VideoSticker: A Tool for Active Viewing and Visual Note-taking from Videos

YINING CAO, University of California, San Diego, USA
HARIHARAN SUBRAMONYAM, Stanford University, USA
EYTAN ADAR, University of Michigan, USA

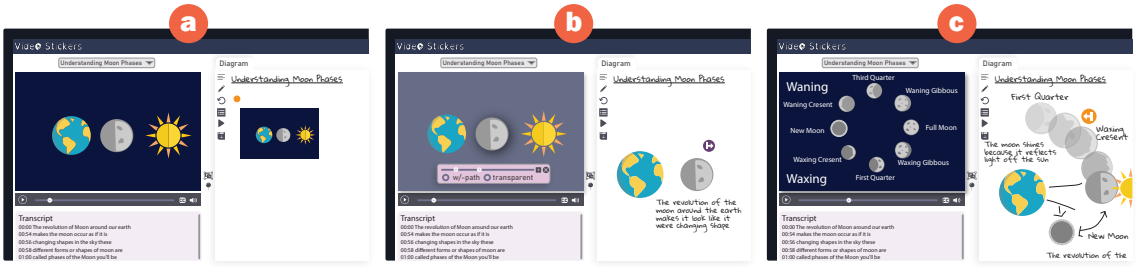

Fig. 1. Note-taking using VideoSticker. While watching the video about Phases of the Moon, the viewer (a) captures the video frame for 'new-moon' as a *frame sticker*. From this captured frame, the viewer (b) extracts the Earth and Moon as *object stickers*, and (c) expands the object sticker to show the motion of the moon around the Earth and annotates these key phases.

Video is an effective medium for knowledge communication and learning. Yet active viewing and note-taking from videos remain a challenge. Specifically, during note-taking, viewers find it difficult to extract essential information such as representation, composition, motion, and interactions of graphical objects and narration. Current approaches rely on creating static screenshots, manual clipping, manual annotation and transcription. This is often done by repeatedly pausing and rewinding the video, thus disrupting the viewing experience. We propose VideoSticker, a tool designed to support visual note-taking by extracting expressive content and narratives from videos as 'object stickers.' VideoSticker implements automated object detection and tracking, linking objects to the transcript, and supporting rapid extraction of stickers across space, time, and events of interest. VideoSticker's two-pass approach allows viewers to capture high-level information uninterrupted and later extract specific details. We demonstrate the usability of VideoSticker for a variety of videos and note-taking needs.

CCS Concepts: • **Human-centered computing** → *Interactive systems and tools*; • **Computing methodologies** → **Computer vision**; • **Applied computing** → **Interactive learning environments**.

Additional Key Words and Phrases: visual note-taking, video object detection, video interaction, education technology

ACM Reference Format:
Yining Cao, Hariharan Subramonyam, and Eytan Adar. 2022. VideoSticker: A Tool for Active Viewing and Visual Note-taking from Videos. In *27th International Conference on Intelligent User Interfaces (IUI '22), March 22–25, 2022, Helsinki, Finland*. ACM, New York, NY, USA, 28 pages. https://doi.org/10.1145/3490099.3511132

1 INTRODUCTION

Video offers an expressive medium for communicating information [66]. By combining graphics, animation, text, and sound, video can convey complex concepts such as change, interactions, spatial and temporal relationships, and causal relationships. For example, viewers can better understand the phenomenon behind 'phases of the moon' by observing changes in the sunlight reflecting off the moon's surface as it revolves around the earth. Such concepts are hard to grasp from text and static graphics alone [48]. The ease and popularity of video production has led to thousands of short-form video content available on open media platforms such as YouTube. This content covers the range of topics from how a virus infects a healthy human cell to techniques for folding dumplings. Video content has high congruence (external representations that match desired internal mental model [66]) and engagement value. Despite this, learning from video is not always effective or easy [60, 63]. As with other learning mediums such as text and static graphics, video has both positives and negatives for learning [8, 60, 69].

Due to the *high-bandwidth* information delivered in a short duration of time, video viewers may only attend to perceptually salient information while omitting thematically relevant details necessary for comprehension [44]. In other words, watching videos can have a high cognitive load [63]. Considering our 'phases of the moon' example, the video toggles between a space-centric view (Figure 1a, animating sunlight reflections off the moon at different orbital positions) and an earth-centric view for teaching about waxing, waning, new-moon, and full-moon (Figure 1c), the narration complements the animation with details such as as: *during a full moon, the earth is between the sun and moon*; *waning gibbous moon rises in the East*; and *the moon's orbital duration is* 27.32 *days*. Unfortunately, with rapid scene switches and the interspersed facts, viewers may not be able to easily learn key facts. For example, they may fail to connect the two 'views' of the moon or learn important names when associating labels and visual elements in each phase. Unlike text, in which active reading and note-taking approaches allow readers to organize and integrate low-level concepts coherently (e.g., texSketch [62]), watching videos is still primarily a passive activity [21]. Further, because video already represents an *integrated* view of concepts, existing active reading strategies do not readily apply. Active viewing requires *decomposition* strategies of graphics, animations, and narration so learners can engage in knowledge construction activities when building notes. In this work, we tackle the problem of note-taking from videos explicitly by introducing a set of techniques for extracting and organizing content from short-form videos.

The interactive elements of existing video players are commonly limited to play, pause, rewind, and random access to the timeline (i.e., manual seek). For taking notes, this makes information extraction and organization particularly difficult. Unlike static images that can be viewed as a 'whole' or compared side-by-side, viewers may have to go through the complete video to understand the context [35]. Further, viewers must remember 'when' they saw something, and a mistake may involve more costly scanning. For example, if the viewer wanted to compare space-centric and earth-centric states of the 'new moon,' they would need to remember or re-find those frames in the video and then jump back and forth or take screenshots. Existing techniques for video navigation (e.g., [22, 52]) and annotation (e.g., [7]) alleviate some of these concerns, but are not specifically intended for note-taking. Due to the video's inherent properties including spatial contiguity and temporal linearity and the limited set of interactions afforded to the viewer, creating visual notes from a video can be difficult. Readers may want to combine linguistic and graphical information, capture object motion in space and time, or compare objects across different frames as part of their note-taking process (i.e., capture the expository purpose of the video [54]). Perhaps worse, with repeated pausing and seeking that is needed for note-taking, viewers may lose many of the benefits of the video format. Therefore, our motivating question for this work is: *"How might we support viewers to effectively take notes from the high-bandwidth information in video content?"*

We propose VideoSticker, a tool for active viewing and visual note-taking from videos (see Figure 1). VideoSticker utilizes object detection, object tracking, and object-narration linking to allow viewers to quickly 'extract' and organize the video content into "semi-animated" graphical notes. As shown in Figure 1 a, the reader can initially add screenshots of salient frames to their notes while watching the video. They can then extract objects in those frames (e.g., earth and moon in Figure 1b) along with transcripts of narration in the form of object (or motion) stickers. Unlike static screenshots, stickers retain the original videos' animation properties (movement in space and time) but are also disentangled from potentially distracting background images. Readers can further 'expand' each video sticker into constituent frames along the motion path and add fine-grain annotations to scaffold their understanding. As shown in Figure 1c, the viewer can expand the space-centric view of the moon along the orbit, label different phases, and add corresponding earth-centric views next to each frame. To support reviewing and re-watching from notes and videos, VideoSticker automatically overlays relevant annotations on top of the original video. Further, as a standalone artifact, the semi-animated notes preserve many of the original advantages of the video. To support these features, we implement computer vision techniques for element extraction from videos. Additionally, because notes include both images and text, we integrate natural language processing approaches to process transcripts and text detected in the video. In combination, this allows viewers to issue expressive queries to generate stickers and rapidly incorporate relevant labels and descriptions to their notes.

VideoSticker supports a wide range of note-taking needs across different *content.* The "input" content can range from education videos, cooking instructions and tutorials, workout routines and many others. This allows for producing novel notes and summaries across *contexts*: teaching science, dynamic recipe cards, analysis of sports events, etc. To support all these, VideoSticker was designed to capture complex narratives through static and dynamic stickers. We contribute specific algorithms for processing video content and novel interactive features to extract object stickers. Finally, we demonstrate the viability of the approach for note-taking through a user study and offer design recommendations for video note-taking tools.

2 RELATED WORK

Video, especially those intended for expository purposes, can be a double-edged sword in educational contexts. Prior research has identified ways in which videos improve student understanding of concepts [34, 46, 47, 57, 58, 61], but also the limitations they pose [19, 20, 66, 71]. Central to these concerns are two key principles necessary for effective video comprehension: (1) the *Congruence Principle* in which "the structure and content of the external representation should correspond to the desired structure and content of the internal representation, [66]"; and (2) the *Apprehension Principle* which requires that "the structure and content of the external representation should be readily and accurately perceived and comprehended. [66]" The former concerns video characteristics such as composition, continuity, and overall aesthetics, while the latter pertains to viewer attributes including attention and cognitive load. Work within education and cognitive psychology have looked at ways to apply both these principles to video comprehension. In terms of video characteristics, prior research has identified effective ways to structure explanations, pauses, spatial contiguity, redundancy etc. [39, 50, 54, 60]. Strategies for viewer's cognitive load and attention includes the use of interactivity [66], and quizzes, etc [46]. Rather than attending to video authoring tools (i.e., video production), our work aims to improve the effectiveness of already existing videos through interactive scaffolding of active viewing and visual note-taking. Particularly, external representations can help viewers in learning through knowledge construction tasks [45, 66]. In this section, we synthesize prior literature on interactive note-taking, video-based interactions, and video-content processing techniques to inform the design of VideoSticker.

2.1 Interactive note-taking tools

Note-taking can be categorized into linear and non-linear processes [49]. Linear note-taking, which is most often used in real-time notes, means extracting key points in the sequence of the received information. In a video context, this may mean creating the notes as the video plays or pausing the video so that note-taking and viewing can be synchronized. Existing work has highlighted the lack of effective note-taking strategies, challenges to transcription, and self-regulation while taking linear notes from videos [9, 16]. In contrast, non-linear note-taking can involve moving between points in the linear material. This approach offers flexibility, and is helpful in organizing and integrating notes, i.e., external representations for knowledge construction [45]. Therefore in VideoSticker we support a non-linear note-taking approach by enhancing non-linear navigation and content extraction. Viewers can create 'rough' notes (e.g., full-frame screenshots and transcript clips) and later revisit them to select frame objects and add accurate textual descriptions. VideoSticker supports viewers by partially automating linear tasks such as transcribing narrations in note-taking.

Further, to inform our non-linear note-taking approach, we take inspiration from existing note-taking tools [6, 27, 32, 33, 49, 62, 64]–most are intended for text. For example, HyNotes transforms linear into non-linear notes by extracting key concepts and creating corresponding concept bubbles that can then be connected into concept maps [49]. Other tools, such as texSketch, introduce active diagramming through pen and ink annotations to construct coherent diagrams from text with the help of natural language processing (NLP) [62]. The cogSketch system allows students to represent animations as connected static objects (snapshots) [27]. In VideoSticker, we implement a side-by-side text and note-taking views that are linked through the video timeline, and support deferred action similar to GatherReader [32]. Like texSketch [62], we enhance the extraction of stickers and labels through AI-based augmentation.

2.2 Interaction techniques for Videos

Interaction for videos can be grouped into navigation techniques, annotation techniques, and re-representation of video content. To support navigation of video timeline (i.e., seeking frames of interest), For example, SmartLinks automatically generates hyperlinked timestamps to associated notes with their video contents [52]. Other approaches use adaptive fast-forwarding to help people quickly browse videos with predefined rules [22]. The Flow Dragging technique introduced a direct manipulation approach which allows users to control video by moving objects of interest along their visual trajectory [25]. Other solutions build on Flow Dragging to include trajectory cues [28, 37, 38]. In VideoSticker, we incorporate an approach similar to SmartLinks to maintain provenance between notes and the video and show object cues for fine-tuning sticker selection. Further, as with active reading for text, annotations are an effective way to increase perceptual saliency and externalize thoughts over video content. Prior work has looked at techniques for semantic and spatial (moving) annotations on videos [4, 7, 28, 51, 73]. These annotations include path arrows, hyperlinks, speech, and thought bubbles that can be directly added on top of the video and tracked across frames [28, 59]. However, these techniques generally support annotations directly over the video. In VideoSticker, we allow viewers to add annotations on the notes view using the extracted motion stickers, and incorporate display techniques similar to current approaches. Lastly, prior research has looked at interactive techniques for summarizing videos through re-representation. For example, work on video-to-still composition allows for the extraction of a single still image by rearranging video frames [28]. Others have looked at extracting a set of key frames from instructional videos [23], or organizing the frames like a comic book display [10]. However, these approaches automate content extraction. In VideoSticker, we prefer that viewers have agency in selection and content extraction, especially in

supporting learning. Therefore we take a mixed-initiative approach to identify points of interest in the video to suggest stickers of interest.

2.3 Video Processing and Content Extraction

We build on existing computer vision techniques for video processing to implement the note-taking strategies discussed above. We also want to ensure a balance of automation and manual actions essential for active engagement in our system. For instance, Vizig automatically localizes and classifies different anchor points in a video, including figures, tables, equations, flowcharts, code snippets, and charts [70]. VideoGraph automatically summarizes the video content across the timeline by structuring scenes in a two-dimensional graph, which provides a condensed representation at the scene level [74]. While automation can be leveraged in note-taking to identify potentially interesting points, ultimately, active learning requires viewer engagement. Hence, rather than a fully automated approach, we assist readers by presenting salient points of interest on the video timeline and support mixed-initiative extraction to create visual notes.

Video object segmentation and tracking (VOST) algorithms are useful in extracting content from video streams [72]. Though the state-of-art deep learning models have achieved high performance on segmenting and tracking single object over frames [17, 21, 31], the goal for extracting video content for note-taking is different. Rather than aiming for an accurate segmentation and tracking of a particular object, we are more interested in extracting expressive content such as transformations (e.g., the process of DNA chain transforms into a protein) and interactions (e.g., merging of two cells, atomic collision). Moreover, motion graphics in educational videos include abstraction, metaphors, and distorted transformations that are hard to extract accurately using VOST. Therefore, in VideoSticker, we adapted and extended the PoolNet [43] network structure and used a hybrid tracking algorithm to extract stickers that are able to capture these expressive contents in the video (see Section 4 for details).

3 USER EXPERIENCE

VideoSticker's user interface consists of three main regions: a *video player* on the left (Figure 2a), a *diagram panel* on the right (Figure 2b), and a transcript viewer placed at the bottom of the video player (Figure 2c). The video player includes a time bar and standard media controls such as play, pause, close-captions, and volume. At the top of the video player is a drop-down list for selecting a video for viewing (Figure 2d). In addition, controls for active note-taking, including a frame marker button and frame-area selection tool, are placed adjacent to the video player (Figure 2e). The diagram panel comprises a tabbed layout with a primary diagram canvas and notes view for accessing saved notes made using VideoSticker. The diagram canvas has a zoomable, scrollable canvas and a set of tools. These include tools for adding text to the diagram, pen annotation, undo, convert to list, replay, and save (Figure 2f). To better understand how end-users can use VideoSticker for active note-taking from videos, let us follow Neela, a high school student who is learning about *Neutron Stars*.

3.1 Active Viewing

Neela opens VideoStickers application on the web browser in full-screen mode. She selects the video 'Introduction to Neutron Star' and loads it on the media player. Neela then begins watching the video by clicking on the play button. To engage with the video uninterruptedly, Neela follows a two-pass note-taking strategy. This process—with an initial scanning pass followed by a detailed comprehension pass—is recommended for active reading [3, 62]. We support a similar strategy for active note-taking from videos. In the first pass, Neela uses the frame marker button to quickly capture salient points in the video while viewing. When Neela clicks on the frame marker button, VideoSticker generates

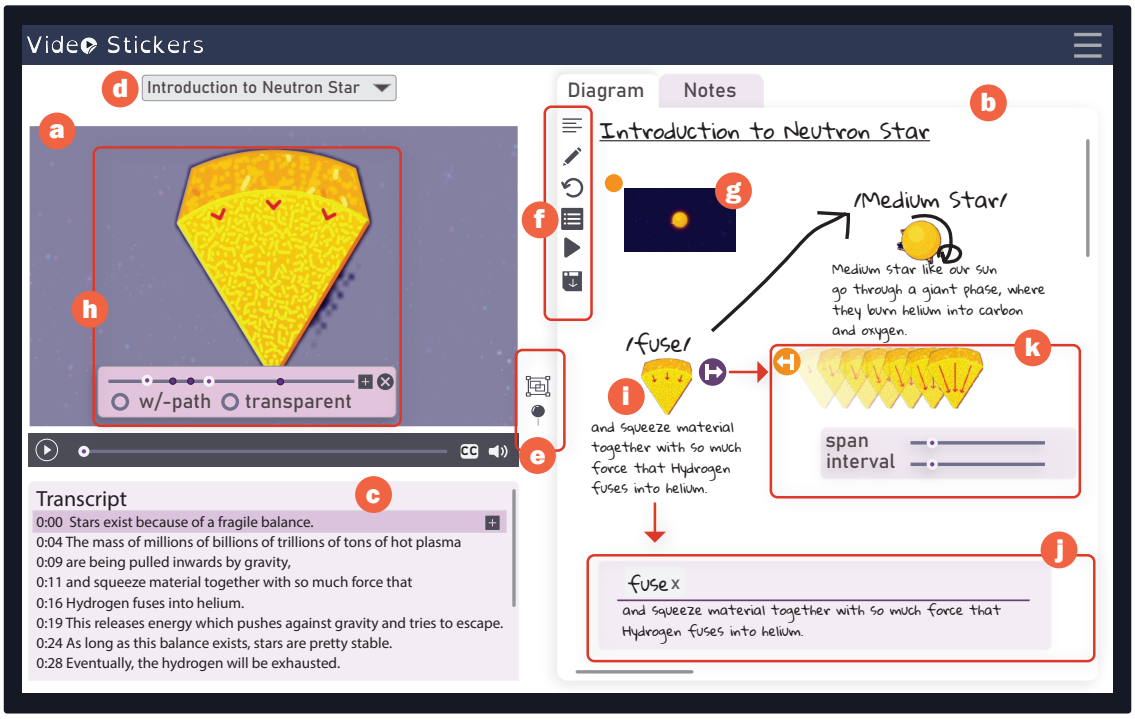

Fig. 2. VideoSticker User Interface

a *frame-sticker*, capturing a screenshot of the current frame, and adds it to the diagram canvas. Though Neela can manually reposition the sticker by dragging, VideoSticker automatically positions the sticker to avoid overlap. As shown in Figure 2g, Neela adds a frame sticker for the point at which the video describes why stars remain stable (i.e., fragile balance). Each frame sticker has a pointer button (orange circle) to support navigating to the corresponding frame in the video time bar. In addition, Neela can add the transcript text to the video for rapid note-taking. While playing, the corresponding lines on the transcript view are highlighted. Each line includes an add (+) button to directly copy the text to the diagram view. Because this is a short (1-minute long) video, Neela watches the entire video in this first pass. For longer videos, Neela can watch a portion of the video and engage in visual note-taking as described below.

3.2 Visual Note-taking

In her second pass, Neela wishes to focus on a few tasks: capturing the mechanism by which stars remain stable, answering why some stars go through a giant phase before transitioning into a white dwarf, and comparing the trajectory of massive stars which collapse into nuclear ash instead. To visually capture these phenomena in her notes, Neela begins with the frame stickers she added in pass one. By clicking on the pointer button associated with the frame sticker, the video player navigates to the corresponding point in the video. Alternately, VideoSticker offers the flexibility to bring up stickers on any frame by simply pausing the video on that frame. In this paused state, VideoSticker displays one or more detected *object stickers* on that frame (see 4 for details on object sticker detection). VideoSticker recognizes

atomic objects on the frame, but also detects composite objects that are made up of multiple elements (e.g., two objects interacting).

Sticker Extraction: As shown in Figure 2h, object stickers are overlaid directly on top of the media player. The opacity of the video on the background is reduced, and the sticker slightly zooms in on mouse-over to indicate sticker extraction action. Neela clicks on the desired sticker object to bring up a local time bar with start-end time ranges for the duration of the video the object is visible. Further, to facilitate sticker extraction, VideoSticker automatically detects *interest points* such as collision displayed as markers on the local timeline. Using this as a reference, Neela first adjusts the start and end duration of the object sticker she wishes to extract. As she moves the slider control, the sticker transitions to the object's state in the new frame to offer a preview. While VideoSticker defaults to the foreground object as a mask for sticker extraction, For adding context, Neela can choose to include the video background around the object by toggling the transparency button below the local time bar. She can also manually select the region of the object sticker by using the frame-area selection tool. Lastly, Neela has the option of capturing the $x - y$ displacement of the object in space for the duration of the sticker. By toggling the path button, instead of an animated sticker fixed in space, VideoSticker generates a motion path sticker. Once satisfied, Neela clicks on the add sticker button, which adds it to the diagram canvas. In addition to the visual element rendered as an *animated* GIF (Graphical Interchange Format), VideoSticker automatically assigns a label for the stickers (see Section 4), and also includes the corresponding transcript text in the generated object sticker (Figure 2i).

Sticker Editing: On the diagram canvas, Neela can edit the label and text for the object sticker. By double-clicking on the sticker, VideoSticker displays an edit panel from which Neela can choose different labels using VideoSticker's predictive labeling feature, or delete and add her own label (Figure 2j). Similarly, Neela can manually edit the label text or directly add text from the transcript using the add button. When the sticker is active, instead of adding the transcript text directly to the canvas, VideoSticker appends it to the end of the current 'active' sticker text. Further, VideoSticker offers frame-level control over object stickers to allow flexible note-taking. By clicking on the *sticker expand* button (purple button with right-pointing arrow), Neela can bring up visuals for individual frames composing the object sticker (Figure 2k). If she wishes to include this frame-by-frame view in her notes, she can adjust the span of all frames along the x axis and the frame interval. In this case, Neela opts to use the sticker expansion feature to take notes about the sequence of events causing the death of a star (i.e., 'Carbon burns to Neon, Neon burns to Oxygen in a Year, Oxygen to Silicon in a month', and so on). In addition, Neela can use the pen tool to manually annotate individual frames (in this case, the duration in months between successive states) or the original composite object sticker. These annotations become linked to the video stickers.

Note-Taking: As part of the note-taking process, Neela can freely re-position individual stickers by clicking and dragging with the mouse pointer. In addition, she can use the pen tool to engage in free-form annotations such as drawing arrows linking multiple object stickers and creating handwritten text. Further, Neela can use the add text button to add text to notes. When adding text, she double clicks on the blank space of the diagram view to trigger an input box. She can use the add button in the transcript view to add transcript or manually type her own text into the input box. After hitting ENTER, a draggable text block will appear on the notes. During this note-taking process, Neela can hover over an object sticker to see the animation in action. She can delete stickers on the diagram canvas by clicking on the sticker and pressing the delete key (or click on the undo button to revert the last action). Finally, VideoSticker supports a convert-to-list feature which automatically reorganizes the stickers on the diagram canvas into a list structure. This is especially useful when taking notes for step-by-step instructional videos such as recipes and workout tutorials (see Section 6). In the future, VideoSticker can be extended for other template types as well.

Through this process, Neela captures the key phenomena demonstrated in the video using object stickers, text, and pen annotations.

3.3 Re-watching, Reviewing, and Export

Once created, Neela can use the semi-animated notes as a reference to re-watch the video or directly engage with the notes as a standalone artifact when reviewing for an exam. Because the object and frame stickers in the notes are linked to the video, Neela can use them as a table of contents to navigate the video sections. Further, the diagram annotations associated with individual stickers are overlaid on the media player at appropriate space and time points when re-watching. This provides a rich and integrated re-watching experience. Third, Neela can use the replay button in the diagram view and VideoSticker automatically plays back each sticker sequentially in the order that occurs in the video, which helps her to get the summary of the video in a quick and concise way. Finally, Neela can export individual stickers for use in other settings, such as authoring a presentation or sharing content on social media. The notes are exported as GIFs.

4 SYSTEM DESCRIPTION

4.1 Video Stickers

VideoSticker supports 2 kinds of stickers: **Frame Stickers** and **Object stickers**. Frame stickers include a full screenshot from the video whereas object stickers capture a sub-object (or objects) that appear the video. Users can create both types of stickers directly. However, when leveraging a two-pass approach of video viewing, the viewer can use a light-weight frame sticker to bookmark an important point and then revisit those points later to replace the frame with object stickers. Internally, frame stickers contain: a screenshot, timestamp, and textual annotations. When the end-user clicks on the 'add text' or 'pin mark' buttons in the interface, VideoSticker will create a frame sticker at that given timestamp. A frame sticker's textual annotations default to the caption/transcript text that was playing at the time of the capture. Frame stickers can be placed and moved in the diagram view. As noted, they serve a secondary role as a marker for fast navigation to a certain part of the video.

Objects stickers can be created both through manual and automated extraction. Both are similar structurally, so we focus on automated extraction. Object stickers encode not just a snapshot of the object (or objects) in a given frame but also the motion information. Like frame stickers, object stickers also include a timestamp relative to the original video. However, as we are tracking motion, VideoSticker stores the *start and end times* for the sticker. In reality, we retain two different ranges. The first is the object's start and end times in the video as calculated by the object tracker (the *object range*). This may track when the object appeared or disappeared in the video. Additionally, we keep the user-defined start and end time-range (called the *sticker range*). This second range is a sub-range of the first and indicates the sub-piece of the video that the viewer would like to include.

Visually, each object sticker is defined by: a *sequence of bounding boxes*, a *sequence of frame masks*, and a list of *interest points*. The bounding boxes serve as rough estimates of where the object are over time. These are simply the top-left and bottom-right corner coordinates of a box that encapsulate the object in the original video frame. The mask sequences are more refined attempts to describe the object. These are a sequence of pixel masks that are tuned to the shape of the object given the extraction process. The interest points indicate where there are potential object interactions (e.g., the collision between two objects in the video).

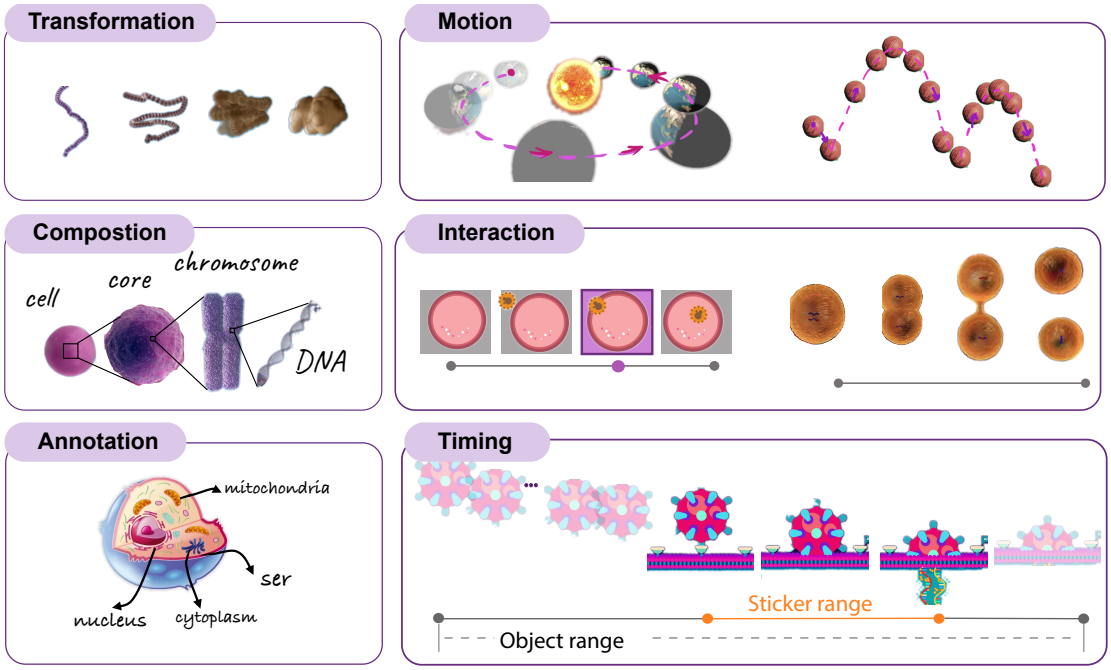

Fig. 3. Six dimensions of video content that encodes in Object Stickers

Like frame stickers, object stickers can store various pieces of associated text. This information is retained as a *label list* (a list of what the object might as defined by an automated process), a *selected label* (what the viewer would like to call the object), and *narration* (the extracted caption/transcript information). As described below, default values for the textual information are set through automatic extraction but can be modified by the viewer. Finally, object stickers also include *end-user annotations* which are the text and pen-and-stroke marks made by the end-user. These can be both displayed in the diagram view but also overlaid in the original video.

There are many different types of motion and interaction that we might want to capture in an animated sticker (see Figure 3). Our stickers are intended to be used to extract and repurpose video content into novel narratives. While flexible, video is nonetheless a linear format. Content producers have learned how to best leverage the video medium for education [13] or for animation [65]. Short pieces of animated content can convey sophisticated narratives. With VideoSticker, our goal is to repurpose video clips as a novel "semi-animated" format that can capture the kinds of narratives that exist in video materials. In studying videos for note-taking, we have identified a number of common dimensions that we would like our stickers to produce.

(1) Graphic Content Transformation: Videos vividly capture the transformation of objects over time. The transformation is often part of a causal narrative. With Object Stickers we would like to similarly capture a sequential shape transformation of each object over time. For example, the protein folding process shown in Figure 3 (Transformation). The sticker shows how a polypeptide chain folds to become a biologically active protein in its native 3D structure. We do this through sequence masks applied to the original video that allow the viewer to focus on the transformation

'story.' Note that while VideoSticker attempts to 'mask' the object(s) of interest, it can still display either a local context (e.g., the immediate background of the object) or global context (e.g., the entire frame).

(2) Motion In addition to transformation, videos create narrative through other kinds of motion. For example, as shown in Figure 3 (Motion), the earth elliptically orbits sun. By showing the bouncing ball over time, students can easily learn how the vertical speed of the ball changes. Object stickers can capture this motion. However, while the object can move significantly in the original video, space for object stickers may be more restricted. To capture the motion narrative in this situation, frames in object stickers can be extracted and laid out to convey this narrative. For example, a basketball bouncing in a sinusoidal fashion can be captured in a sequence of frames laid out as in Figure 3 (Motion, left). Internally, we support this through a combination of bounding boxes and masks. Spatial positioning of each object sticker is stored in bounding boxes. Showing this information in sequence allows the exploration of object's trajectories. Moreover, since the frames are sampled in a fixed time interval, the positioning information can be used for speed visualization.

(3) Composition Video narratives often seek to explain ideas of scale and containment. Traditional videos can often display by 'zooming' in and out and changing the objects based on the level. For example, a video might illustrate how we go cell to nucleus to chromosome to DNA (see Figure 3, Composition). Stickers can represent these as separate objects connected either graphically (e.g., by lines) or through animation. Internally, we can capture this by enabling image masks to capture the Zoom in/out animation where we would like to narrate a part-to-whole narrative.

(4) Interactions One of the most significant narrative types in videos is the interaction between objects. Objects collide or separate to convey different kinds of interactions. In some cases the interactions capture a 'reality' (e.g., an animation of a virus entering a cell). However, interactions may also be used in a stylized way. Figure 3 (Interaction) shows two examples of interactions captured by the object stickers: the point when an virus enter the cell and two cells split during mitosis. Notably many interactions can lead to the appearance of a new 'joint' object or a transformation to one or both of the original objects. In VideoSticker, we support this in two ways. First, object stickers do not necessarily track single objects. Two separate objects that have interacted to be become one can be represented as a 'meta' object (e.g., we can create a new sticker representing the infected cell). Second, VideoSticker will try to identify and annotate interactions to allow the end-user to decide how they want to create stickers for that interaction. These interaction points are retained and can be used for narrative creation and other applications.

(5) Text annotations In instructional videos we often find text annotations. These help guide the narrative or call attention to interesting or important elements (e.g., Figure 3, annotation). With stickers, we attempt to both preserve textual information in the original video as well as allowing for new annotations on the stickers themselves. The end-user can use this text to create a new narrative that both explains what is happening in the sticker as well as the connection between that sticker and others (e.g., in a diagram view). Our goal is to preserve the narrative of the original video as well as allowing the end-user to create their own version.

(6) Temporal Context Within videos, time is an important instrument for creating narratives. Objects in videos change, interact, and move over time. In some situations 'context' is important to narratives. It may be important not just to understand the specific instant in which the virus enters a cell, but also the states immediately before and immediately after. In creating an object sticker, we would like to support ways of recording both the context and key frames. To do so, we can encode two time ranges (as described above). The object range associates with the existence of certain object in the video, which is helpful to segment video in time and identify topic transitions in a video. Whereas the sticker range encodes the segments of the video that drives viewer's interest, which highlight the video contents and helps user quickly navigate to certain parts of the video. As shown in the Figure 3 (Timing), the the sticker range includes

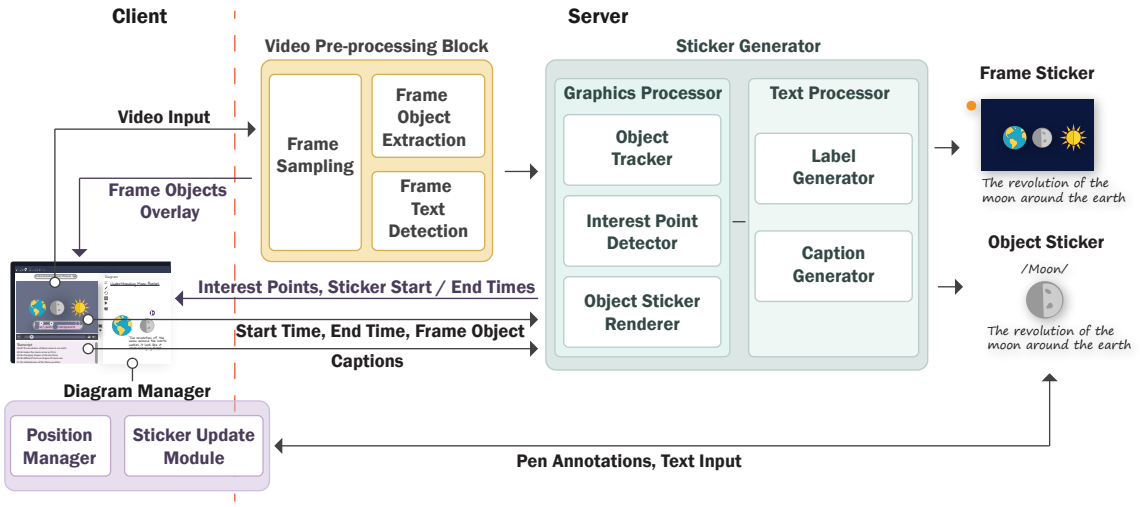

Fig. 4. VideoSticker Architecture

interaction frames that capture how the corona virus injects generic material to the victim cell, while the object range touches from the start to the end of how the corona virus affect the cell.

We note that these may not capture *all* possible types of video sticker facets. However, they demonstrate the flexibility and breadth of object stickers across multiple use cases.

4.2 System Architecture

This section describes in detail how video stickers are generated and user interactions are supported by the system. At a higher level, the input video first go through a *Video Pre-processing Block* to have the most computational tasks done to guarantee a real-time user interaction. The video watching and note-taking process start after the pre-processing. In the 'Active Viewing' and 'Sticker Extraction' stages, client interacts with the server by either direct retrieving pre-processed data or calls for further computation from the Sticker Generator. After the stickers are extracted from the video, the *Diagram Manager* is responsible for organizing the stickers and user inputs while 'Sticker Editing' and 'Note-taking'.

4.2.1 Video Pre-processing. The pre-processing stage goes through a 2-step processing for an input video: (1) Frame sampling, and (2) Frame-by-frame content extraction, which includes object extraction and text detection with the sampled frame images.

The **Frame Sampling** module takes in a video and stores sampled frame images that are ready for direct retrieval in the server. As a trade-off between saving computation power and ensuring a smooth animation, We adjust a sampling rate based on the video's FPS (frame per second) to ensure 5-8 images are included in 1 second (e.g. for a 60 fps video, we choose sampling rate to be 0.1, so that 6 frame images and included in 1 seconds).

Each sampled frame image will be processed for content extraction. The **Frame Object Extraction** module takes in each frame image and detects multiple frame objects and texts in the image. For **Frame Object Extraction**, the outputs are binary masks for salient object (saliency maps), separate images for multiple frame objects and a frame object JSON file that stores information about each detected frame object's image path and bounding-box within the frame image.

The technical details of frame object extraction are described in the Appendix. For the **Frame Text Detection**, we used the Optical Character Recognition in Google Cloud Vision API. All texts detected are stored in a dictionary with timestamp as the key.

4.2.2 Sticker Generator. The **Sticker Generator** takes in the caption file of the video and the outputs from Pre-processing Block and generates stickers by calling the Graphic and Text Processor.

Frame Stickers are generated by querying screenshot with timestamps marked by end-users. When user pauses the video, all the objects detected at current frame will show up on top of the screen. Object Sticker extraction is triggered when the user clicks on an object to indicate selection. The **Object Tracker** starts to track the object along the video timeline and returns a list of object images (see Appendix A.1.2). At the meantime, the interest point detector detects a list of interest points (see Appendix A.1.3). With these 2 parts of results, the system shows a local timeline with marked interest point and 2 functional toggles. With the 2 toggles, we define 3 rendering modes: *mode0:* object with transparent background and motion path; *mode1:* object with transparent background and fixed center; *mode2:* object with a blurring background. With the time duration and rendering mode specified, the **Motion Sticker Renderer** works to render an animated object using image processing function built in openCV [12]. The rendered object will show up as a preview on the video screen, replacing the static frame objects. When the 'add to diagram' button is clicked, the **Text Processor** gets the corresponding caption and a list of detected labels (see Appendix A.1.4) with time duration of the extracted sticker.

4.2.3 Diagram Manager. The **Diagram Manager** manages all the stickers on an SVG canvas created using the D3 JavaScript library [11]. When a sticker is added, the diagram module defines its default position to avoid occlusion of elements and add the positioning attribute to the sticker instance. This module is also responsible for updating the sticker instance based on user inputs: i.e., changing of caption/label and adding associated annotations.

4.3 Graphical User Interface

VideoSticker's user interface is implemented as a Web application using HTML and JavaScript. The interface connects to a Python Web Server that implements the features described above. We implemented the video panel using Video.js [14] with transcript support through an interactive plugin [67]. The functions of the diagram panel are implemented using D3.js [11] to support the layout of stickers (automated and user-driven), sketching, annotations, and interactivity. In addition, we used the Tagify [53] library to implement label recommendation and editing.

5 PRELIMINARY USER STUDY

We conducted a user study to gather feedback on the note-taking experience with VideoSticker. Specifically, our aim was to (1) assess whether participants can use VideoStickers features to engage in active viewing and visual note-taking from videos, (2) determine the cognitive load from using VideoSticker, and (3) gather feedback on the overall usability of VideoSticker. We hosted the VideoSticker on a remote server that participants could access via a web URL and conducted the study online via Zoom. We recruited 10 participants (undergraduate and graduate students) with prior experience with visual note-taking. As self-reported by participants, they took visual notes during lectures and e-learning activities. Each session lasted for 75-90 minutes, and participants received a $25 Amazon gift card for their time.

5.1 Method

For the study, we selected two expository science videos from YouTube. The first described the process titled 'MITOSIS, CYTOKINESIS, AND THE CELL CYCLE' produced by the Neural Academy channel [2]. We specifically selected the explanation of Mitosis (running between 5:32 and 6:42). The second video, 'Neutron Stars – The Most Extreme Things that are not Black Holes' was produced by the "Kurzgesagt - In a Nutshell" channel [1]. In this video we selected the piece explaining the formation of the 'Neutron Star' (0:29 to 1:51). We clipped the original video to approximately 90 seconds to only include the core explanations necessary for understanding. We selected these scientific videos because they contain connected causal concepts and rich animated visuals that are crucial to understanding the structural and mechanistic details of the phenomenon, thus are well suited to stress-test a wide range of VideoSticker features. We restricted the length of the video to balance the number of conditions we could test, the VideoSticker features we could evaluate, and at the same time ensure our participants did not become over-tired. From an earlier pilot session, we learned that participants could take 10 minutes to engage with one minute of the video depending on how engaged they were and what types of notes they were creating (though part of this was due to experimenting with system features). Each participant worked on three note-taking tasks: one using pen-and-paper or an existing tool of their choice (i.e., baseline condition), the remaining two tasks using VideoSticker.

At the start of the session, the study coordinator provided a brief overview of the study's purpose, the data that will be collected (screen recording, notes, and VideoSticker usage log), and received consent from participants. Next, participants were asked to watch the video about Mitosis and take notes using a tool of their choice (Task 1). We instructed participants to engage with the video and notes as if they were studying for an exam. Two of the participants used Google Docs, one participant used the Notes application in Mac OS, one used a tablet-based note-taking app, and the rest used pen and paper and later shared a photo of their notes. We aimed to understand their current note-taking behavior from videos and elicit comparative feedback between their current approach and VideoSticker. Once participants completed Task 1, the study coordinator introduced VideoSticker by providing a guided walkthrough using a third video about protein structure from 'PDB-101' [55]. Next, participants launched VideoSticker on their web browser (with screen-sharing enabled in record mode) and practiced using VideoStickers features for the same video. Once participants indicated familiarity with VideoSticker, they proceeded to Task 2. Here, we asked them to reconstruct the given visual notes for the video about Neutron Stars. As shown in participant P10's reproduction of the notes (Figure 5), the task required them to incorporate all of the features implemented in VideoSticker, including frame and motion stickers, sticker annotations, frame-by-frame expansion, and incorporating transcripts and labels. The intent was to assess participants' understanding and recall of the features in VideoSticker in a controlled manner. During this task, participants had access to PDF copies of the notes we asked them to recreate by watching the video. The task was time-bound to 20 minutes. At the end of this time, we asked participants to begin task 3, in which they revisited the Mitosis Video and constructed notes, this time using VideoSticker.

At the end of each note taking task with VideoSticker (i.e., Tasks 2 & 3), we asked participants to fill the NASA Task Load Index (NASA-TLX) questionnaire [29]. At the end of the study, we asked participants to respond to the Post-Study System Usability Questionnaire (PSSUQ) [42]. Finally, participants provided open-ended feedback on VideoSticker including thoughts on how they might use the tool in their own learning scenarios and also features they'd like to see.

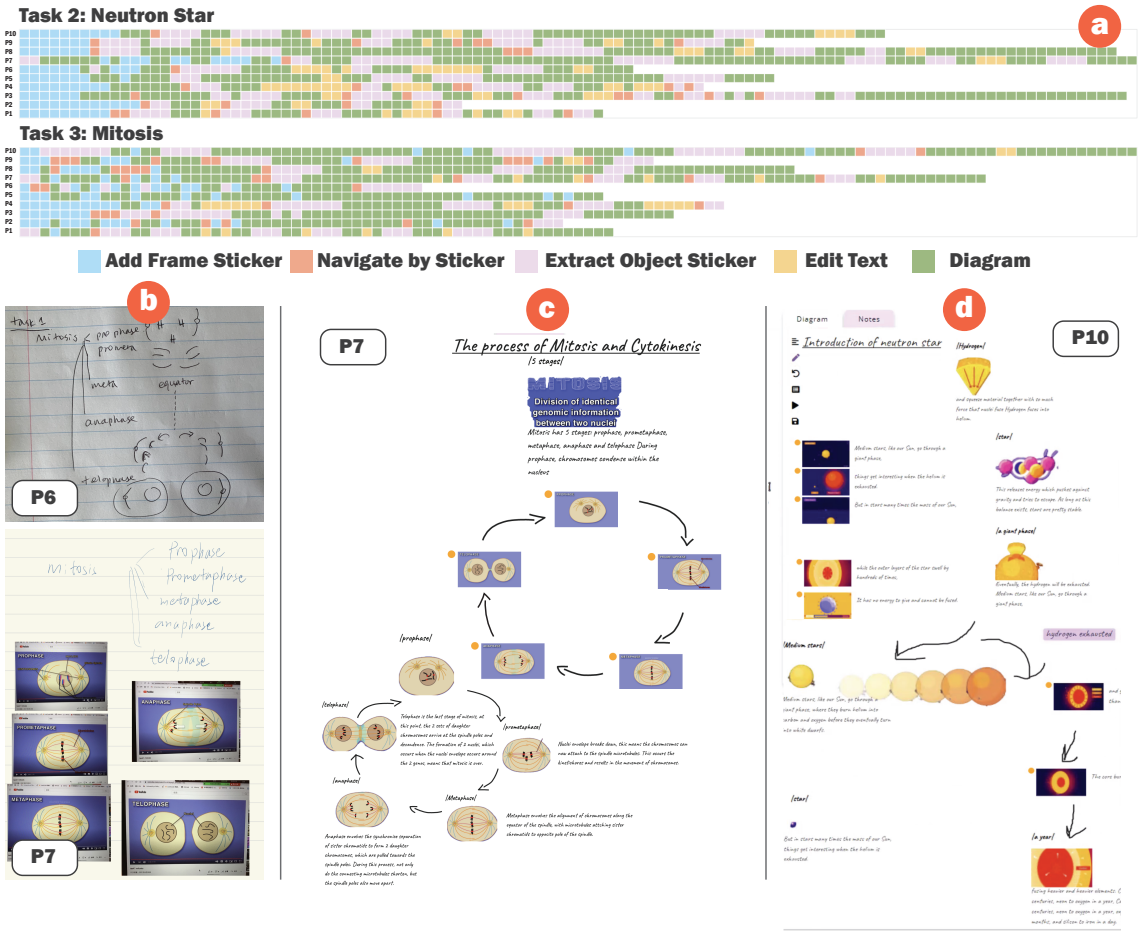

Fig. 5. Results from User Study: (a) System Usage Log for Tasks 2 & 3 (Each square indicates a single operation); (b) Example notes for Task 1—baseline note-taking condition; (c) Example notes for Mitosis; (d) Example notes for Neutron Star.

5.2 Results

5.2.1 Active Viewing and Note-Taking using VideoSticker: All participants followed the two-pass approach (viewing through, followed by a second viewing for notes) for the Neutron Star video in task 2. All but one used the two-pass approach in revisiting the Mitosis video for task 3. P1 reported familiarity with the topic, and as shown in Figure 5 a (system usage log), they directly worked with object stickers. They took less time and fewer steps compared to other participants. Further as shown in the log data visualization, participants would first add frame stickers, then extract object stickers, diagram (drag and arrange them, add annotations and edit the text), and repeat the process across different frame stickers (→ → +). When adding frame stickers, participants reported selecting frames with the most information about the concept being explained. In both videos, this included the end states of animations and frames with embedded labels. All but one participant organized the frames as a list structure before adding object-specific details. For instance, participants would list the five items for the mitosis video, one for each

phase (prophase, prometaphase, metaphase, anaphase, and telophase). In the second pass, six participants replaced frame stickers with object stickers, and four retained both frame and introduced object stickers in a side-by-side view. Rather than organizing stickers as a list view, P7 opted to arrange the frame stickers cyclically. P7 also annotated the frame stickers with the arrows and used it as a template (or structural overview) to generate a detailed version using object stickers. P7's notes demonstrate a novel use of frame stickers beyond provisional 'bookmarking.'

Across all sessions, participants took an average of 11.1 seconds (σ = 8.4 seconds) to extract a single object sticker. As we discuss in section 7, this was partially due to online nature of the study and running the server remotely. Participants used the text editing features in VideoSticker to rephrase the extracted transcript text and make it more concise. In some cases where the transcript was misaligned or did not overlap with the animation (e.g., the speaker introduces a concept and then shows the animation). A few participants addressed these animation-transcript coordination issues by editing. Some participants added stickers by selecting the lines in the transcript directly. In supporting these note-taking requirements, all participants commented on VideoSticker's flexibility in navigating between their notes and video views. As P7 reported: *"Detaching the objects and controlling them in the video viewing panel is cool... So I can see how it transforms in the context of the video."* (P7). Further, they appreciated the object extraction and transcript alignment features implemented in VideoSticker. Based on participants response to the cognitive load questionnaire (by converting tick marks to a 100 point scale), participants rated mental demand as 43.5 (σ = 20.8), physical demand as 31 (σ = 23.3), and temporal demand as 38 (σ = 35.6). Further, they rated their performance (0–Perfect, 100–Failure) as 37 on average (σ = 25.5). Lastly, they self reported their effort as 53.5 (σ = 16) and frustration as 26.5 (σ = 21).

5.2.2 Visual Notes: Six of the participants' notes primarily consisted of textual representations such as lists and flow diagrams in the baseline condition. Three participants attempted to visually illustrate how a single cell divides into two daughter cells. However, because of the time to sketch and content similarity between phases, subsequent drawings lacked details and were incomplete. For instance, P6 only represented the visual differences between phases, such as the chromosomes align at the center in metaphase and the cell pinches in telophase. In the baseline condition, one participant directly took screenshots of the video and included them in their notes as an ordered list (a feature directly implemented in VideoSticker through Frame Stickers). Across all sessions, participants found it effortful to capture the rich mechanistic details present in the narration. As P6 reported: *"Writing all the words and illustrating the graphics in a diagram is particularly difficult... I sketched for some phases and left several keywords for the others... it demands a lot of effort."* As a standalone resource, the notes were inadequate in describing Mitosis. In comparison, notes made by all participants using VideoSticker captured animated transitions for all phases. Specifically, participants used the frame expand feature for the stickers in each phase to visually see the cell states between phases. As P5 reported: *"I really like that I can extract the animation out, especially the (expand) tool with interval and span. It is kind of a screenshot but in a sequence... it saves a lot of time for me"* (P5). Similarly P7 commented on the advantage of VideoSticker over conventional note taking tools: *"I really like the feature of capturing animation in the video... why you choose to watch video is that you can see the motions, like you can see the planet changing. That's how we can understand the concepts in a very live way. It will help us understand and remember things that conveyed in the visuals."* In addition, the notes contained both labels and text from the transcript, which captured the expository details of the mitosis mechanism.

5.2.3 Usability: Based on responses to the usability questionnaire on a 7-point scale ranging from 'strongly disagree' to 'strongly agree,' participants rated VideoStickers *usefulness* as 4.48 (σ = 1.29). One concern was that the diagramming panel in VideoSticker did not support sophisticated note-taking features typical in commercial tools (e.g., text styling, pen customization, and scale and rotate the stickers). As P10 commented: *"I feel the functionality for extracting information*

from videos is sufficient, but it is hard to draw a desirable diagram with the tools in the diagram panel…". In addition, 3 participants attempted to drag the sticker directly from the video panel to the diagram panel before they realized that they needed to hit the 'add button.' We plan to incorporate these suggestions in future iterations and evaluate VideoSticker with more expressive diagramming features and support for touch-based direct manipulation. Further, participants rated *learnability* of VideoSticker as 4.60 (σ = 0.70), and likeability as 5.30 (σ = 0.67). In providing feedback about object stickers, P8 commented *"I really like the idea that I can extract the animations and the images in the video. I actually did not have that idea before and I've never seen tools that can do this. It's pretty cool to extract a sticker…"* (P8). However, they had a moderate agreement that they could recover from mistakes easily and quickly ($\mu = 3.75, \sigma = 1.16$). Follow-up studies may be useful to assess error recovery needs and support validation and error messages. Lastly, participants also imagined other uses for VideoSticker beyond video comprehension, such as making slideshows. According to P6: *"It would be great to export those stickers and put them in slides for presentations…"* (P1).

In summary, the findings from the user study suggest that active viewing and note-taking with VideoSticker can be an effective way to learn from high-bandwidth video content. In a future study, we aim to incorporate participant feedback regarding note-taking and evaluate the learning effectiveness in science classrooms.

6 USAGE SCENARIOS

In addition to the common learner comprehension scenario described in Section 3, VideoStickers features can support a variety of communicative, analytics, and skill-learning needs. We briefly describe additional use cases demonstrating the broader applicability of VideoSticker.

6.1 Using Videos in a Lecture

Videos are often embedded inside other educational modalities. In the context of a lecture, instructors regularly embedded external videos for concept illustration. However, the intent is not just to 'show' the video to students but to walk them through the key concepts in the video to meet specific learning/communication objectives. Unfortunately, it is hard to "break down" video content dynamically using current tools. For example, an instructor in a biology classroom may want to teach students about protein synthesis. With traditional slideshow approaches, they can narrate the video as it is playing while repeatedly pausing and rewinding the video to control the pace of the lecture. An alternative would be to create small, focused video clips in advance. In both cases, control is limited or effortful. As an alternative, the presenter can use VideoSticker to extract 'motion stickers' and embed them in their slideshow dynamically (Figure 6a). In this approach, the presenter controls the display of individual concepts in the video but can also leverage the rich set of features in presentation software such as highlighting, annotating, and animations to reveal concepts one at a time. For example, a strategy could be to present the video in its entirety and then use motion stickers in slides to draw attention to key concepts in the video. In this case, describing the structural composition of protein molecules in the cell, and then elaborating about key steps in protein synthesis, such as the mechanism for creating a template for protein formation.

6.2 Sports Analytics

During Soccer practice, a coach may record the game at play and later discuss it with the team. The intent is to revisit shots and passes, discuss player performance, identify alternative strategies, and offer feedback. With VideoSticker, the coach can extract motion stickers of key play points such as goals, shots, and ball 'pass' interactions between players. By using the sticker expansion feature (see Figure 6b), they can lay out the player movement frame-by-frame and

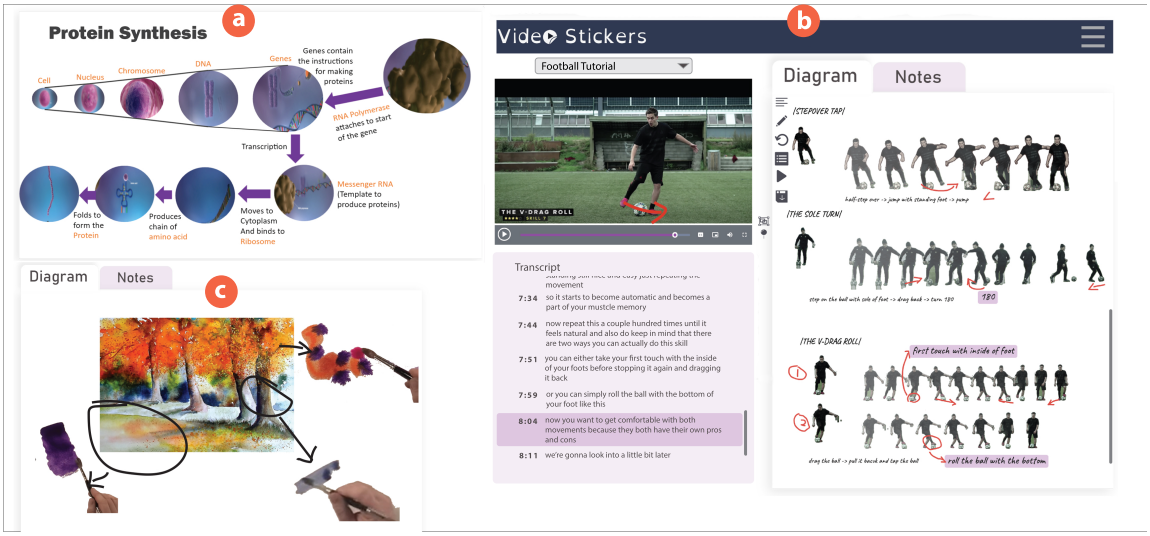

Fig. 6. Other Usage Scenarios for VideoSticker: (a) Presentation of video content by embedding motion stickers in a slideshow; (b) Analysis of soccer practice by annotating frame level stickers; (c) learning to paint with watercolors by extracting motion stickers for brush techniques.

further annotate player stance and alternative kick strategies (e.g., step-over tap). The coach can also organize stickers side by side to support comparative analysis (e.g., desired vs. current kick stance) and generate notes for each player in the team. In addition to offering feedback using stickers and notes, the players can re-watch the video with the feedback annotations overlaid on top of the video.

6.3 Skill Learning

Finally, videos are effective at demonstrating skills such as knitting, painting, or folding bread. Due to the high bandwidth nature of information in videos, learners may find it challenging to grasp precise details of the technique they are learning. They may have to re-watch the video multiple times while practicing 'hands-on' to learn the fundamental techniques (e.g., crocheting a slip stitch, painting gradated wash with watercolors, or shaping a loaf of bread). Additionally, learners may wish to refer to multiple videos to understand variations and alternative techniques to find one that works for them in acquiring the skill. VideoSticker can support skill learning in an informal setting by allowing learners to curate motion stickers and organize and annotate the techniques to support their learning goals. For example, a learner might want to create a watercolor painting that incorporates different types of brush strokes (see Figure 6c). The learner annotates the target painting with motion stickers of different brush strokes sourced from multiple tutorial videos to support the creation process. VideoStickers replay features allow them to view the brushing techniques just-in-time as they are painting.

7 DISCUSSION

7.1 Guidelines for Video Notes and Stickers

VideoSticker offers a solution for video note-taking and potentially a new type of note format, i.e., the semi-animated narrative structure. Through our design and user studies we have identified various guidelines that have motivated our

implementation. We share these as potentially important for video note-taking, in general, and sticker-style systems, in particular.

When taking notes from videos, we believe that there is both a need and opportunity to create a ***bi-directional link between the notes and video***. Standard video formats make certain tasks hard. For example, 'random access' or finding points of interest in a video is difficult. Similarly, understanding narratives in anything other than the linear 'path' laid out by the creator can be challenging. One approach is to create better links between the video and note format. In VideoSticker, we have attempted to do this by using the stickers as a mechanism for re-organizing the video information while still supporting linked-navigation to points of interest in the source video. Similarly, annotations created in the notes can be added (i.e., overlaid) on the original video content. This allows for a different, augmented, experience for the viewer. A key element of creating these kinds of connections is to ***minimize the interruptions of creating links***. While we would like to build robust links, this should not come at the expense of the video watching experience. For example, well produced videos will have a flow and pacing that can be broken with frequent pausing. If the system requires heavy-weight interactions to create links, the experience is damaged. VideoSticker addresses this by allowing for a light-weight mechanism for saving bookmarks on the initial viewing (pass 1), and later refinement of the diagrams and stickers.

With sticker-style notes, we believe it is desirable to support different types of selections to create stickers as well as different visual representations of the stickers themselves. Providing a variety of extractions and representations ensures that we can ***support a broad range of graphical narrative structures.*** Conventional video narratives can have specific ways by which we can convey causality, interactions, transformations, and other 'story' elements. With notes, which are often summaries of the video, we would like to ensure that the same range of narrative elements can be maintained in the new format. Relatedly, we believe that notes for videos are useful in diverse scenarios. Thus, it is desirable to ensure ***support for a broad range of layout formats.*** This includes not just flexible drawing canvases, but potentially more structured templates (e.g., lists, tables, teacher-provided visual organizers, partially-complete notes, etc.). Stickers should also be ***extractable to other media.*** Animated clips, static graphics, or stickers should be usable outside of any particular framework. We have represented ours as animated gifs. However, more structured formats may allow for embedding in presentation software or even 'placing' back in videos. We also note that VideoSticker works on the assumption that the video creator deployed the materials without considering the possibility of video note-taking. An alternative approach to support the range of narrative structures and formats is to build video creation software that not only produces videos, but also the stickers that go with them.

Notes are visually varied, in part due to the range of narrative tasks. To better support note-based storytelling, it is useful to ***support a broad range of annotations*** and ***mechanisms for connecting stickers***. Videos contain various forms of explication of an idea. Not only do we have the actual information moving, we may have narrations, text overlays, and other ways of describing what is going on. In building VideoSticker, we have to capture these information channels and to allow the user to add them as annotations (transcripts, labels, etc.). Additionally, we support other kinds of annotations, both textual and graphical, to be associated with both individual stickers and notes as a whole. Because we are moving from one medium (the video) to another, the semi-animated diagram, a rich language for translating the narrative becomes crucial. For connecting stickers, we have focused on standard diagram representations (e.g., with sketched lines, arrows, and boxes). However, it is possible that other types of connections are desirable. For example, one could (spatially) move the stickers themselves within the context of the diagram to represent interactions.

With videos, the creator has defined a specific way of watching the content. With few exceptions (speeding up, slowing down, rewinding, etc.), the experience of 'watching' is largely pre-determined. Video notes, and in particular

the sticker-based form, break this structure in some ways: the are semi-animated. Parts of the notes may be video clips, parts may be static. There may be hand drawn elements, screen captures, and text passages of different lengths. Unlike linear videos, notes can be used to represent branching or cyclic narrative structures. However, as they contain a blend of content and structure it becomes critical to ***support a diverse set of playback options***. There may be situations in which the ideal playback follows the order in the video (which VideoSticker supports). However, we have observed that there may be other playback options that might be better (e.g., simultaneous playback, staged playback, etc.). Here, we feel like there is a significant opportunity to consider how video notes should be sequenced, interacted with, and 'played'.

Though our exploration and construction of VideoSticker, we believe that there is a rich area of research for new video note-taking applications. The 'sticker' implementation presents a possible point in this design space.

7.2 Limitations and Future Work

In constructing the VideoSticker prototype we opted to implement key features for extracting different types of stickers and building notes. However, our prototype was not a full featured commercial application. The user study showed our participants often identified features of diagramming applications they were familiar with (e.g., rotation, different pen styles, etc.). We believe that adding these features would enhance the experience and view some of the usability scores as a baseline. Moreover, as we noted above, object sticker extraction time was relatively slow. This was an unfortunate side-effect of running the experiment over (a video-conferencing software). We utilized a basic remote server and for some participants we experienced significant network delays. In contrast, a local execution of VideoStickers server and client on a performant machine resulted in near real-time object sticker extraction. However, we note that many remote hosting problems might be alleviated through more pre-processing of the videos and caching. VideoSticker can leverage stickers extracted by one user, to speed up the process for others.

One key future research direction would be to better understand end-user expectations given this sticker 'modality' which is part diagram and part video. First, we observed that our participants would sometimes treat the video itself as if it was built from stickers. That is, the system had access to the "objects" and text in the videos (particularly the more stylized figures). Just as drawn annotations on the sticker appear in the video, it is possible to imagine replacing video content itself with object stickers from somewhere else. Secondly, one of the main motivations for VideoSticker was to allow viewers to diagram not just individual objects but the interactions between them. However, one aspect that remained unclear from our study is how viewers would want to see these interactions when replayed in the diagram. For example, when two objects collide and interact should they be displayed as three stickers in the diagram (two for the independent objects and one for the collision/interaction)? Or should the stickers themselves move in the diagram to approximate the interaction? Additionally, many interactions are not simply two objects bumping into each other. For example, we may observe one object (e.g., a chromosome) inside another (e.g., a cell). This presents a challenge for our automatic interaction point detection and prevents VideoSticker from accurately detecting the start and end times. While an end-user can extract stickers for either or both objects, this is more time consuming. An area of future focus for us is to support better diagrams for and automated tools for object interactions.

As there is often an expectation that sticker extraction would be more accurate and simultaneously flexible, future system can improve extraction in both algorithm and interaction aspects. To evaluate the accuracy of extraction algorithm, we have started to study the efficacy of the sticker extraction across a wider range of videos. While this is outside the focus area of this paper, we have sampled videos from a range of educational video producers and are analyzing our ability to track and extract content automatically. As might be expected, videos with simpler backgrounds

and fewer non-overlapping objects tend to work better. This is not a particular limitation of VideoSticker, but rather of state-of-the-art tracking algorithms. As these algorithms are enhanced, so will the capabilities of VideoSticker and tools like it. That said, it would be beneficial to know how well and how quickly stickers can be created across a more diverse set of videos. This will also guide error correction features.

Moreover, the system would benefit from a multi-modal sticker extraction. When extracting object stickers, we observed that participants tried to match the duration of the sticker (the slider ranges) with corresponding narration timestamps (seen in the transcript). However, in many videos, the narration and animation do not align for some content. This can happen when the narrator introduces a topic and then the actual animation happens or when the narration is shorter or longer than the animation sequence. While we can record corrections made by one user for others, an interesting research question is if we can automatically align the transcript and video content.

As for interaction techniques, one design decision we made for our current prototype was to focus on the automated extraction possibilities for stickers. However, we do not believe that this is the best strategy for realistic deployment [5]. Additional support is needed for human feedback and error correction. For example, end-users should be able to mark the location of objects on the video by some common practices in annotation such as drawing regions, or 'scribbling' [24]. A similar approach would enable the end-user to define their own stickers, correct extraction mistakes, or even guide the system. Additional correction and control features can improve the end-user experience. However, VideoSticker can also be enhanced by allowing video content creators to generate stickers themselves and embed these as part of the video. This can be done *after* the video is produced, or even, with access to the motion software, during production.

Our study was designed to better understand the feasibility and possibilities of our animated sticker metaphor in the context of note-taking. By allowing participants to use whatever tools they were familiar with, we were able to better contrast how VideoSticker features could replace or enhance existing practice. This design is limited and clearly not intended as a controlled test. With the current study design, we do not yet know if the system leads to better learning outcomes. Based on participant feedback, we can hypothesize that VideoSticker can facilitate long-term learning by facilitating future reference but there is no direct quantitative results testing the hypothesis. A future experiment would allow us to validate the tool in a real learning setting with with broader genres and longer length of videos and design better controlled study to quantify the learning outcomes.

8 CONCLUSION

Short-form videos offer an expressive medium for knowledge representation and communication. Viewers regularly refer to online videos about scientific concepts, skill learning, and how-to instructions. While these videos are highly engaging with rich graphics, narration, and animation, they also demand a high level of attention from viewers. Unfortunately, current approaches to viewing are primarily passive and do not support viewers in managing the cognitive load of video comprehension. In this work, we propose VideoSticker, an active viewing and interactive note-taking tool that scaffolds viewers in comprehending high-bandwidth video content. Specifically, VideoSticker allows viewers to externalize their understanding by deconstructing the video into animated stickers and linked labels and descriptions. Through these features, viewers can extract, annotate, and organize the dense information in videos into digestible representations such as lists and diagrams. In addition, VideoSticker automatically links the visual notes with the source videos allowing viewers to leverage the benefits of both formats. We demonstrate the utility of VideoSticker through a variety of usage scenarios and offer design considerations for note-taking from videos.

ACKNOWLEDGMENTS

This research was partially supported by grant R305A170489 from the Institute for Educational Sciences.

REFERENCES

[1] Kurzgesagt – In a Nutshell. 2019. *Neutron Stars – The Most Extreme Things that are not Black Holes.* Youtube. https://www.youtube.com/watch?v=udFxKZRyQt4

[2] Neural Academy. 2019. *MITOSIS, CYTOKINESIS, AND THE CELL CYCLE.* Youtube. https://www.youtube.com/watch?v=8uzHTKdv_Sw

[3] Mortimer J Adler and Charles Van Doren. 2014. *How to read a book: The classic guide to intelligent reading.* Simon and Schuster.

[4] Megha Agarwala, I-Han Hsiao, Hui Soo Chae, and Gary Natriello. 2012. Vialogues: Videos and dialogues based social learning environment. In *2012 IEEE 12th International Conference on Advanced Learning Technologies.* IEEE, 629–633.

[5] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19).* Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300233

[6] Aaron Bauer and Kenneth R Koedinger. 2007. Selection-based note-taking applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems.* ACM, 981–990.

[7] Clément Benkada and Laurent Moccozet. 2017. Enriched interactive videos for teaching and learning. In *2017 21st International Conference Information Visualisation (IV).* IEEE, 344–349.

[8] Mireille Bétrancourt and Kalliopi Benetos. 2018. Why and when does instructional video facilitate learning? A commentary to the special issue "developments and trends in learning with instructional video". *Computers in Human Behavior* 89 (2018), 471–475.

[9] Janice M Bonner and William G Holliday. 2006. How college science students engage in note-taking strategies. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching* 43, 8 (2006), 786–818.

[10] John Boreczky, Andreas Girgensohn, Gene Golovchinsky, and Shingo Uchihashi. 2000. An interactive comic book presentation for exploring video. In *Proceedings of the SIGCHI conference on Human factors in computing systems.* 185–192.

[11] Mike Bostock. 2012. D3.js - Data-Driven Documents. http://d3js.org/

[12] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[13] Cynthia J. Brame. 2016. Effective Educational Videos: Principles and Guidelines for Maximizing Student Learning from Video Content. *CBE—Life Sciences Education* 15, 4 (2016), es6. https://doi.org/10.1187/cbe.16-03-0125 arXiv:https://doi.org/10.1187/cbe.16-03-0125 PMID: 27789532.

[14] Inc. Brightcove. 2015. Video.js - open source HTML5 & Flash video player. https://github.com/videojs/video.js.

[15] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods* 46, 3 (01 Sep 2014), 904–911. https://doi.org/10.3758/s13428-013-0403-5

[16] Dung C Bui, Joel Myerson, and Sandra Hale. 2013. Note-taking with computers: Exploring alternative strategies for improved recall. *Journal of Educational Psychology* 105, 2 (2013), 299.

[17] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. 2017. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 221–230.

[18] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. 2019. The 2019 DAVIS Challenge on VOS: Unsupervised Multi-Object Segmentation. *arXiv:1905.00737* (2019).

[19] Robert Carlson, Paul Chandler, and John Sweller. 2003. Learning and understanding science instructional material. *Journal of educational psychology* 95, 3 (2003), 629.

[20] Paul Chandler. 2004. The crucial role of cognitive processes in the design of dynamic visualizations. *Learning and Instruction* 14, 3 (2004), 353–357.

[21] Lin Chen, Jianbing Shen, Wenguan Wang, and Bingbing Ni. 2015. Video object segmentation via dense trajectories. *IEEE Transactions on Multimedia* 17, 12 (2015), 2225–2234.

[22] Kai-Yin Cheng, Sheng-Jie Luo, Bing-Yu Chen, and Hao-Hua Chu. 2009. Smartplayer: user-centric video fast-forwarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 789–798.

[23] Chekuri Choudary and Tiecheng Liu. 2007. Summarization of visual content in instructional videos. *IEEE Transactions on Multimedia* 9, 7 (2007), 1443–1455.

[24] Stamatia Dasiopoulou, Eirini Giannakidou, Georgios Litos, Polyxeni Malasioti, and Yiannis Kompatsiaris. 2011. *A Survey of Semantic Image and Video Annotation Tools.* Springer Berlin Heidelberg, Berlin, Heidelberg, 196–239. https://doi.org/10.1007/978-3-642-20795-2_8

[25] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. 2008. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 237–246.

[26] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *kdd,* Vol. 96. 226–231.

[27] Kenneth Forbus, Jeffrey Usher, Andrew Lovett, Kate Lockwood, and Jon Wetzel. 2011. CogSketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science* 3, 4 (2011), 648–666.

[28] Dan B Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M Seitz. 2008. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 3–12.

[29] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. arXiv 2015. *arXiv preprint arXiv:1512.03385* (2015).

[31] David Held, Devin Guillory, Brice Rebsamen, Sebastian Thrun, and Silvio Savarese. 2016. A Probabilistic Framework for Real-time 3D Segmentation using Spatial, Temporal, and Semantic Cues.. In *Robotics: Science and Systems*.

[32] Ken Hinckley, Xiaojun Bi, Michel Pahud, and Bill Buxton. 2012. Informal information gathering techniques for active reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1893–1896.

[33] Ken Hinckley, Shengdong Zhao, Raman Sarin, Patrick Baudisch, Edward Cutrell, Michael Shilman, and Desney Tan. 2007. InkSeine: In Situ search for active note taking. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 251–260.

[34] Tim N Höffler and Detlev Leutner. 2007. Instructional animation versus static pictures: A meta-analysis. *Learning and instruction* 17, 6 (2007), 722–738.

[35] Shruti Jadon and Mahmood Jasim. 2019. Video summarization using keyframe extraction and video skimming. *arXiv preprint arXiv:1910.04792* (2019).

[36] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. 2010. Forward-backward error: Automatic detection of tracking failures. In *2010 20th International Conference on Pattern Recognition*. IEEE, 2756–2759.

[37] Thorsten Karrer, Malte Weiss, Eric Lee, and Jan Borchers. 2008. Dragon: a direct manipulation interface for frame-accurate in-scene video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 247–250.

[38] Thorsten Karrer, Moritz Wittenhagen, and Jan Borchers. 2009. Pocketdragon: a direct manipulation video navigation interface for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–3.

[39] Tim Kuehl, Alexander Eitel, Gregor Damnik, and Hermann Koerndle. 2014. The impact of disfluency, pacing, and students' need for cognition on learning with multimedia. *Computers in Human Behavior* 35 (2014), 189–198.

[40] Mackenzie Leake, Hijung Valentina Shin, Joy O. Kim, and Maneesh Agrawala. 2020. Generating Audio-Visual Slideshows from Text Articles Using Word Concreteness. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376519

[41] Ville Lehtola, Heikki Huttunen, Francois Christophe, and Tommi Mikkonen. 2017. Evaluation of visual tracking algorithms for embedded devices. In *Scandinavian Conference on Image Analysis*. Springer, 88–97.

[42] James R Lewis. 1992. Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. In *Proceedings of the human factors society annual meeting*, Vol. 36. Sage Publications Sage CA: Los Angeles, CA, 1259–1260.

[43] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. 2019. A Simple Pooling-Based Design for Real-Time Salient Object Detection. In *IEEE CVPR*.

[44] Richard K Lowe. 1999. Extracting information from an animation during complex visual learning. *European journal of psychology of education* 14, 2 (1999), 225–244.

[45] Richard E Mayer. 1984. Aids to text comprehension. *Educational psychologist* 19, 1 (1984), 30–42.

[46] Richard E Mayer. 2002. Multimedia learning. In *Psychology of learning and motivation*. Vol. 41. Elsevier, 85–139.

[47] Richard E Mayer. 2005. Cognitive theory of multimedia learning. *The Cambridge handbook of multimedia learning* 41 (2005), 31–48.

[48] Richard E Mayer and Patricia A Alexander. 2016. *Handbook of research on learning and instruction*. Taylor & Francis.

[49] Xiaojun Meng, Shengdong Zhao, and Darren Edge. 2016. HyNote: Integrated Concept Mapping and Notetaking. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. 236–239.

[50] Martin Merkt, Anne Ballmann, Julia Felfeli, and Stephan Schwan. 2018. Pauses in educational videos: Testing the transience explanation against the structuring explanation. *Computers in Human Behavior* 89 (2018), 399–410.

[51] Leann J Mischel. 2019. Watch and learn? Using EDpuzzle to enhance the use of online videos. *Management Teaching Review* 4, 3 (2019), 283–289.

[52] Xiangming Mu. 2010. Towards effective video annotation: An approach to automatically link notes with video content. *Computers & Education* 55, 4 (2010), 1752–1763.

[53] Yair Even Or. 2017. Tagify - tags input component. https://github.com/yairEO/tagify.

[54] Rolf Ploetzner and Richard Lowe. 2012. A systematic characterisation of expository animations. *Computers in Human Behavior* 28, 3 (2012), 781–794.

[55] RCSBProteinDataBank. 2017. *What is a Protein? (from PDB-101)*. Youtube. https://www.youtube.com/watch?v=wvTv8TqWC48

[56] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 658–666.

[57] Lloyd P Rieber and Asit S Kini. 1991. Theoretical foundations of instructional applications of computer-generated animated visuals. *J. COMP. BASED INSTR.* 18, 3 (1991), 83–88.

[58] Gavriel Salomon. 2012. *Interaction of media, cognition, and learning: An exploration of how symbolic forms cultivate mental skills and affect knowledge acquisition*. Routledge.

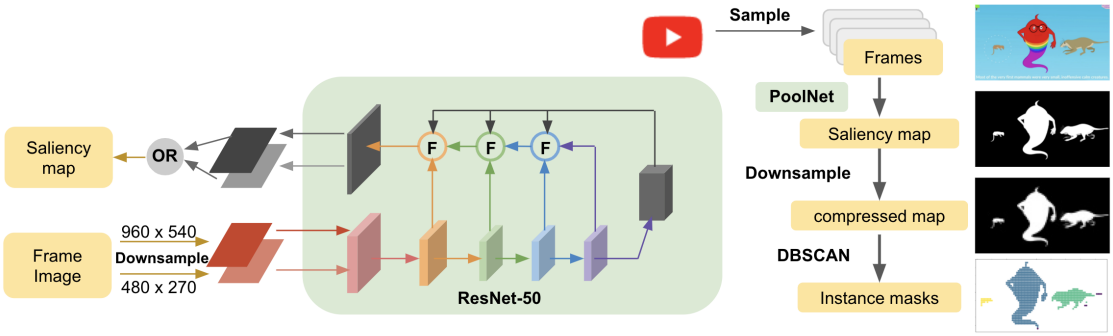

Fig. 7. Dynamic Visual Representative Object Generation Schema

[59] Klaus Schoeffmann, Marco A Hudelist, and Jochen Huber. 2015. Video interaction tools: A survey of recent work. *ACM Computing Surveys (CSUR)* 48, 1 (2015), 1–34.

[60] Abdulhadi Shoufan. 2019. Estimating the cognitive value of YouTube's educational videos: A learning analytics approach. *Computers in Human Behavior* 92 (2019), 450–458.

[61] Robert E Slavin. 2019. *Educational psychology: Theory and practice.*

[62] Hariharan Subramonyam, Colleen Seifert, Priti Shah, and Eytan Adar. 2020. texSketch: Active Diagramming through Pen-and-Ink Annotations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[63] Huib K Tabbers, Rob L Martens, and Jeroen JG Van Merriënboer. 2004. Multimedia instructions and cognitive load theory: Effects of modality and cueing. *British journal of educational psychology* 74, 1 (2004), 71–81.

[64] Craig S Tashman and W Keith Edwards. 2011. LiquidText: a flexible, multitouch environment to support active reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3285–3294.

[65] Frank Thomas, Ollie Johnston, and Frank Thomas. 1995. *The illusion of life: Disney animation.* Hyperion New York.

[66] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. 2002. Animation: can it facilitate? *International journal of human-computer studies* 57, 4 (2002), 247–262.

[67] Matthew Walsh. 2017. Video.js Transcript. https://github.com/walsh9/videojs-transcript.

[68] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. 2017. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 136–145.

[69] Greg Winslett. 2014. What counts as educational video?: Working toward best practice alignment between video production approaches and outcomes. *Australasian Journal of Educational Technology* 30, 5 (2014).

[70] Kuldeep Yadav, Ankit Gandhi, Arijit Biswas, Kundan Shrivastava, Saurabh Srivastava, and Om Deshmukh. 2016. Vizig: Anchor points based non-linear navigation and summarization in educational videos. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. 407–418.

[71] Eun-Mi Yang, Thomas Andre, Thomas J Greenbowe, and Lena Tibell. 2003. Spatial ability and the impact of visualization/animation on learning electrochemistry. *International Journal of Science Education* 25, 3 (2003), 329–349.

[72] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. 2019. Video object segmentation and tracking: A survey. *arXiv preprint arXiv:1904.09172* (2019).

[73] Ahmed Mohamed Fahmy Yousef, Mohamed Amine Chatti, Narek Danoyan, Hendrik Thüs, and Ulrik Schroeder. 2015. Video-mapper: A video annotation tool to support collaborative learning in moocs. *Proceedings of the Third European MOOCs Stakeholders Summit EMOOCs* (2015), 131–140.

[74] Lei Zhang, Qian-Kun Xu, Lei-Zheng Nie, and Hua Huang. 2014. VideoGraph: a non-linear video representation for efficient exploration. *The Visual Computer* 30, 10 (2014), 1123–1132.

A APPENDIX

A.1 Technical Details

A.1.1 Frame Object Extraction. As shown in 7, each frame image is sent to our neural network to derive a binary saliency map. For this step, we adapted the PoolNet[43] network structure. The architecture has high performing

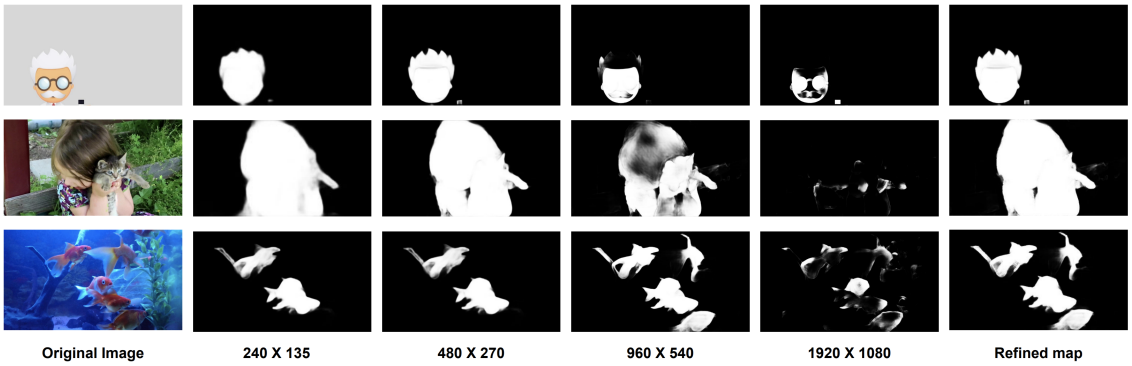

Fig. 8. Qualitative comparisons between image resolution and detected object

results for salient object detection (i.e., state-of-art at the time of implementation). The PoolNet authors describe several backbone choices and training options. We choose the pre-trained network that takes ResNet-50[30] as the backbone, the DUTS dataset [68] as a training set, and jointly trained the network with edge detection. These choices are made based on the best model performance reported in the PoolNet paper [43]

On top the PoolNet network, we add a 'refine' layer that conducts a pixel wise 'OR' operation over two saliency maps of a single frame image. That is, each frame image goes into the network twice with different resolutions ($960 \times 540p$ and $480 \times 270p$) and is combined with the refine layer. This layer was added based on the finding that the granularity of detected salient objects largely depends on original image resolution. We conducted an experiment over several graphic videos and the DAVIS-2017 [18] dataset. We found that images with lower resolution will result in a coarse capture of salient objects. Those with higher resolution will result in finer grained detection (see Figure 8). A combination allows us to benefit from both.

We aimed to extract objects that with three considerable dimensions: (1) *coverage*: the proper number of objects detected in the video, (2) *completeness*: objects are extracted in the whole without clipping (i.e., missing parts), and (3) *exactness*: objects are extracted along the right object contours without additional background pixels. We choose the resolutions of $960 \times 540p$ and $480 \times 270p$ for salient object detection based on the following observations: (1) $960 \times 540p$ maps are more comprehensive on the instance level (i.e., as many salient objects will be detected); (2) $480 \times 270p$ salient maps generate more compact and complete masks over objects. The bit-wise 'OR' adds up all salient regions that are detected in a frame image under these two resolutions.

We segmented objects in a frame by leveraging spatial gaps between the objects in the image. To do this, we down-sampled the saliency map to $96 \times 54p$. All pixels detected as 'on' (white object pixels in Figure 8) are clustered using the DBSCAN algorithm[26] to find those pixels that likely belong to the same object. We choose DBSCAN as it is unsupervised and can find arbitrarily sized and arbitrarily shaped clusters. Though not optimal as an image segmentation solution, we found it suitable for VideoSticker. Our goal is to capture the most 'interesting' component in the frame. This is distinct from trying to exactly find the shape of individual objects. For example, when 'atomic' objects interact or collide with each other, it is often the combination of these that is the object of interest for the sticker. Thus, it is reasonable to group them in a single sticker (e.g., instead of capturing a single nuclei in a video about fusion, we are more interested in its fusion process, or interaction with other nuclei). We found that DBSCAN does a good job at finding these 'meta' objects.

After clustering, the saliency map is segmented into different regions indicating different objects. These regions define the bounding boxes of each frame object. Each frame object will be extracted using the different masks and stored in a png image file.

A.1.2 Object Tracker. In our current prototype, object detection and object tracking (both needed for creating animated stickers) are treated as separate steps. Rather than pre-calculating the motion of all objects, the VideoSticker system currently responds to viewer's input. Specifically, the Object tracker begins to work after the viewer clicks on an object at $frame^{(i)}$ in the video. Base the selected object, we can find the bounding box containing that object ($BdBox_{trk}^{(i)}$). Taken this bounding box as a start point, the **Object Tracker** first uses the median-flow algorithm [36] (as implemented in OpenCV [12]) to track the object forward and backward and get a sequence of bounding boxes. The median-flow algorithm can quickly track the object based on point movement in the original selection over multiple frames. Although we could pre-calculate this, we find that in practice the tracking is extremely fast. This is confirmed by other work [41], that demonstrates that the tracker is performant on real-time tracking tasks with sequential frames and is good at reporting failures.

The output of this step is a range of bounding boxes for the tracked object: ($BdBox_{trk}^{(i-j)}, BdBox_{trk}^{(i+k)}$) (i.e., the object is reported to be exist in $frame_{(i-j)}$ to $frame_{(i+k)}$). Given this range, we iterate over $frame_{(i-j)}$ to $frame_{(i+k)}$ and compare the bounding boxes of each extracted objects in each frame (i.e, the objects detected in Frame Object Extraction stage) to connect the bounding boxes to our pre-calculated frame objects. A decision of equivalence between the sticker bounding box and tracked bounding box is made by the Intersection Over Union (IOU) metric [56]. This metric measures similarity between two shapes and is commonly used in object tracking tasks as a benchmark. For a $frame_{(i+s)}, s \in [-j, k]$, there may be multiple objects detected. We choose the object whose bounding box has the largest IOU with $BdBox_{trk}^{(i+s)}$ to be the subsequent object.

We increase our algorithm's robustness by restricting both the forward and backward tracking to be longer than three seconds: if the object loses track within three seconds, object with the largest IOU in the subsequent frame will be chosen. This method allows us to capture the transformations. For example, the protein folding process shown in Figure 9 . The median-flow tracker will lose track when the amino acids chain folds into a protein, since graphically, they are not the same objects. However, since the transformation happens within three seconds, this process will be completely included in the sticker. Moreover, this hybrid tracking method is useful in capturing the interactions into the sticker (see the following section).

We note that it is possible to train a single object extraction and tracking model. However, in experimenting with these we have found that they were extremely slow in current implementations. By utilizing a different extraction and tracking system we can pre-calculate some things (e.g., salient objects across frames) and interactively track only those that the user would like to create a sticker for. Additionally, the two approaches have complementary characteristics. Object extraction can create high quality masks while the median-flow algorithm identifies more rough bounding boxes. In contrast, if we only use the coordinates identified through object extraction, we might think there is a single object in the middle of the screen through the entire video. That is, all extracted objects have overlapping bounding boxes. Using the median-flow tracker allows us to identify when there is a new object. Finally, as we describe below, having two different bounding boxes—one produced by the extraction and another by the tracker—can help us identify critical interactions between objects.

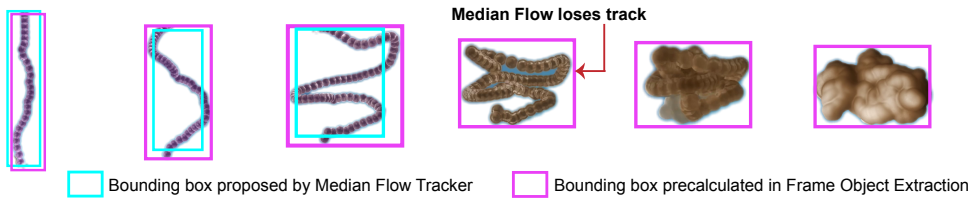

Fig. 9. Hybrid tracking method. The system will keep tracking based on the frame object bounding boxes if median-flow tracker loses track within 3s.

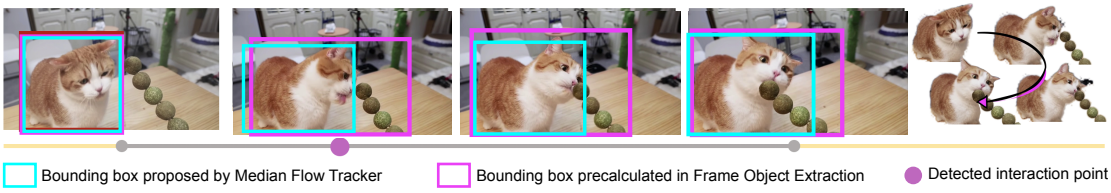

Fig. 10. Tracking and interest point detection. In the start frame, the 2 bounding boxes coincide. From the second frame, IOU drops due to interaction.

A.1.3 Interaction Point Detector. Figure 10 illustrates this approach with a video of a cat starting to eat a food. Depending on the kind of narrative we are trying to construct with stickers, we may want: one sticker for *both* cat and food; one sticker for the cat without food and one with; just a sticker of the cat before it interacted with the food; or even a combination (a sticker for the cat, a sticker for the food, and a sticker for the cat with food). To make it easier for the end-user to specify these we would like to identify key interaction points between objects. To do so, we leverage both the bounding boxes pre-identified by the extractor and those dynamically calculated by the tracker.

As a simple example, assume that in the first frame the end-user has clicked on the cat. VideoSticker has previously extracted a bounding box and mask for this object in that frame (the pink box). This becomes the input to the median-flow algorithm. The two boxes at this point are equivalent. Consider the backward tracking, the algorithm will advance to the next frames (second, third and fourth in the image). In all three of these frames, the object extraction has created one 'meta-object' for both the cat and the food. This is reasonable as the cat and food objects have 'interacted' to be become one (a cat eating a snack object). However, the median-flow algorithm, which tracks the movements of each points in the original selection will continue to focus on the cat (the blue boxes). The overlap between these two boxes can be used to identify possible interaction points. In the first frame, we calculate the IOU between the two boxes as 1. In each of the subsequent frames, the IOU is less as the two boxes are tracking two different things. We have found that when we see a significant frame-over-frame IOU drop (we found that 0.3 works well) we are seeing a likely interaction between objects. In our specific example, as the cat begins to eat the item (second frame) our IOU is calculated at 0.67 (a drop of 0.33). VideoSticker will note this time and will indicate this point visually on the range slider that is used to specify the temporal bounds of the sticker. While the cat is in all the frames (hence the slider extends across all the frames), the user can see the interaction point (represented by the pink dot) on the slider. Knowing this, they can move their selection to create a sticker for just the cat (from the start to the pink dot), the cat both before and during eating (from the start to the end of the range), or just the cat eating (from the pink dot to the end).

A.1.4 Label List Generator. To generate a set of plausible sticker labels we apply two different strategies. In some cases the name of the object appears in text in the video itself. As a pre-processing step, we use Google's Cloud Vision API OCR system on the video to extract these labels. The API provides recognized text, a time span, and bounding box. Because the OCR output is noisy, we can contrast the extracted terms to the transcript or closed caption text for the video. This allows us to either find the 'nearest match' in the text (e.g., if the OCR is 'aple' and we see 'apple' in the text) or discard words that do not seem to appear in the transcript at all.

Our second mechanism for extracting labels is to identify concrete words in the transcript text. We adapt Leake et al.'s system for finding these terms [40]. A concrete term is defined as a word that could be connected to find a perceptible concept (e.g., 'apple' or 'sun' rather than 'happiness'). In the original work, concrete words in an article were used to identify which images to use in a auto-generated video or slideshow. We apply this idea in reverse as we already have the video and transcript. Our goal is to identify those terms in the transcript that are most likely to be manifested in the video.

We largely apply the same algorithm used in the original system. We label each term in the transcript based on a lexical concrete score (derived from [15]). Using a dependency parser we identify all noun 'chunks' that assign a concreteness score to these based on the underlying words. All named entities (person, organization or geo-political entity) are also assumed to be concrete and retained. For both dependency parsing and named entity recognition we utilize the Spacy library. As a final step, we utilize a coreference resolution library[1] to replace ambiguous terms (e.g., 'it' in 'Mark picked up the apple. He ate it.') with concrete ones (e.g., 'apple' for 'it' and 'Mark' for 'He'). Terms that have been resolved are assigned a concreteness score based on the lexical scores. An optional pre-processing step uses a neural transformer architecture to re-integrate punctuation into the transcription. For this step we use the Python **rpunct** package (https://github.com/Felflare/rpunct). This step necessary when the transcription is done through automated speech-to-text rather than produced manually as closed captions.

When producing likely labels for specific stickers we use both the OCR text and concrete text extraction. These are pulled from a time window around when the object is selected and can be ranked both based on the concreteness score.

[1]We utilize the Allen NLP Library's module: https://github.com/allenai/allennlp

A.2 Sample Videos

The following videos are used as examples of VideoSticker note-taking and in our experiments.

Table 1. Sample Video List

Title	Source	Creator	Figures
[V1] Neutron Stars – The Most Extreme Things that are not Black Holes	https://www.youtube.com/watch?v=udFxKZRyQt4	Kurzgesagt – In a Nutshell	5
[V2] The Coronavirus Explained & What You Should Do	https://www.youtube.com/watch?v=BtN-goy9VOY	Kurzgesagt – In a Nutshell	3
[V3] MITOSIS, CYTOKINESIS, AND THE CELL CYCLE	https://www.youtube.com/watch?v=8uzHTKdv_Sw	Neural Academy	5
[V4] From DNA to protein - 3D	https://www.youtube.com/watch?v=gG7uCskUOrA	yourgenome	3, 6, 9
[V5] What is a virus? How do viruses work?	https://www.youtube.com/watch?v=7KXHwhTghWI	Nathan Winch	3, 8
[V6] Meiosis 1 - Class 11	https://www.youtube.com/watch?v=LUFYcqtMKf0	Uniclass Content	3
[V7] 1 HOUR of tutorials | Learn 35 football skills	https://www.youtube.com/watch?v=y1d__uHGQso	Unisport	6
[V8] The Biggest Mistake Beginners Make In Watercolour	https://www.youtube.com/watch?v=Uzw7RBZnuSw	Karen Rice	6
[V9] Evolution of Mammals	https://www.youtube.com/watch?v=N-Xs_mdrgic	Learning Junction	7
[V10] Stacked Ball Drop	https://www.youtube.com/watch?v=2UHS883_P60	Physics Girl	3
[V11] How Earth Moves	https://www.youtube.com/watch?v=IJhgZBn-LHg	Vsauce	3