. Santos, and B. S. Santos, "Platform for setting up interactive virtual environments," *SPIE Proceedings*, vol. 9012, pp. 187–195, 2 2014.

[34] F. Filicori, D. P. Bitner, H. F. Fuchs, M. Anvari, G. Sankaranaraynan, M. B. Bloom, D. A. Hashimoto, A. Madani, P. Mascagni, C. M. Schlachta, M. Talamini, and O. R. Meireles, "Sages video acquisition framework-analysis of available or recording technologies by the sages ai task force.," *Surgical endoscopy*, vol. 37, pp. 4321–4327, 2 2023.

[35] N. Aslam, K. Xia, F. Rustam, A. Hameed, and I. Ashraf, "Using aspect-level sentiments for calling app recommendation with hybrid deep-learning models," *Applied Sciences*, vol. 12, pp. 8522–8522, 8 2022.

[36] J. Tian, "Artificial intelligence advanced imaging report standardization and intra-interdisciplinary clinical workflow.," *EBioMedicine*, vol. 44, pp. 4–5, 5 2019.

[37] Q. Liu, C. Feng, Z. Song, J. Louis, and J. Zhou, "Deep learning model comparison for vision-based classification of full/empty-load trucks in earthmoving operations," *Applied Sciences*, vol. 9, pp. 4871–, 11 2019.

[38] Z. Wu, T. Pei, Z. Bao, S. T. Ng, G. Lu, and K. Chen, "Utilizing intelligent technologies in construction and demolition waste management: From a systematic review to an implementation framework," *Frontiers of Engineering Management*, vol. 12, pp. 1–23, 6 2024.

[39] S. Armitage, K. Awty-Carroll, D. Clewley, and V. Martinez-Vicente, "Detection and classification of floating plastic litter using a vessel-mounted video camera and deep learning," *Remote Sensing*, vol. 14, pp. 3425–3425, 7 2022.

[40] D. S. Abdelminaam, A. M. Almansori, M. Taha, and E. M. Badr, "A deep facial recognition system using computational intelligent algorithms.," *PloS one*, vol. 15, pp. e0242269–, 12 2020.

[41] K. Kadam, S. Ahirrao, and K. Kotecha, "Multiple image splicing dataset (misd): A dataset for multiple splicing," *Data*, vol. 6, pp. 102–, 9 2021.

[42] S. M. N. Sakib, "Artificial intelligence in marketing," 8 2022.

[43] A. Lazarov and T. Kostadinov, "Bistatic forward isar with dvb-t transmitter of opportunity," *Sensors (Basel, Switzerland)*, vol. 21, pp. 6662–, 10 2021.

[44] S.-K. Zhang, J.-H. Liu, Y. Li, T. Xiong, K.-X. Ren, H. Fu, and S.-H. Zhang, "Automatic generation of commercial scenes," in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1137–1147, ACM, 10 2023.