Automated Span-Level Visual Variation: Architectures for Context-Aware Text-to-Video Prompt Engineering

1. Introduction: The Deterministic Imperative in Generative Video

The rapid ascent of high-fidelity Text-to-Video (T2V) generative models—exemplified by OpenAI's Sora, Google's Veo, and RunwayML's Gen-3—has fundamentally altered the landscape of digital content creation. We are witnessing a transition from a regime of *manual rendering*, where visual fidelity was the primary constraint, to a regime of *prompt-mediated synthesis*, where the limiting factor is the user's ability to articulate high-dimensional visual concepts in natural language. Unlike their text-based predecessors (Large Language Models or LLMs), where ambiguity can occasionally yield serendipitous literary results, T2V models operate under a stricter modality that necessitates a paradigm shift from conversational requests to structured, cinematic briefings.[1] The T2V model functions less like a chatbot and more like a virtual production crew, requiring precise, technical, and visually grounded instructions to maintain spatio-temporal coherence and avoid hallucinations.[1]

This report presents a comprehensive architectural and theoretical framework for a "Prompt Builder" tool designed to facilitate high-precision video generation through span-level, context-aware lexical substitution. The proposed system functionality—detecting semantic spans within a user's prompt and generating 12 distinct replacement phrases—represents a non-trivial Natural Language Processing (NLP) challenge. It sits at the intersection of **Controlled Text Generation (CTG)**, **Context-Aware Lexical Substitution (CALS)**, and **Vision-Language Alignment**. The objective is not merely to find linguistic synonyms (e.g., replacing "car" with "automobile") but to generate "visual synonyms" and "visual variants" that meaningfully alter the aesthetic, compositional, or narrative output of the video model while rigorously preserving the logical consistency and grammatical integrity of the prompt.[2]

1.1 The Visual-Semantic Gap in T2V Prompting

A central finding in current research is the existence of a "visual-semantic gap": the discrepancy between what a word signifies in a linguistic context and how a diffusion model interprets it in a visual latent space. Research indicates that optimal video generation necessitates a modular and hierarchical prompt structure, often referred to as the **Universal Prompt Framework**.[1] This framework prioritizes elements in a logical sequence: **Shot Type > Subject > Action > Setting > Camera Behavior > Lighting > Style**. The order is non-arbitrary; elements placed earlier in the prompt receive greater attention weight during the denoising process.[1]

Consequently, a "Prompt Builder" tool cannot treat all text spans equally. A span replacement for a "Subject" must be handled with different constraints than a replacement for "Lighting." Introducing a replacement phrase that accidentally shifts the "Shot Type" (e.g., replacing "looking at" with "close-up view of") can disrupt the global composition of the scene, violating the user's intent. Furthermore, simple lexical substitution is often insufficient. The tool must bridge the visual-semantic gap by enforcing **Visual Grounding**: the translation of abstract concepts into observable physical details. A prompt for a "sad man" is considered weak because "sadness" is an internal state; a strong replacement would be "a man with slumped shoulders and downcast eyes," providing concrete visual signals for the model to render.[1]

1.2 Core Operational Risks and Failure Modes

Developing an automated span replacement system for T2V introduces specific operational risks that differ from standard text editing assistants. Inferred from literature on LLM hallucinations, instruction following, and the specific architectural constraints of video diffusion models, we identify five primary failure modes that the system architecture must mitigate.

Failure Mode Category	Specific Manifestation	T2V Impact & Research Context
Semantic Drift	The replacement phrase alters the core narrative or causal logic of the scene (e.g., changing "walking" to	Violates the "One Clip, One Action" principle, leading to temporal incoherence and hallucinated motion.[1]

	"flying" in a realistic drama).	
Grammatical Fracture	The inserted span does not match the tense, number, or case of the surrounding text (e.g., inserting a noun phrase where a verb phrase is required).	Confuses the T2V text encoder (often T5 or CLIP), resulting in ignored prompts or visual artifacts due to attention misalignment.[2]
Visual Collapse	The 12 generated options are linguistically distinct but visually identical (e.g., "big dog," "large dog," "huge dog").	Wastes user interaction cycles and fails to provide meaningful creative control. High semantic similarity does not equal high visual diversity.[8]
Attribute Bleed	A replacement span introduces conflicting attributes (e.g., introducing "nighttime" into a prompt already defined as "sunny day").	Causes "flickering" or hybrid visual artifacts where the model attempts to render incompatible states simultaneously.[1]
Role Drift	The LLM generating the replacements begins to "chat," explain its choices, or hallucinate conversational filler rather than outputting raw phrases.	Breaks the tool's UI integration and requires robust, often fragile, output parsing logic.[10]

The following sections will dissect these challenges using evidence from **Lexical Substitution**, **LLM-as-a-Judge**, and **Diversity-Aware Sampling** literature, culminating in three actionable prompt architectures and a multi-tiered evaluation plan.

---

2. The Linguistics of Visual Generation: Deconstructing the Prompt

To engineer a system capable of meaningful text replacement for video, one must first understand the unique linguistic register required by state-of-the-art T2V models. Unlike human-to-human communication, where context is inferred, human-to-model communication requires explicit, denotative instructions. The literature suggests that successful T2V prompts are less like prose and more like code—syntactic structures that compile into specific visual outcomes.

2.1 The Universal Prompt Framework and Hierarchy

The consensus among researchers and expert practitioners establishes that the most robust prompts follow a modular, hierarchical structure.[1] This structure is not merely a stylistic preference but a reflection of how transformer-based text encoders process input sequences. Attention mechanisms in models like CLIP (Contrastive Language-Image Pre-training) or T5 (Text-to-Text Transfer Transformer) often bias towards the beginning of the sequence. Therefore, the "Prompt Builder" must respect the following hierarchy when suggesting replacements:

1. **Shot Type / Framing:** This establishes the spatial boundaries of the scene (e.g., "Wide shot," "Extreme Close-Up").
2. **Subject:** The primary anchor of the visual generation (e.g., "A 30-year-old astronaut").
3. **Action:** The temporal dynamic (e.g., "floating weightlessly").
4. **Setting:** The environmental context (e.g., "in a dimly lit space station").
5. **Camera Behavior:** The movement of the observer (e.g., "slow dolly in").
6. **Lighting:** The illumination attributes (e.g., "soft blue rim lighting").
7. **Style:** The aesthetic parameters (e.g., "cinematic, 35mm film grain").

This hierarchy dictates the *risk profile* of a span replacement. Modifying a span related to **Lighting** (Tier 6) is a relatively safe, orthogonal operation that changes the mood without altering the scene's geometry. Modifying a span related to **Subject** (Tier 2) or **Shot Type** (Tier 1) is a destructive operation that effectively resets the random seed's generation path. The tool's logic must be aware of which "slot" in the hierarchy the selected span occupies to generate appropriate alternatives. For instance, if a user selects "Wide shot," the replacements must be other framing terms (e.g., "Bird's eye view"), not lighting terms, to preserve the structural integrity of the prompt.[1]

2.2 The "One Clip, One Action" Principle

A critical constraint identified in the analysis of T2V failure modes is the models' inability to handle complex, sequential narratives within a single generation window. The "One Clip, One Action" principle asserts that a single prompt should describe one continuous temporal event.[1] Prompts containing conjunctions like "and then," "after," or complex temporal dependencies often lead to "morphing" artifacts, where the subject inexplicably transforms or the background shifts unnaturally.

This has profound implications for the "Action" span replacements. If a user highlights a verb phrase like "runs down the street," the system must generate replacements that are also singular, continuous actions (e.g., "sprints aggressively," "jogs leisurely"). It must strictly avoid generating sequential actions (e.g., "runs and then stops") or complex state changes that the diffusion model cannot resolve temporally.[6] This requirement aligns with findings in **Controlled Text Generation (CTG)**, where constraints on output complexity are necessary to ensure valid downstream execution.[12]

2.3 Visual Grounding: The Accessibility Connection

A surprisingly relevant domain for T2V prompt engineering is **accessibility and audio description**.[13] The guidelines for describing video content to visually impaired audiences—focusing on objective, observable details rather than subjective interpretation—mirror exactly what T2V models require.

When a user writes "a scary monster," they are using a high-level semantic concept. However, the model needs low-level visual features. The Prompt Builder should function as a "Visual Grounding Engine," translating abstract concepts into concrete descriptors.

- **Abstract:** "Scary"
- **Grounded Replacement 1:** "looming, shadowed figure with elongated limbs"
- **Grounded Replacement 2:** "jagged, chitinous beast with glowing red eyes"

This process, supported by research into **Visual-Semantic Arithmetic** [15] and **Captioning by Discriminative Prompting** [16], involves decomposing the latent vectors of abstract terms into their constituent visual attributes. By leveraging the vocabulary of audio description (e.g., specifying hair color, clothing texture, exact physical actions), the tool can guide the user toward prompts that yield higher fidelity results.[13]

2.4 The Director's Lexicon: Technical Specificity

To achieve "cinematic" results, the prompt must utilize the lexicon of professional cinematography. The research highlights that T2V models, having been trained on vast datasets of captioned video and film, are highly responsive to technical terminology.[1] The Prompt Builder must be equipped with a knowledge base of these terms to offer as replacements.

- **Camera Movement:** Instead of "the camera moves," replacements should offer "Truck Left," "Pan Right," "Pedestal Up," or "Rack Focus".[1]
- **Lighting:** Instead of "dark," replacements should offer "Low-key lighting," "Chiaroscuro," "Silhouette," or "Volumetric fog".[17]
- **Lens:** Specificity regarding focal lengths (e.g., "85mm" for portraits, "24mm" for landscapes) acts as a powerful style conditioner.[1]

The system must categorize the highlighted span. If the user highlights "moves," the system should recognize this as a *Camera Behavior* or *Subject Action* slot and retrieve the corresponding technical lexicon, ensuring the replacements are domain-appropriate and precise.

---

3. Theoretical Framework: Context-Aware Lexical Substitution

To implement the features described above, we must ground the system in the theoretical mechanisms of **Context-Aware Lexical Substitution (CALS)**. This field of NLP focuses on replacing words or phrases in a sentence with suitable alternatives that fit the context, a task that has evolved significantly with the advent of Transformer models.

3.1 From Masked Language Modeling (MLM) to Infilling

Historically, CALS approaches utilized Masked Language Models (MLMs) like BERT. The target word would be replaced with a `` token, and the model would predict the probability distribution of tokens that could fill that slot.[18] While effective for single-word synonyms, standard MLM approaches have limitations:

1. **Token Count Constraints:** They struggle to replace a single token with a multi-token phrase (e.g., replacing "car" with "vintage red convertible") without complex decoding heuristics.
2. **Diversity vs. Accuracy:** They tend to prioritize the most statistically probable words (synonyms) rather than creative variants, leading to "Visual Collapse."

Modern Large Language Models (LLMs) offer a superior paradigm: **Text Infilling** or **Fill-in-the-Middle (FIM)**.[20] In FIM training, models are taught to generate text given both a *prefix* (context before the span) and a *suffix* (context after the span). This is formally represented as maximizing $P(Span | Prefix, Suffix)$. This architecture is ideal for the Prompt Builder because it naturally handles variable-length replacements and maintains coherence with the entire prompt structure, not just the preceding words. The tool's backend should leverage models fine-tuned on FIM objectives or use specific prompt engineering patterns (like providing the prefix and suffix explicitly) to simulate this behavior.[22]

3.2 Orthogonality and Disentangled Representations

A key theoretical concept for generating *diverse* video prompts is **orthogonality** in attribute generation. Research on **disentangled representation learning** in generative models suggests that visual attributes (color, shape, lighting, texture) should ideally be controlled independently.[3]

In the context of the Prompt Builder, this means that replacement options should be **orthogonal vectors** in the semantic space.

- If the user highlights a *Lighting* span, the 12 replacements should vary the lighting conditions (Golden Hour vs. Blue Hour vs. Neon) *without* inadvertently altering the *Subject* (e.g., changing "man" to "woman") or the *Action*.
- This requires **Attribute-Specific Prompting** [25], where the LLM is explicitly constrained to manipulate only specific semantic dimensions. For example, a prompt might instruct the model: "Generate variations of the lighting description only. Keep the subject and setting constant."

Achieving this orthogonality is crucial for preventing **Attribute Bleed**, where a replacement introduces conflicting information that confuses the video generation model.[3]

3.3 Entropy, Temperature, and Diversity Sampling

Generating 12 options that are genuinely distinct requires careful management of the decoding strategy (sampling). Standard "greedy" decoding or low-temperature sampling will result in 12 variations of "very similar" phrases. To achieve **Visual Variance**, the system needs high-entropy sampling strategies.

Research into **Entropy-based Dynamic Temperature (EDT)** and **Top-H sampling** [27] suggests that dynamic adjustment of temperature based on the model's confidence can balance coherence with creativity.

- **Contrastive Decoding:** A powerful technique found in the literature is **Contrastive Prompting** or Decoding.[29] This involves penalizing the model for generating tokens that are too similar to a reference set (e.g., the original span or previously generated options). By explicitly maximizing the semantic distance between the generated options (while staying within the grammatical constraints), the system can force the LLM to explore "orthogonal" areas of the latent space, moving from "synonyms" to "alternatives."

3.4 Lessons from Watermarking: Semantic Preservation

While the user's goal is creative generation, the *mechanisms* required are surprisingly similar to those used in **text watermarking**.[2] In robust watermarking, a system must substitute words in a text to embed a signal *without* changing the text's meaning or breaking its grammar.

- **Relevance:** The algorithms developed for watermarking—specifically those using BERT to score the "Semantic Relatedness" and "Grammatical Fit" of a substitution—can be repurposed as quality filters for the Prompt Builder.
- **Application:** Before presenting the 12 options to the user, the system can run a "Watermark-style" check: Does this replacement preserve the fundamental *logic* of the prompt (Semantic Integrity) while changing the *visuals*? If a replacement breaks the causal logic of the sentence (a failure mode known as **Semantic Drift**), it should be discarded.[19]

---

4. Designed Prompt Architectures

Synthesizing the T2V constraints and NLP theories discussed, we propose three distinct prompt architectures (patterns) for the "Prompt Builder." Each design targets a specific user

intent and mitigates specific failure modes.

Design 1: The "Orthogonal Attribute Injector" (Technical Variation)

Purpose: To generate precise, technical variations of a specific cinematic element (e.g., camera, lighting, lens) without altering the narrative content. This addresses the need for the "Director's Lexicon" and enforces **Orthogonality**.

Mechanism: This pattern uses a **Role-Conditioned** approach combined with **JSON Schema Enforcement**. The LLM adopts the persona of a Cinematographer/Gaffer. It first classifies the span and then generates variations strictly within that category.

Prompt Pattern:

SYSTEM ROLE

You are an expert Cinematographer and Prompt Engineer for high-end AI video generation (Sora/Veo). Your goal is to provide precise, technical variations for specific elements of a video prompt.

TASK

1. **Analyze** the Context Sentence and the Highlighted Span.
2. **Classify** the Highlighted Span into one of the Universal Prompt Framework categories:.
3. **Generate** 12 distinct replacement phrases that fit the grammatical slot of the Highlighted Span.
4. **CONSTRAINTS:**
    - Replacements must be distinct technical approaches within the identified category (e.g., if Lighting, vary direction/quality/source).
    - Do NOT change the Subject or the Main Action.
    - Use industry-standard terminology (e.g., "Rembrandt lighting," "Rack focus").
    - Output strictly in valid JSON format.

INPUT

Context: "A {{Wide shot}} of a cyberpunk city street at night, neon rain falling."
Highlighted: "{{Wide shot}}"

OUTPUT SCHEMA

{
"category": "Shot Type",
"replacements":
}
Rationale: By explicitly asking for classification first, we force the model to recognize the "slot" the span occupies.[1] The JSON constraint [30] prevents **Role Drift** (the model chatting with the user). The "Cinematographer" persona primes the latent space for technical vocabulary.

Design 2: The "Visual Decomposition" Expander (Creative Grounding)

Purpose: To translate abstract, subjective, or simple descriptors into rich, visually grounded details. This addresses the **Visual Grounding** requirement and prevents **Visual Collapse** (generic synonyms).

Mechanism: This pattern uses **Chain-of-Thought (CoT)** prompting [31] to force the model to "imagine" visual details before writing the phrase. It essentially performs "Visual-Semantic Arithmetic".[15]

Prompt Pattern:

INSTRUCTION

You are a Visual Director for an AI video generation engine. The user has provided a simple descriptor. Your task is to "explode" this descriptor into 12 visually distinct, highly detailed manifestations suitable for a generative video model.

PRINCIPLES

- **Show, Don't Tell:** Convert abstract adjectives (e.g., "scary") into physical descriptions (e.g., "shadows elongating," "paint peeling").
- **Visual Variance:** Ensure the 12 options represent different art styles, moods, or physical configurations. They must not be synonyms; they must look different.
- **Grammatical Continuity:** The output must be a Noun Phrase or Adjectival Phrase that fits the original sentence structure perfectly.

INPUT

Full Prompt: "A robot walking through a [futuristic] garden."
Highlighted Span: "[futuristic]"

STEPS

1. **Analyze:** The span modifies "garden."
2. **Brainstorm (Internal Monologue):** Think of 12 different "futuristic" aesthetics (Cyberpunk, Solarpunk, Bioluminescent, Sterile/Minimalist, Ruined/Post-Apocalyptic, Crystal, Holographic, etc.).
3. **Draft:** Write descriptive replacements based on these aesthetics.

OUTPUT (JSON List Only)

[
"neon-soaked, rain-slicked cyberpunk",
"lush, overgrown solarpunk",
"sterile, white porcelain and chrome",
"bioluminescent, glowing fungal",
"floating, zero-gravity",
"geometric, fractal-based",
"holographic, glitching simulation",

"rusty, steam-powered intricate",
"glass and steel architectural",
"crystalline, semi-translucent",
"dense, vertical hydroponic",
"ethereal, fog-filled nanotech"
]
Rationale: This pattern leverages CoT to break the immediate association between "futuristic" and generic sci-fi tropes. By instructing the model to brainstorm distinct sub-genres (Solarpunk vs. Cyberpunk), it ensures the 12 options are **orthogonal** in visual style.[33] This directly combats visual collapse.

Design 3: The "Grammar-Constrained" Narrative Editor (Story Consistency)

Purpose: To change the narrative action or subject while strictly maintaining the sentence flow, logical coherence, and temporal constraints. This addresses **Grammatical Fracture** and **Semantic Drift**.

Mechanism: This uses **Few-Shot Prompting** [34] with **Negative Constraints**. It explicitly forbids sequential logic ("and then") to uphold the "One Clip, One Action" principle.

Prompt Pattern:

TASK

Generate 12 lexical substitutions for the highlighted span. The replacements must change the *narrative content* (what is happening) but strictly adhere to the *grammatical structure* (tense, number, transitivity) of the surrounding sentence.

CONSTRAINTS

- **One Clip, One Action:** Do not introduce sequences (avoid "and then", "after", "starts to"). Describe a continuous state or action.
- **Tense Consistency:** Match the tense of the original sentence exactly.

- **Logical Fit:** The replacement must make sense within the surrounding context provided.

EXAMPLES

Context: "The astronaut [floats aimlessly] through the void."
-> Replacements: "drifts slowly", "spins uncontrollably", "propels forward", "hangs suspended"
Context: "A cat [sitting on] a wall."
-> Replacements: "sleeping atop", "prowling along", "jumping off", "balancing on"

INPUT

Context: "A medieval knight [raises his sword] in victory."
Highlighted: "[raises his sword]"

GENERATE 12 VARIATIONS

(Output as JSON list)

Rationale: By providing few-shot examples, we align the model with the specific syntactic requirements (e.g., Transitive Verb + Object). The negative constraints explicitly block the complex temporal structures that confuse video models.[1]

---

5. Evaluation Plan: Metrics for Visual and Semantic Integrity

Evaluating the quality of generative prompt suggestions is complex because "quality" is subjective. However, we can decompose quality into measurable dimensions: **Grammaticality**, **Diversity**, and **Visual Alignment**. We propose a three-tiered evaluation strategy that synthesizes metrics from NLP and Computer Vision.

5.1 Tier 1: Automated Structural & Grammatical Integrity

This tier represents the "sanity check" layer. It runs in real-time to filter out broken suggestions before they reach the user interface.

- **Metric 1: Grammatical Perplexity (PPL):** We can use a lightweight language model (like DistilGPT-2 or a small Llama model) to calculate the perplexity of the sentence *after* the replacement is inserted. If the perplexity spikes significantly compared to the original sentence, it indicates a **Grammatical Fracture** (e.g., "The cat *blue* the wall" vs. "The cat *climbed* the wall"). High-PPL options are discarded.[35]
- **Metric 2: Format Compliance Rate:** A simple validation check (Regex or JSON parser) to ensure the output matches the requested schema (e.g., purely a JSON list, no conversational filler). This measures the system's resistance to **Role Drift**.[30]

5.2 Tier 2: Semantic and Visual Diversity

This tier assesses whether the 12 options are actually different from each other, addressing the **Visual Collapse** failure mode.

- **Metric 3: Semantic Diversity (Self-BERTScore / Self-BLEU):** We calculate the pairwise BERTScore or BLEU score between all 12 generated options. A score of 1.0 indicates synonyms (low diversity). We aim for a lower similarity score, indicating high semantic diversity.[36]
- **Metric 4: Visual Diversity via Proxy (Vendi Score on CLIP):** This is a novel application of the **Vendi Score**.[9] The Vendi Score measures the effective number of unique elements in a distribution.
    - *Procedure:* Ideally, we would generate 12 videos and compare them, but that is cost-prohibitive. Instead, we use the **CLIP Text Encoder** as a proxy. We encode the 12 full prompts (Context + Replacement) into CLIP embeddings. We then compute the Vendi Score on these text embeddings.
    - *Rationale:* Since CLIP aligns text and image spaces, high variance in CLIP *text* embeddings serves as a strong proxy for high variance in the resulting *visual* output.[39] This metric mathematically quantifies "Visual Variance" without pixel generation.

5.3 Tier 3: LLM-as-a-Judge (Semantic Alignment)

This tier evaluates the qualitative aspects: Does the prompt sound like a director wrote it? Is it safe?

- **Mechanism:** We employ the **LLM-as-a-Judge** paradigm [41], utilizing a high-capability model (e.g., GPT-4o) to grade the generated list against a rubric.
- **Rubric:**
  1. **Cinematic Quality:** (1-5) Does the phrase use technical lexicon (e.g., "dolly," "volumetric")? [1]
  2. **Visual Grounding:** (1-5) Is the description concrete and observable? [5]
  3. **Safety & Policy:** (Pass/Fail) Does it violate safety guidelines?
  4. **Temporal Coherence:** (Pass/Fail) Does it violate the "One Clip, One Action" rule? [1]
- **Implementation:** The Judge LLM receives the list and outputs a JSON scorecard. This allows for A/B testing different prompt architectures (e.g., comparing Design 1 vs. Design 2) to see which yields higher "Cinematic Quality" scores.

5.4 T2V-Specific Benchmarks

To further validate the system, we can leverage datasets like **T2V-CompBench** and **DEVIL** (Dynamics Evaluation).[44] These benchmarks specifically test "attribute binding" (e.g., red car vs. blue car) and "dynamic consistency."

- **Adaptation:** We can create a "Golden Set" of test cases derived from T2V-CompBench prompts. We feed these into our Prompt Builder and verify if the suggested replacements maintain the rigorous attribute binding required by the benchmark metrics. For example, if the original prompt is "A red cube," and the tool suggests "A blue sphere," we use the benchmark's VQA (Visual Question Answering) methodology to verify if a standard video model actually renders the change correctly.[46]

---

6. Actionable Recommendations for Implementation

To transform this research into a robust production feature, we recommend the following implementation roadmap, focusing on architectural decisions that enforce the constraints discussed.

6.1 Step 1: Implement Grammar-Constrained Decoding (GCD)

Do not rely solely on prompt engineering (instructions) to format the output. Prompting is probabilistic; strict syntax is deterministic.

- **Action:** Implement **Grammar-Constrained Decoding (GCD)** or **JSON Mode** at the inference level.[47] Libraries like outlines, guidance, or llama-cpp-python allow you to supply a formal grammar (e.g., a JSON schema or a Regex) that masks the logits during generation.
- **Benefit:** This mathematically guarantees that the model *cannot* output conversational filler or invalid JSON. It forces the model to output *only* the list of strings, completely eliminating the **Role Drift** failure mode and the need for fragile parsing logic.[50]

6.2 Step 2: Taxonomy-Driven Prompt Routing

Do not use a "one-size-fits-all" prompt for all spans. The visual impact of changing a "Subject" is vastly different from changing "Lighting."

- **Action:** Build a lightweight classifier (using a small BERT model or even a Regex-based heuristic) to detect the *type* of span the user highlighted based on the Universal Prompt Framework categories.
  - If Span = **Verb** $\rightarrow$ Route to **Design 3** (Narrative Editor) to enforce "One Action" constraints.
  - If Span = **Adjective/Noun** $\rightarrow$ Route to **Design 2** (Visual Decomposition) to maximize grounding.
  - If Span = **Technical Term** (e.g., "zoom") $\rightarrow$ Route to **Design 1** (Orthogonal Attribute) to suggest technical variants.
- **Benefit:** This "Router" architecture ensures the LLM instructions are tailored to the specific linguistic and visual function of the selected text, maximizing relevance.

6.3 Step 3: Diversity Sampling with Contrastive Penalty

Standard temperature sampling is often insufficient for generating 12 truly distinct ideas; models tend to cluster around the most probable synonyms.

- **Action:** Implement **Contrastive Prompting** or iterative generation with repetition penalties.
  - Generate the first 4 options with standard sampling (temperature=0.7).
  - Generate the next 4 with a higher temperature (temperature=0.9) and a **Negative Constraint**: "Do not use concepts related to [List of first 4 options]."
  - Repeat for the final 4.
- **Benefit:** This forces the model to explore orthogonal areas of the latent space.[29] It ensures the list offers a wide "Visual Variance" rather than 12 shades of the same concept (e.g., moving from "lighting" variations to "texture" variations to "camera" variations).

6.4 Step 4: The "Visual Rationale" Feedback Loop

Users may not understand technical terms like "Rembrandt Lighting" or "Dutch Angle."

- **Action:** Since generating preview videos is slow and expensive, generate a text-based **"Visual Rationale"** for each option (using a variant of Design 2). Display this as a tooltip in the UI.
  - *Option:* "Rembrandt Lighting"
  - *Tooltip:* "High contrast, moody shadows, focuses attention on the face with a triangle of light."
- **Benefit:** This bridges the knowledge gap for non-expert users [1], making the "Director's Lexicon" accessible and educational.

6.5 Future Outlook: Automated Prompt Optimization (APO)

The system should be designed to learn. The literature points toward **Reinforcement Learning from Human Feedback (RLHF)** as a key driver for alignment.[51]

- **Action:** Log user interactions. Which of the 12 suggestions do users actually click? Do they edit the suggestion after clicking?
- **Future State:** Use this click-stream data to fine-tune a **LoRA (Low-Rank Adaptation)** adapter specifically for "high-acceptance video prompt suggestions".[52] This moves the system from a static generator to an adaptive assistant that learns the specific aesthetic preferences of its user base.

Conclusion

The development of a "Prompt Builder" for T2V is not merely a text processing task; it is an exercise in **Visual-Semantic Engineering**. By acknowledging the deterministic nature of video diffusion models and strictly adhering to the **Universal Prompt Framework**, the tool can guide users away from the ambiguity that plagues current generators. The integration of **orthogonal attribute injection**, **grammar-constrained decoding**, and **CLIP-based diversity metrics** provides a robust technical foundation for solving the "Visual Collapse" and "Semantic Drift" problems. This system ultimately empowers users to act as directors, translating their intent into the precise, structural language that modern AI video models require to perform.

Citations:
1 - T2V Prompting Frameworks & Lexicon
2 - Context-Aware Lexical Substitution & Watermarking
20 - Infilling & Fill-in-the-Middle
8 - Diversity, Entropy, & Contrastive Sampling
6 - Constraints, GCD, JSON, Role Drift
36 - Evaluation, LLM-as-Judge, Benchmarks
13 - Visual Grounding & Accessibility
3 - Future Trends, Orthogonality, RLHF

Works cited

1. LLM Video Prompt Research and Template copy.pdf
2. arXiv:2504.06575v2 [cs.CR] 10 Apr 2025, accessed November 22, 2025, https://arxiv.org/pdf/2504.06575?
3. OmniPrism: Learning Disentangled Visual Concept for Image Generation - arXiv, accessed November 22, 2025, https://arxiv.org/html/2412.12242v1
4. Ultimate prompting guide for Veo 3.1 | Google Cloud Blog, accessed November 22, 2025, https://cloud.google.com/blog/products/ai-machine-learning/ultimate-prompting-guide-for-veo-3-1
5. How to write effective text prompts to generate AI videos? - FlexClip Help Center, accessed November 22, 2025, https://help.flexclip.com/en/articles/10326783-how-to-write-effective-text-prompts-to-generate-ai-videos
6. Guiding LLMs The Right Way: Fast, Non-Invasive Constrained Generation - arXiv, accessed November 22, 2025, https://arxiv.org/html/2403.06988v1
7. Defending LLM Watermarking Against Spoofing Attacks with Contrastive Representation Learning - ResearchGate, accessed November 22, 2025, https://www.researchgate.net/publication/390638664_Defending_LLM_Watermar

king_Against_Spoofing_Attacks_with_Contrastive_Representation_Learning
8. Zero-shot World Models via Search in Memory - arXiv, accessed November 22, 2025, https://arxiv.org/html/2510.16123v1
9. Scendi Score: Prompt-Aware Diversity Evaluation via Schur Complement of CLIP Embeddings - CVF Open Access, accessed November 22, 2025, https://openaccess.thecvf.com/content/ICCV2025/papers/Ospanov_Scendi_Score_Prompt-Aware_Diversity_Evaluation_via_Schur_Complement_of_CLIP_ICCV_2025_paper.pdf
10. The Hidden Risk of Role Drift in Permissioned Contracts and Multisigs - Olympix, accessed November 22, 2025, https://olympixai.medium.com/the-hidden-risk-of-role-drift-in-permissioned-contracts-and-multisigs-20f809b9ab77
11. Echoing: Identity Failures when LLM Agents Talk to Each Other - arXiv, accessed November 22, 2025, https://arxiv.org/html/2511.09710v1
12. Controlled Text Generation for Large Language Model with Dynamic Attribute Graphs - arXiv, accessed November 22, 2025, https://arxiv.org/html/2402.11218v2
13. Description of Visual Information | Web Accessibility Initiative (WAI) - W3C, accessed November 22, 2025, https://www.w3.org/WAI/media/av/description/
14. How To Create Helpful Visual Descriptions For Visually Impaired Audiences - Veroniiiica, accessed November 22, 2025, https://veroniiiica.com/how-to-create-visual-descriptions/
15. Understanding Visual Concepts Across Models, accessed November 22, 2025, https://visual-words.github.io/
16. It's Just Another Day: Unique Video Captioning by Discriminitive Prompting - arXiv, accessed November 22, 2025, https://arxiv.org/html/2410.11702v1
17. LLM Evaluation Metrics: A Complete Guide - F22 Labs, accessed November 22, 2025, https://www.f22labs.com/blogs/llm-evaluation-metrics-a-complete-guide/
18. Tracing Text Provenance via Context-Aware Lexical Substitution - Jie Zhang, accessed November 22, 2025, https://zjzac.github.io/publications/pdf/aaai22.pdf
19. [2112.07873] Tracing Text Provenance via Context-Aware Lexical Substitution - arXiv, accessed November 22, 2025, https://arxiv.org/abs/2112.07873
20. Natural Language Outlines for Code: Literate Programming in the LLM Era - arXiv, accessed November 22, 2025, https://arxiv.org/html/2408.04820v4
21. Enabling Language Models to Fill in the Blanks - ResearchGate, accessed November 22, 2025, https://www.researchgate.net/publication/343298917_Enabling_Language_Models_to_Fill_in_the_Blanks
22. Efficient Training of Language Models to Fill in the Middle - arXiv, accessed November 22, 2025, https://arxiv.org/pdf/2207.14255
23. Orthogonal Adaptation for Modular Customization of Diffusion Models - Ryan Po, accessed November 22, 2025, https://ryanpo.com/ortha/static/pdfs/ortha.pdf
24. Advancing Textual Prompt Learning with Anchored Attributes - CVF Open Access, accessed November 22, 2025, https://openaccess.thecvf.com/content/ICCV2025/papers/Li_Advancing_Textual_Prompt_Learning_with_Anchored_Attributes_ICCV_2025_paper.pdf

25. Tailor: A Soft-Prompt-Based Approach to Attribute-Based Controlled Text Generation - ACL Anthology, accessed November 22, 2025, https://aclanthology.org/2023.acl-long.25.pdf
26. Tailor: A Prompt-Based Approach to Attribute-Based Controlled Text Generation, accessed November 22, 2025, https://www.researchgate.net/publication/360254458_Tailor_A_Prompt-Based_Approach_to_Attribute-Based_Controlled_Text_Generation
27. [2509.02510] Top-H Decoding: Adapting the Creativity and Coherence with Bounded Entropy in Text Generation - arXiv, accessed November 22, 2025, https://arxiv.org/abs/2509.02510
28. Control the Temperature: Selective Sampling for Diverse and High-Quality LLM Outputs, accessed November 22, 2025, https://arxiv.org/html/2510.01218v1
29. Diverse Text-to-Image Generation via Contrastive Noise Optimization - arXiv, accessed November 22, 2025, https://arxiv.org/html/2510.03813v1
30. JSON prompting for LLMs - IBM Developer, accessed November 22, 2025, https://developer.ibm.com/articles/json-prompting-llms
31. Prompt Engineering Techniques | IBM, accessed November 22, 2025, https://www.ibm.com/think/topics/prompt-engineering-techniques
32. Simulation to Rules: A Dual-VLM Framework for Formal Visual Planning - arXiv, accessed November 22, 2025, https://arxiv.org/html/2510.03182v1
33. PromptMap: Supporting Exploratory Text-to-Image Generation - arXiv, accessed November 22, 2025, https://arxiv.org/html/2510.02814v1
34. Prompting Techniques | Prompt Engineering Guide, accessed November 22, 2025, https://www.promptingguide.ai/techniques
35. LLM Evaluation: Metrics, Methodologies, Best Practices - DataCamp, accessed November 22, 2025, https://www.datacamp.com/blog/llm-evaluation
36. [PDF] Semantic Diversity in Dialogue with Natural Language Inference, accessed November 22, 2025, https://www.semanticscholar.org/paper/32f87b51e3ba42894821716b8145bde41fc65983
37. Measuring and Improving Semantic Diversity of Dialogue Generation - ACL Anthology, accessed November 22, 2025, https://aclanthology.org/2022.findings-emnlp.66.pdf
38. Scendi Score: Prompt-Aware Diversity Evaluation via Schur Complement of CLIP Embeddings - arXiv, accessed November 22, 2025, https://arxiv.org/html/2412.18645v3
39. Diving Into CLIP by Creating Semantic Image Search Engines | by Ahmad Anis | Red Buffer, accessed November 22, 2025, https://medium.com/red-buffer/diving-into-clip-by-creating-semantic-image-search-engines-834c8149de56
40. CLIP: Connecting text and images - OpenAI, accessed November 22, 2025, https://openai.com/index/clip/
41. LLM-as-a-judge: a complete guide to using LLMs for evaluations - Evidently AI, accessed November 22, 2025, https://www.evidentlyai.com/llm-guide/llm-as-a-judge

42. LLM-as-a-judge: can AI systems evaluate human responses and model outputs? - Toloka AI, accessed November 22, 2025, https://toloka.ai/blog/llm-as-a-judge-can-ai-systems-evaluate-model-outputs/
43. LLM-as-a-Judge Simply Explained: The Complete Guide to Run LLM Evals at Scale, accessed November 22, 2025, https://www.confident-ai.com/blog/why-llm-as-a-judge-is-the-best-llm-evaluation-method
44. T2V-CompBench: Video Synthesis Benchmark - Emergent Mind, accessed November 22, 2025, https://www.emergentmind.com/topics/t2v-compbench
45. DEVIL: Evaluation of Text-to-Video Generation Models: A Dynamics Perspective, accessed November 22, 2025, https://t2veval.github.io/DEVIL/
46. [2503.16867] ETVA: Evaluation of Text-to-Video Alignment via Fine-grained Question Generation and Answering - arXiv, accessed November 22, 2025, https://arxiv.org/abs/2503.16867
47. Grammar-Constrained Decoding Makes Large Language Models Better Logical Parsers - ACL Anthology, accessed November 22, 2025, https://aclanthology.org/2025.acl-industry.34/
48. Structured Output Generation in LLMs: JSON Schema and Grammar-Based Decoding | by Emre Karatas | Medium, accessed November 22, 2025, https://medium.com/@emrekaratas-ai/structured-output-generation-in-llms-json-schema-and-grammar-based-decoding-6a5c58b698a6
49. Flexible and Efficient Grammar-Constrained Decoding - arXiv, accessed November 22, 2025, https://arxiv.org/html/2502.05111v1
50. Flexible and Efficient Grammar-Constrained Decoding - OpenReview, accessed November 22, 2025, https://openreview.net/pdf?id=L6CYAzpO1k
51. A Multimodal LLM Pipeline for Video Understanding | by Mohamed Hasan | Medium, accessed November 22, 2025, https://eng-mhasan.medium.com/a-multimodal-llm-pipeline-for-video-understanding-b1738304f96d
52. Impact of LLM-based Review Comment Generation in Practice: A Mixed Open-/Closed-source User Study - arXiv, accessed November 22, 2025, https://arxiv.org/html/2411.07091v1
53. Making Your Videos Accessible: A Step-by-Step Guide - ADA Site Compliance, accessed November 22, 2025, https://adasitecompliance.com/videos-accessible-step-by-step-guide/
54. On text simplification metrics and general-purpose LLMs for accessible health information, and a potential architectural advantage of the instruction-tuned LLM class. - arXiv, accessed November 22, 2025, https://arxiv.org/html/2511.05080v1
55. Measuring Data Diversity for Instruction Tuning: A Systematic Analysis and A Reliable Metric - ACL Anthology, accessed November 22, 2025, https://aclanthology.org/2025.acl-long.908.pdf
56. Visual Diversity and Region-aware Prompt Learning for Zero-shot HOI Detection - arXiv, accessed November 22, 2025, https://arxiv.org/abs/2510.25094
57. [CVPR 2024] EvalCrafter: Benchmarking and Evaluating Large Video Generation Models - GitHub, accessed November 22, 2025,

https://github.com/evalcrafter/EvalCrafter
58. Guidelines for Creating Image Descriptions - American Anthropological Association, accessed November 22, 2025, https://americananthro.org/accessibility/image-descriptions/
59. Unlocking Text Capabilities in Vision Models - arXiv, accessed November 22, 2025, https://arxiv.org/html/2503.10981v2
60. Vision Language Model Prompt Engineering Guide for Image and Video Understanding, accessed November 22, 2025, https://developer.nvidia.com/blog/vision-language-model-prompt-engineering-guide-for-image-and-video-understanding/