A Multimodal Predictive Framework for Temporal Scene Synthesis via Video-Audio-Text Alignment and Perceptual Feedback

Abstract This work describes a temporal scene synthesis system that predicts the next video frame using synchronized video, audio, text, and perceptual state. Video frames are encoded with a Variational Autoencoder (VAE). Audio is embedded from aligned windowed speech. Text is tokenized into a dense control language that captures action, lighting, position, speaker identity, and scene state. In parallel, nine temporal perceptual metrics are computed from regions of interest (ROI) in each frame to measure exposure, motion, entropy, gaze, and structural continuity over time. These streams are fused and passed through a staged recurrent architecture. The network is trained to predict the next latent frame and is scored against ground truth. After a single title converges to perfect next-frame prediction, training expands across multiple titles to drive generalization. Once generalization stabilizes, weights are frozen and the model can synthesize new shots from prompt control alone.

1. System Goal The objective is not open-ended “video generation.” The objective is frame-accurate temporal prediction. The model learns how a world evolves, not just how it looks. After convergence the model can be prompted to render novel intermediate moments that obey the learned temporal physics.

2. Data Stream For each timestep t we build a multimodal vector X_t: • Video: A frame F_t is encoded by a VAE encoder into latent Z_t. • Audio: The aligned audio slice A_t (e.g. Mel/MFCC from Whisper-aligned speech). • Text prompt: The aligned transcript tokenized into structured control tokens. Example token classes: action_, cam_, light_style_, obj_, body_, conn_, dir_, state_, Speaker_#. These tokens compress scene semantics into machine-stable IDs and remove natural language noise. • Perceptual metrics P_t: Nine temporal sensory channels derived from ROI analysis: 1. Exposure: Exponential moving average (EMA) of mean luminance. 2. Contrast Drift: Change in standard deviation of intensity. Tracks tone shifts. 3. Color Balance Shift: Magnitude of RGB channel offset. Captures warmth / tint changes. 4. Motion Flow Magnitude: Average optical-flow magnitude. Captures movement energy. 5. Edge Persistence: Correlation of Sobel edge maps between frames. Captures structural continuity. 6. Spatial Entropy: Intensity histogram entropy. Captures visual complexity / clutter. 7. Flicker Frequency: Rate of luminance oscillation. Captures lighting instability and strobe effects. 8. Temporal Coherence: Pearson correlation between consecutive frames. Captures global scene stability. 9. Gaze Density: Concentration of fixation heatmap in the ROI. Captures focus of visual attention.

Each metric produces a 1-D signal S_t^i. We concatenate these signals into P_t = [S_t^1 … S_t^9]. The final per-frame model input is: X_t = [Z_t , A_t , T_t , P_t]

3. Temporal Architecture The model uses three recurrent stages:

3.1 Pattern LSTM Input: X_t. Role: learns short-horizon temporal structure across modalities. It captures motion rhythm, dialogue timing, lighting fluctuation, and camera behavior.

3.2 Compression LSTM Input: hidden state from Pattern LSTM. Role: produces a compact latent C_t. This is a bottleneck memory that encodes “what is actually happening right now” with minimal redundancy. It acts like short-term working memory across several frames.

3.3 Central LSTM (Predictive Core) Input: C_t. Output: predicted latent for the next frame, Z_hat_(t+1). This predicted latent is decoded by the VAE decoder to produce F_hat_(t+1), the predicted next frame.

4. Training Loop Step 1. Extract aligned data: • Video frames at target FPS. • Whisper-aligned transcript with per-word timestamps. • Audio chunks aligned to each frame window. • Perceptual metrics P_t from ROI analysis. • Scene tokens from the scripted descriptor pipeline.

Step 2. Forward pass: • Build X_t for each timestep. • Run Pattern LSTM → Compression LSTM → Central LSTM. • Decode Z_hat_(t+1) to F_hat_(t+1).

Step 3. Loss: • Latent reconstruction loss: MSE(Z_t+1 , Z_hat_t+1). • Perceptual frame loss: LPIPS / VGG-style distance between F_t+1 and F_hat_t+1. • Audio continuity loss: cosine(A_t+1 , A_hat_t+1) if the model also predicts next-step audio embedding. • Text alignment loss: cross-entropy between next-step control tokens and predicted next-step control tokens if predicted. • Temporal stability penalty: deviation between predicted P_hat_t+1 (optional head) and observed P_t+1. Total loss is a weighted sum. The fitness score for a timestep is 1 - ||Z_t+1 - Z_hat_t+1||.

Step 4. Iterate until next-frame prediction on the source video approaches perfect accuracy. At convergence the model can generate F_hat_(t+1) that is visually indistinguishable from F_t+1 using only the prompt/audio context plus recent state.

5. Generalization Phase After convergence on one video: • Introduce additional videos with different visual styles, lighting regimes, pacing, and dialogue rhythm. • Continue training using the same architecture and loss. • Goal: force the model to separate “style” (palette, camera bias, acting style) from “physics” (temporal continuity, causal progression). • Criterion: the model must achieve the same near-perfect next-frame prediction on each title independently.

This step prevents trivial memorization and teaches transferable temporal laws.

6. Freeze and Synthesis When cross-video generalization is stable: • Freeze weights. • Drive the system using mixed prompts. For example, feed lighting/camera tokens from Video A, dialogue rhythm from Video B, and structural motion pacing from Video C. • The Central LSTM now acts as a causal scene synthesizer. It renders new in-between states that never appeared in any source, but still obey the learned temporal physics.

This is not diffusion sampling. This is controlled temporal continuation.

7. Why Perceptual Metrics Matter The nine perceptual channels P_t give the model an internal sensor for scene stability: • Exposure, flicker, and contrast define lighting state. •

Motion flow and edge persistence define spatial continuity and action intensity. • Temporal coherence defines how “same” the world remains between t and t+1. • Gaze density defines where attention is centered.

These channels act as a self-check. The model learns not just to match pixels, but to maintain believable continuity of light, motion, focus, and visual complexity across time.

8. Deployment Concept The trained model can be used in three modes: 1. Assisted Post: Relight, reblock, or restage an existing shot by modifying the text control tokens and regenerating forward. 2. World Player: Autonomously roll forward from a prompt to create new, coherent shots in the style of a learned show. 3. Engine Integration: Stream the predicted latent states into a realtime engine (e.g. Unity) to instantiate geometry, lighting, and animation based on the learned temporal physics.

9. Summary This pipeline unifies video, audio, language, and perceptual feedback into a recurrent next-frame predictor. The model is trained to perfect recall on one sequence, then forced to generalize across multiple sequences with different styles. When frozen, it becomes a controllable temporal world model. Generation is no longer frame guessing. It is causally consistent scene continuation under prompt control.