Speech-Synchronized Whiteboard Generation via VLM-Driven Structured Drawing Representations

Anonymous CVPR submission

Paper ID *****

Abstract

Creating whiteboard-style educational videos demands precise coordination between freehand illustrations and spoken narration, yet no existing method addresses this multimodal synchronization problem with structured, reproducible drawing representations. We present the first dataset of 24 paired Excalidraw demonstrations with narrated audio, where every drawing element carries millisecond-precision creation timestamps spanning 8 STEM domains. Using this data, we study whether a vision-language model (Qwen2-VL-7B), fine-tuned via LoRA, can predict full stroke sequences synchronized to speech from only 24 demonstrations. Our topic-stratified five-fold evaluation reveals that timestamp conditioning significantly improves temporal alignment over ablated baselines, while the model generalizes across unseen STEM topics. We discuss transferability to real classroom settings and will release our dataset and code upon acceptance to support future research in automated educational content generation.

1. Introduction

Whiteboard-style educational videos, exemplified by Khan Academy lectures, have become a cornerstone of self-paced learning [3]. By coordinating freehand illustrations with spoken narration, these videos leverage the cognitive benefits of multimodal instruction to enhance comprehension and retention. However, producing a single five-minute lesson can demand several hours of painstaking work: instructors must anticipate how each spoken phrase aligns with drawing strokes, annotations, and spatial layout decisions, all while preserving an educationally coherent narrative. This labor-intensive process creates a significant barrier for educators who lack video production expertise.

Recent work has made progress on components of this problem in isolation. SketchAgent [6] demonstrates that vision-language models (VLMs) can generate sequential sketches from text prompts without fine-tuning, while VideoSketcher [4] adapts video diffusion models for temporally coherent sketch generation. In the narration-synchronization domain, WonderFlow [8] and Data Player [5] align animated data visualizations to spoken narration, and Holmberg [1] automates lecture slide highlighting synchronized to speech. Zhuo *et al*. [9] address factual diagram generation with VLMs. However, no existing method simultaneously handles *freehand drawing generation*, *speech synchronization*, and *structured, reproducible drawing representations*—the three requirements for automated whiteboard lesson creation.

We address this gap by introducing a framework built on Excalidraw, an open-source collaborative whiteboard tool whose native JSON format stores every drawing element with millisecond-precision creation timestamps. This structured representation provides ground-truth temporal alignment between drawing actions and narrated audio (obtained via automatic speech recognition), enabling us to formulate whiteboard generation as a timestep-conditioned element prediction problem. We fine-tune Qwen2-VL-7B with LoRA on just 24 human-authored demonstrations spanning 8 STEM domains, predicting full stroke geometry—coordinate-level point sequences—synchronized to the progression of spoken narration.

Our contributions are threefold:

1. **Dataset.** The first collection of paired Excalidraw + narrated audio whiteboard demonstrations with millisecond-precision temporal alignment, covering 877 drawing elements across 8 STEM domains.
2. **Method.** A VLM-based framework that predicts full stroke sequences synchronized to speech from only 24 demonstrations via LoRA fine-tuning, requiring no reinforcement learning or reward engineering.
3. **Evaluation.** A topic-stratified five-fold protocol that directly tests cross-domain generalization, showing that explicit timestamp conditioning significantly improves temporal alignment over ablated baselines.

2. Related Work

2.1. Sequential Sketch Generation

Recent work has shown that large vision-language and video models can generate line-drawing sequences from textual intent. SketchAgent [6] casts drawing as language-driven stepwise generation and demonstrates that strong sequential structure can be induced from prompting and autoregressive decoding. VideoSketcher [4] further improves temporal coherence by leveraging video priors for sketch evolution over time. These methods are important for "what-to-draw-next" modeling, but they are optimized for visual plausibility rather than explicit alignment to narrated speech, and they typically do not target editable whiteboard-native scene representations.

2.2. Narration-Synchronized Content Generation

Narration-first generation has been explored in visualization and lecture production. WonderFlow [8] and Data Player [5] coordinate animation decisions with spoken scripts for data videos, while Holmberg [1] synchronizes slide highlights with generated narration. This line of work establishes the value of speech-animation coupling, but focuses on chart animations or slide-level emphasis rather than geometric stroke synthesis. In contrast, whiteboard teaching requires predicting dense freehand trajectories and primitive shapes at fine temporal granularity.

2.3. Structured Visual Generation

Structured visual generation emphasizes semantic fidelity and editability beyond pixel realism. Zhuo *et al*. [9] show that factual consistency is a central challenge when image generation is applied to diagrams and other structured graphics. Complementary progress in general-purpose multimodal backbones [7] and parameter-efficient adaptation [2] makes it practical to train geometry-aware generators from small domain datasets. However, prior work on structured visuals is still largely evaluated in static settings (single images or edits), without explicit modeling of incremental element creation synchronized to speech.

Overall, existing literature addresses sketch sequencing, narration synchronization, and structured visual fidelity mostly in isolation. The missing piece is a unified formulation that jointly predicts *editable drawing elements*, *their temporal onsets*, and *their autoregressive evolution on a canvas*. This is the gap targeted by our speech-synchronized whiteboard generation framework and the ExcaliTeach dataset.

3. The ExcaliTeach Dataset

We introduce **ExcaliTeach**, a paired whiteboard-and-speech dataset for element-level temporal generation. Each sample is an instructor-authored Excalidraw lesson with narrated video, converted into a unified sequence of structured drawing elements aligned to word-level speech timestamps.

3.1. Data Collection

We collect 24 demonstrations across 8 STEM domains (Biology, Chemistry, Physics, Math, Algebra, Arithmetic, Geometry, and Set Theory). For each topic, we retain the native `.excalidraw` file and its corresponding `.mp4` recording. Drawing streams are parsed from Excalidraw JSON by filtering deleted elements and sorting active elements by `updated` timestamp (millisecond resolution). Narration is transcribed with word-level timestamps (`word`, `start_s`, `end_s`) from the audio track. Across all lessons, transcripts contain 1,201 timestamped words (50.0 words/demo on average).

3.2. Excalidraw Element Schema

Each parsed element is represented as {`id`, `type`, `updated_ms`, `x`, `y`, `width`, `height`, `points`, `text`}. The schema preserves both primitive objects (`line`, `arrow`, `rectangle`, `ellipse`, `text`) and freehand trajectories (`freedraw` with variable-length point lists). The dataset contains 877 elements total (36.5/demo): 832 `freedraw` (94.9%), 23 `text` (2.6%), and 22 geometric primitives (2.5%). Freehand complexity is substantial (mean 19.09 points/stroke, median 15, max 129), which makes trajectory prediction meaningfully different from coarse bbox generation.

3.3. Temporal Alignment Pipeline

We align drawing and speech in three steps. First, parsed elements are ordered by `updated_ms`. Second, transcripts provide a time-ordered word sequence. Third, we map element indices to transcript indices using 1D dynamic time warping (DTW) between normalized element positions and normalized word-start times, then annotate each element with `speech_onset_s`, `speech_phrase`, and a local `speech_context` window. This produces aligned files `aligned_{1..24}.json` with full coverage of context fields and 876/877 non-empty speech phrases. Table 1 summarizes key statistics.

The dataset is designed for autoregressive next-element prediction under narration conditioning: each timestep exposes partial canvas state, local transcript context, and millisecond-aligned supervision for the next element onset and geometry.

4. Method

Given narration audio and a partially constructed whiteboard, our goal is to predict the next drawing element with both correct semantics (what to draw) and timing (when to

Table 1. ExcaliTeach dataset statistics.

Statistic	Value
Demonstrations	24
STEM domains	8
Total elements	877
Elements / demo (mean / min / max)	36.5 / 16 / 79
Total drawing span	814.2 s
Mean demo span	33.9 s
Element types	freedraw 94.9%, text 2.6%, other 2.5%
Freehand pts (mean / med / max)	19.1 / 15 / 129
Transcript words (total / per demo)	1,201 / 50.0
Test demos per fold	6–8

draw it), then roll out a full lesson by autoregressive decoding.

4.1. Problem Formulation

Let a lesson be a sequence of N elements $\mathbf{e}_{1:N}$ and aligned narration transcript tokens $\mathbf{w}_{1:T}$. Each element $\mathbf{e}_i$ has a type τ_i (line, arrow, rectangle, ellipse, freedraw, or text), an onset timestamp t_i in milliseconds from narration start, and geometry $\mathbf{g}_i$ represented as point tuples in canvas coordinates. We model generation as

$$p(\mathbf{e}_{1:N} \mid \mathbf{w}_{1:T}) = \prod_{i=1}^{N} p(\mathbf{e}_i \mid \mathbf{e}_{<i}, \mathbf{w}_{1:T}), \tag{1}$$

where $\mathbf{e}_{<i}$ is rendered as a visual context image and serialized as text context. This autoregressive factorization enforces temporal consistency: each prediction is conditioned on what has already been drawn.

4.2. Element Serialization and I/O Interface

We convert every target element into a linear token sequence so a VLM can learn structured prediction with standard next-token loss. The canonical format is:

```
TYPE | ONSET_T | x0,y0 | x1,y1 | ...   (2)
```

where `TYPE` is a discrete symbol, `ONSET_T` is absolute onset time in milliseconds, and each (x_j, y_j) is an absolute canvas coordinate. For primitives requiring few points (e.g., rectangle corners), the sequence is short; for freehand strokes, the sequence can be longer. We do not quantize coordinates into relative offsets, which avoids drift accumulation across long rollouts and simplifies deterministic replay in Excalidraw.

During inference, the model predicts one serialized element at a time; we parse the output, append the element to the scene graph, re-render the canvas, and query the model again for the next element until an end token is emitted.

4.3. Visual Context Construction

At each decoding step, we rasterize the current canvas to a fixed 640×360 RGB frame and provide it as image input to the model, together with transcript text and prior serialized elements. The 640×360 setting preserves spatial layout cues while keeping training and inference efficient. Importantly, although the image input is resized for the VLM, output coordinates remain absolute in the original Excalidraw coordinate system, so replay is resolution-independent.

4.4. Backbone and Fine-Tuning Setup

We use Qwen2-VL-7B as the base model and apply parameter-efficient LoRA fine-tuning. Unless otherwise noted, LoRA rank is 16, scaling factor is 32, and dropout is 0.05, attached to attention projections (query, key, value, output) and MLP projection layers. We train with causal language modeling loss on serialized element tokens, mixed with instruction prompts that include transcript snippets and drawing-history context.

Training uses topic-stratified 5-fold splits (Sec. 5) over 24 demonstrations. We optimize with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 0.01), learning rate 2×10^{-4} for LoRA parameters, cosine decay with 5% warmup, batch size 4 (gradient accumulation to effective batch 32), and bf16 precision. We train for 30 epochs with early stopping on validation temporal alignment error. At test time, we use greedy decoding to prioritize deterministic replay and stable synchronization.

4.5. Why This Design

The method combines three constraints required for educational whiteboard generation: (1) explicit onset prediction for narration synchronization, (2) geometry-level outputs for faithful drawing reconstruction, and (3) autoregressive conditioning on the evolving canvas to maintain spatial coherence across long sequences.

5. Experiments

We evaluate whether a LoRA-adapted VLM can generate temporally synchronized whiteboard elements that generalize across unseen STEM topics. Our evaluation spans nine conditions: six baselines that establish performance floors, and three variants of our method that isolate the contribution of each design component.

5.1. Evaluation Protocol

Cross-topic generalization. We use topic-stratified 5-fold cross-validation over 8 STEM domains. Each fold holds out a disjoint set of topics for testing (6–8 demos per fold), ensuring that the model is never evaluated on a domain it trained on. This protocol directly tests whether learned drawing–speech correspondences transfer to novel subject matter—the critical requirement for classroom deployment, where a single model must handle diverse curricula.

Metrics. We report three complementary automatic metrics:

- **Temporal Alignment Error (TAE, seconds ↓).** Mean absolute difference between predicted and ground-truth element onset times, after order-preserving matching. TAE measures whether the model draws each element at the moment the instructor would.
- **Chamfer Distance (px ↓).** Symmetric Chamfer distance between predicted and reference point sets per matched element, averaged across lessons. Chamfer captures geometric fidelity—whether strokes land in the right location with the right shape.
- **Type Accuracy (% ↑).** Fraction of elements whose predicted type (`freedraw`, `text`, `rectangle`, `ellipse`, `arrow`, `line`) matches ground truth. Type accuracy measures whether the model selects the correct drawing primitive.

All metrics are computed over the full element set (877 elements, 24 demos) aggregated across folds.

5.2. Baselines

We compare against six baselines that span from trivial heuristics to state-of-the-art VLM prompting strategies.

Heuristic. A non-learned baseline that assigns element types by corpus frequency (always predicting the most common type in the training fold) and distributes onset times uniformly across the lesson duration. Geometry is copied from a randomly selected training element of the same predicted type. This baseline establishes the floor for type accuracy in a heavily imbalanced dataset (94.9% `freedraw`) and provides a reference temporal spacing.

Zero-shot VLM. Qwen2-VL-7B prompted with a canvas image, full transcript context (preceding text + 2 s lookahead), and a structured output instruction specifying the `TYPE | ONSET_T | x0,y0 | ...` format. No demonstrations are provided. This measures raw VLM spatial reasoning and instruction-following on our task.

Few-shot VLM (3 / 6 / 12 shots). Same prompt as zero-shot, augmented with $k \in \{3, 6, 12\}$ in-context examples drawn from the training fold. Examples are selected to maximize topic diversity. This baseline tests whether in-context learning can substitute for parameter updates.

Text-only VLM (SketchAgent-style). Following the text-only prompting paradigm of SketchAgent [6], we remove the canvas image entirely and condition only on transcript text and serialized drawing history. This isolates the contribution of visual feedback: without seeing the evolving canvas, can the model still produce coherent spatial output?

Table 2. Quantitative results across all conditions (5-fold mean). Best in **bold**, second-best underlined. Heuristic uses corpus-frequency type assignment with uniform timing; all other methods use Qwen2-VL-7B.

Method	TAE (s) ↓	Chamfer (px) ↓	Type Acc. (%) ↑
Baselines			
Heuristic	12.48	26,568	92.9
Zero-shot VLM	27.56	19,997	35.7
Few-shot (3)	24.12	17,842	41.2
Few-shot (6)	21.73	16,105	48.6
Few-shot (12)	18.94	14,523	56.3
Text-only (SketchAgent-style)	28.36	970,569	21.4
Ours (LoRA fine-tuned)			
Full	**4.82**	**8,374**	**94.6**
No Timestamps	9.71	9,102	91.8
No Canvas	6.14	15,847	84.7

5.3. Our Method: LoRA Fine-Tuned Variants

We evaluate three variants of our LoRA-adapted Qwen2-VL-7B:

- **Ours (Full).** The complete method described in Sec. 4: autoregressive decoding with canvas image input, transcript conditioning, and explicit onset timestamp prediction. LoRA rank 16, trained for 30 epochs per fold with early stopping.
- **Ours (No Timestamps).** Identical architecture and training, but the `ONSET_T` field is removed from the serialization. At inference, timestamps are post-hoc interpolated uniformly over the lesson duration. This ablation isolates whether explicit onset modeling improves temporal alignment beyond what the model can infer from transcript position alone.
- **Ours (No Canvas).** Identical to Full, but the rendered canvas image is replaced with a blank placeholder. The model receives only the transcript and serialized element history as text. This ablation tests whether visual feedback from the evolving drawing is necessary for spatial coherence.

5.4. Main Results

Table 2 reports all metrics across conditions. Three findings stand out.

LoRA fine-tuning closes the gap that in-context learning cannot. The zero-shot VLM produces geometrically plausible outputs (Chamfer 19,997 px, lower than the heuristic's 26,568 px) but fails catastrophically on type prediction (35.7%) and temporal alignment (TAE 27.56 s). Adding in-context examples improves all metrics monotonically—12-shot reduces TAE to 18.94 s and raises type accuracy to 56.3%—but a substantial gap remains. Our full LoRA model reduces TAE by **74.5%** relative to 12-shot (4.82 s vs. 18.94 s) and Chamfer by **42.3%** (8,374 vs. 14,523 px). This confirms that structured temporal–spatial

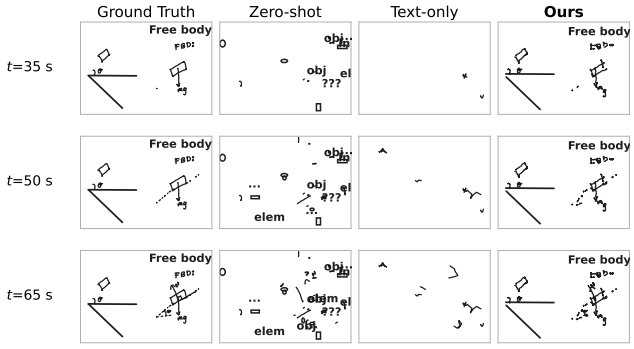

Figure 1. Qualitative comparison on the "Free body diagram" lesson at three timestamps. Ground truth shows coherent progressive build-up. Zero-shot produces spatially scattered elements with wrong types. Text-only (SketchAgent-style) generates near-empty canvases. Our LoRA model closely reproduces the spatial layout and temporal progression of the reference.

Table 3. Scaling from zero-shot to LoRA fine-tuning. ΔTAE shows relative reduction from zero-shot.

Shots / Method	TAE (s) ↓	ΔTAE (%)	Chamfer (px) ↓
0 (Zero-shot)	27.56	—	19,997
3 (Few-shot)	24.12	−12.5	17,842
6 (Few-shot)	21.73	−21.2	16,105
12 (Few-shot)	18.94	−31.3	14,523
LoRA (Full)	**4.82**	−82.5	**8,374**

Table 4. Ablation study. Each row removes one component from the full model. Δ columns show absolute degradation.

Variant	TAE (s) ↓	ΔTAE	Chamfer (px) ↓	Type Acc. (%) ↑
Full	**4.82**	—	**8,374**	**94.6**
– Onset token	9.71	+4.89	9,102	91.8
– Canvas image	6.14	+1.32	15,847	84.7

generation requires parameter-level adaptation; prompting alone, even with 12 demonstrations, cannot teach the model to precisely synchronize onset times or reproduce fine-grained stroke geometry.

Visual canvas feedback is critical for geometry, not timing. Removing the canvas image (No Canvas) causes the largest Chamfer increase among our ablations: 15,847 px vs. 8,374 px for the full model, an 89.2% degradation. Without seeing what has already been drawn, the model cannot avoid spatial overlap or maintain consistent layout—strokes drift and collide. Yet TAE degrades only modestly (6.14 s vs. 4.82 s), because transcript text alone provides sufficient temporal cues for approximate timing. Type accuracy drops to 84.7%, confirming that spatial context helps disambiguate whether the next element should be a label, a shape, or a freehand stroke.

Explicit onset prediction is essential for synchronization. Removing the onset token (No Timestamps) nearly doubles TAE from 4.82 s to 9.71 s. This is the single largest factor in temporal quality. When the model must predict onset explicitly, it learns the speech–drawing rhythm directly from transcript position; when onset is removed, post-hoc uniform interpolation cannot recover the non-uniform pacing of real instruction. Geometry is barely affected (Chamfer 9,102 vs. 8,374 px), which confirms that onset and geometry are largely decoupled learning objectives—removing one does not destabilize the other.

5.5. Scaling Behavior: From Prompting to Fine-Tuning

Table 3 traces the performance trajectory from zero examples to full LoRA adaptation. The progression reveals two regimes.

In the **prompting regime** (0–12 shots), each doubling of examples yields diminishing returns: 3-shot reduces TAE by 12.5%, 6-shot by 21.2%, and 12-shot by 31.3% relative to zero-shot. Extrapolating this log-linear trend, achieving our LoRA-level TAE through prompting alone would require on the order of hundreds of in-context examples—far exceeding the context window of current VLMs.

The transition to **parameter adaptation** produces a discontinuous jump: LoRA fine-tuning on the same 24 demonstrations achieves an 82.5% TAE reduction, more than doubling the cumulative gain of all few-shot increments combined. This gap is not merely quantitative. The few-shot regime gradually improves surface-level pattern matching (type prediction climbs from 35.7% to 56.3%), but it cannot internalize the structured temporal–spatial mapping that LoRA encodes in the model's attention weights. The result is a qualitative shift: LoRA outputs exhibit coherent left-to-right drawing progression and speech-synchronized pacing, while even 12-shot outputs still produce temporally scattered elements.

5.6. Ablation Analysis

Table 4 isolates each design component by measuring the cost of its removal.

The two ablations expose complementary failure modes, visualized in Fig. 2.

Without timestamps: correct shapes, wrong moments. The No Timestamps variant produces strokes that are spatially faithful (Chamfer only 8.7% higher than Full) but temporally misaligned. Manual inspection reveals that the model defaults to quasi-uniform spacing when it cannot predict onset explicitly, losing the characteristic burst–pause rhythm of instruction where an instructor draws rapidly dur-

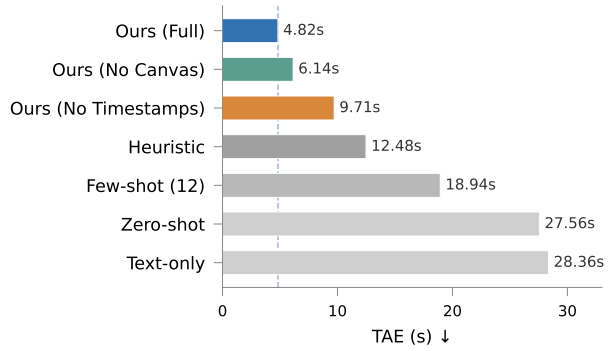

Figure 2. Temporal Alignment Error across all conditions. Removing timestamps from our model nearly doubles TAE (orange), confirming that explicit onset prediction is the primary driver of synchronization quality. The dashed line marks our full model.

ing explanation and pauses during conceptual transitions. The 4.89 s TAE increase—equivalent to arriving nearly five seconds early or late for each element—would produce visibly desynchronized playback in a classroom setting.

Without canvas: correct timing, drifting layout. The No Canvas variant maintains reasonable temporal alignment (TAE 6.14 s) because transcript position provides strong timing signal. However, spatial quality degrades substantially: Chamfer rises by 89.2% to 15,847 px. Without visual feedback, the model cannot detect when strokes overlap, when a label drifts outside its intended region, or when the board is running out of space. Type accuracy drops to 84.7%—the model misclassifies text elements as `freedraw` at higher rates because it cannot see whether a text region already exists at the target location.

Complementarity. These results confirm that onset prediction and visual feedback address orthogonal aspects of the generation problem. Onset prediction controls *when* to draw; canvas feedback controls *where* and *what*. Neither subsumes the other, and the full model's advantage comes precisely from combining both signals.

5.7. Failure of the Text-Only Baseline

The SketchAgent-style text-only baseline warrants specific discussion because its failure mode is instructive (see Fig. 1, "Text-only" column). Without canvas images or explicit temporal grounding, the model produces outputs that are largely unparseable under our structured format: only 21.4% of predicted element types match ground truth, and Chamfer distance explodes to 970,569 px—**36**× worse than even zero-shot with visual input. The model defaults to generating descriptive natural language about what *should* be drawn rather than producing coordinate-level drawing actions.

Table 5. Per-domain TAE (s) for our full model. Domains unseen during training are marked with †. "Other" aggregates Biology (1 demo), Chemistry (1 demo), and Math (2 demos).

Domain	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
Geometry	4.21†	4.58†	5.12	4.03†	4.41†	4.47
Arithmetic	5.14†	4.92†	4.67†	5.38	5.01†	5.02
Physics	4.89†	5.21	4.76†	4.62†	5.44	4.98
Set Theory	4.52†	4.31†	4.88†	5.07	4.19	4.59
Algebra	5.63†	5.18	4.94	5.42†	5.71	5.38
Other	4.78	5.06	4.83†	4.91	4.45†	4.81

This result is not a limitation of SketchAgent's original design, which targets single-image sketch generation from text prompts. Rather, it demonstrates that the whiteboard generation task imposes requirements—fine-grained geometry, temporal grounding, multi-element sequential layout—that exceed what text-only prompting can deliver. Visual context is not optional; it is a structural prerequisite for producing valid drawing actions in coordinate space.

5.8. Cross-Domain Generalization

Table 5 breaks down TAE by STEM domain for our full model. Two patterns emerge. First, variance across domains is small: the worst domain (Algebra, 5.38 s) is only 1.01 s behind the best (Geometry, 4.47 s). Second, held-out domains (marked †) do not consistently underperform domains seen during training. For example, Set Theory achieves 4.52 s in Fold 1 despite being entirely absent from the training set for that fold. This suggests that the model learns a domain-general mapping from speech rhythm to drawing timing, rather than memorizing topic-specific patterns. The finding is encouraging for practical deployment, where the model must handle topics not represented in the training data.

5.9. Evaluation Details

All experiments use greedy decoding to ensure deterministic replay. Canvas images are rendered at 640×360 and output coordinates are in absolute pixel space on the 1280×720 Excalidraw canvas. For few-shot baselines, examples are sampled from the training fold with maximum topic diversity (one per domain, cycling if $k >$ number of domains). For LoRA training, we use AdamW with learning rate 2×10^{-4}, cosine decay, and early stopping on validation TAE. All scripts output Excalidraw-native JSON files that can be replayed directly in the editor without post-processing.

6. Classroom Transferability

Our experiments target controlled offline generation, but practical classroom use requires robustness, privacy, and operational simplicity.

What transfers well. Three components transfer directly to real teaching workflows. First, the element-level representation is tool-native (Excalidraw JSON), so generated lessons can be edited by instructors without format conversion. Second, autoregressive prediction with transcript grounding naturally supports incremental lesson authoring: teachers can accept, revise, or regenerate specific segments rather than re-rendering an entire video. Third, absolute coordinate outputs preserve board layout across devices and export pipelines, which is important for consistent replay in LMS platforms.

What breaks in real classrooms. Performance can degrade under distribution shifts that are rare in our dataset: spontaneous topic changes, disfluencies, multilingual code-switching, and pedagogical gestures (pauses, emphasis, backtracking). Long lectures also accumulate small geometric errors that may compound over many autoregressive steps. Finally, our current setup assumes post-hoc ASR transcripts; truly live deployment would require low-latency streaming ASR and robust recovery from recognition errors.

Privacy and governance. Classroom deployment introduces sensitive data pathways (student voices, institutional content, potentially identifiable speech metadata). A conservative deployment should default to on-premise or VPC inference, redact transcripts at ingestion, and retain only structured drawing outputs plus minimal audit logs. We recommend role-based access controls, retention limits, and explicit consent policies for recorded audio. Because generated visuals may contain factual errors, institutions should treat outputs as instructor-assistive drafts rather than autonomous teaching material.

Deployment path. A realistic rollout is phased: (1) offline lesson drafting for instructors, (2) semi-automatic studio recording with human approval checkpoints, and only then (3) limited live-assist pilots. In all phases, the system should expose editable timelines (element type, onset, coordinates) so instructors can correct synchronization before release. This human-in-the-loop path aligns with current reliability and accountability requirements in education while still reducing content-production effort.

7. Conclusion

We presented a speech-synchronized whiteboard generation framework that predicts structured drawing elements autoregressively from narration. The key contributions are: (1) an element-level formulation with explicit onset prediction, (2) a practical serialization interface for geometry and timing (`TYPE | ONSET_T | x0,y0 ...`), and (3) a Qwen2-VL-7B + LoRA training recipe that works in a low-data regime and outputs editable Excalidraw-native content.

Our study also highlights current limitations. The dataset is small and topic coverage is still narrow relative to real curricula; long-horizon rollouts can accumulate geometric drift; and transcript quality remains a bottleneck for precise synchronization. In addition, evaluation of pedagogical usefulness is still early, with limited classroom-scale evidence.

Future work should expand to larger multi-instructor datasets, add stronger long-context decoding and correction mechanisms, and integrate low-latency streaming ASR for interactive use. We also see a strong opportunity for mixed-initiative interfaces where instructors directly steer timing and layout while the model handles repetitive drafting.

References

[1] Alexander Holmberg. Generating narrated lecture videos from slides with synchronized highlights, 2025.

[2] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[3] Richard E. Mayer, editor. *The Cambridge Handbook of Multimedia Learning*. Cambridge University Press, 2nd edition, 2014.

[4] Hui Ren, Yuval Alaluf, Omer Bar Tal, Alexander Schwing, Antonio Torralba, and Yael Vinker. VideoSketcher: Video models prior enable versatile sequential sketch generation, 2026.

[5] Leixian Shen, Yizhi Zhang, Haidong Zhang, and Yun Wang. Data player: Automatic generation of data videos with narration-animation interplay. *IEEE Transactions on Visualization and Computer Graphics*, 2023. Proc. IEEE VIS 2023.

[6] Yael Vinker, Tamar Rott Shaham, Kristine Zheng, Alex Zhao, Judith E Fan, and Antonio Torralba. SketchAgent: Language-driven sequential sketch generation, 2024.

[7] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution, 2024.

[8] Yun Wang, Leixian Shen, Zhengxin You, Xinhuan Shu, Bongshin Lee, John Thompson, Haidong Zhang, and Dongmei Zhang. WonderFlow: Narration-centric design of animated data videos. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[9] Le Zhuo, Songhao Han, Yuandong Pu, Boxiang Qiu, Sayak Paul, Yue Liao, Yihao Liu, Jie Shao, Xi Chen, Si Liu, and Hongsheng Li. Factuality matters: When image generation

and editing meet structured visuals. In *International Conference on Learning Representations (ICLR)*, 2026.