PSL-C: A Domain-Specific Cinematic Language for Structured Text-to-Video Prompting

Zach Bogart
PromptBuilder Research Lab
Clyde, NY

Abstract

PSL-C (Prompt Structure Language — Cinematic) is a domain-specific language (DSL) designed to formalize and optimize text-to-video prompting across modern generative video models. By decomposing cinematic intention into structured, tokenized categories, PSL-C improves clarity, reproducibility, and cross-model portability compared to freeform natural language prompts. This paper introduces the design goals, architecture, grammar, and implementation of PSL-C, and argues that it constitutes a novel, protectable intellectual contribution to the emerging ecosystem of prompt engineering tools.

1. Introduction

Text-to-video systems such as Sora, Runway Gen-2, Kling, Pika, and others allow users to create complex, cinematic sequences from natural language descriptions. However, as these systems grow more capable, they also become more sensitive to the structure, ordering, and clarity of prompts. Freeform prose prompts frequently produce inconsistent results, are difficult to reproduce across sessions or models, and make it challenging for users to iteratively refine intent.

PSL-C addresses this gap by introducing a structured cinematic prompting language which maps directly onto intuitive filmmaking concepts: visual style, color and grading, location, vibe, motion, character behavior, camera techniques, lighting behavior, physics, environment effects, audio context, timing, and optional dialogue. Rather than replacing natural language, PSL-C provides a scaffold: a compact, token-centric layer that can be expanded into fluent prose for any supported language while preserving a consistent underlying structure.

2. Background and Related Work

Prior work relevant to PSL-C spans several domains: screenplay formatting, storyboarding languages, animation timing charts, VFX shot planning, and informal prompt engineering conventions developed around image models such as Midjourney and Stable Diffusion. These systems offer valuable patterns but do not constitute a unified, formally defined DSL for cinematic prompting.

PSL-C differs in that it explicitly defines a token grammar, a taxonomy of cinematic dimensions, and a rendering pipeline linking structure to natural-language expansions. It is designed from the outset to be model-agnostic, allowing the same PSL-C scene to target multiple video engines with minimal adaptation.

3. Design Goals and Motivation

- Reproducibility: the same PSL-C scene should yield similar behavior across runs and models.
- Compactness: tokens should compress complex cinematic ideas into short, reusable identifiers.
- Interpretability: both humans and downstream tools should be able to read and reason about PSL-C scenes.
- Extensibility: new tokens and sections should be easy to add without breaking existing content.
- Multilingual reach: token semantics should render cleanly into multiple human languages.
- Assistive potential: the structure should support users with language-production or executive-function challenges.

These goals position PSL-C not only as a power-user tool for filmmakers and creators, but also as a foundation for accessibility and AAC (augmentative and alternative communication) applications.

4. Language Architecture

A PSL-C scene is composed of ordered sections. Each section corresponds to a cinematic dimension and may contain one or more tokens. The high-level section set currently includes:

- STYLE / VISUAL LOOK
- COLOR / GRADING
- LOCATION / SETTING
- VIBE / EMOTIONAL TONE
- MOTION / ACTION VERBS
- CHARACTER BEHAVIOR & EXPRESSION
- ANIMALS & CREATURES (OPTIONAL)
- CAMERA BEHAVIOR / ANGLES / TRICKS
- LIGHTING BEHAVIOR
- PHYSICS & WEIGHT BEHAVIOR
- ENVIRONMENT INTERACTION
- AUDIO / TONAL CONTEXT
- TIMING INSTRUCTIONS
- OPTIONAL DIALOGUE BLOCK

Each of these sections is backed by a curated vocabulary of PSL-C tokens. For example, the LIGHTING section may contain tokens such as " soft-backlight-halo " or " golden-hour-deep-shadows ", while the CAMERA section may include " slow-steadicam-follow " or " extreme-closeup-on-eyes ".

5. Formal Grammar (EBNF-Style)

The top-level structure of a PSL-C scene can be captured using an Extended Backus–Naur Form (EBNF) sketch:

<scene> ::= <style-block> <color-block> <location-block> <vibe-block> <motion-block> <character-block> [<creature-block>] <camera-block> <lighting-block> <physics-block> <environment-block> <audio-block> <timing-block> [<dialogue-block>]

Each block in turn is defined as one or more tokens drawn from a section-specific vocabulary:

<style-block> ::= " STYLE: " <style-token> { "," <style-token> }
<color-block> ::= " COLOR: " <color-token> { "," <color-token> }
<camera-block> ::= " CAMERA: " <camera-token> { "," <camera-token> }

6. Token Taxonomy and Examples

PSL-C tokens are short, hyphenated identifiers that encode a specific cinematic behavior or stylistic idea. Tokens are grouped by section, and each token has a corresponding semantic expansion used when rendering a natural-language description.

Example tokens:

- soft-backlight-halo → " soft backlight creating a faint halo outline around subjects "
- surges-in-bursts → " surges ahead in sudden bursts of movement, then settles briefly "
- micro-stabilized-handheld → " handheld camera with micro-stabilization, retaining slight organic sway "
- hesitates-then-commits → " character hesitates mid-motion before fully committing to the action "
- delays-stack-gradually → " delays accumulate over time, stretching the scene ' s rhythm in subtle ways "

Token design emphasizes readability, internal consistency, and modular composition. A single scene may use only a handful of tokens to express detailed cinematic behavior that would otherwise require several lines of prose.

7. Semantic Mapping and Rendering

PSL-C scenes are not meant to be passed directly to a video model as-is. Instead, they serve as an intermediate representation that is expanded into natural-language prompts. The rendering pipeline performs the following steps:

- Parse the PSL-C scene into sections and tokens.
- Lookup each token ' s semantic expansion in a token dictionary.
- Assemble expansions into coherent, section-ordered prose paragraphs.
- Optionally adapt phrasing to the style of a target model (e.g., Sora vs. Runway).
- Insert model-specific hints or instructions if required.

This pipeline allows the same PSL-C scene to generate multiple textual prompts, such as a verbose description for storyboarding and a compact, model-optimized version for inference.

8. Cross-Model Behavior and Portability

Different text-to-video engines interpret natural language in slightly different ways. By anchoring prompts in PSL-C, a user can keep the underlying cinematic intent constant while adjusting only the rendering step to suit each engine ' s preferred phrasing or constraints.

For example, a PSL-C scene describing handheld micro-stabilized motion, soft backlight, and hesitant character behavior may be rendered with more explicit camera language for one model, and more general phrasing for another, without altering the original scene definition.

9. Multilingual Extensions

PSL-C is designed to be language-independent at the token level. Each token serves as a canonical key that can map to different natural-language expansions for different languages. A token dictionary may therefore contain entries such as:


```
{
    "soft-backlight-halo": {
        "en": "soft backlight creating a faint halo outline around subjects",
        "es": "contraluz suave que crea un halo tenue alrededor de los sujetos",
        "fr": "rétroéclairage doux créant un léger halo autour des sujets",
        "ja": "柔らかな逆光が被写体の周囲に淡いハローを作り出す"
    }
}
```


At render time, the user selects a target language, and the renderer swaps in the appropriate expansions. This allows PSL-C to support Spanish, Japanese, French, and other languages without changing the underlying DSL.

10. PSL-C in Assistive and AAC Contexts

Because PSL-C decomposes the complex task of “ describe a video scene ” into clear, labeled categories, it can function as a cognitive scaffold for users who struggle with spontaneous language production, working memory, or executive function. Rather than requiring a user to invent a full paragraph of descriptive prose, PSL-C lets them select from menus of behaviors, camera motions, and environmental effects.

This structured interaction may support non-verbal or minimally verbal users in expressing internal imagery or narrative intent. Coupled with generative video, PSL-C can potentially act as a bridge from thought to shared audiovisual output, which can then be discussed with caregivers, therapists, or collaborators.

11. Safety, Constraints, and Robustness

By constraining prompts to a curated token set and section structure, PSL-C reduces the likelihood of unexpected or unsafe prompt content. Tokens can be filtered, disabled, or mapped to safe expansions as needed, and malicious or disallowed behavior does not naturally emerge from the controlled vocabulary.

Additionally, PSL-C facilitates auditing and logging of user intent: since each scene is a structured collection of tokens, it is easier to inspect what was requested compared to scanning large blocks of unstructured text.

12. Intellectual Property Considerations

PSL-C, as described in this paper, is an original domain-specific language authored by Zach Bogart. Its section layout, token taxonomy, grammar, and rendering pipeline constitute a creative and technical work subject to copyright protection. Because it is a specific implementation of a cinematic prompting system rather than a broad abstract idea, PSL-C may also form the basis for a provisional patent filing if the author chooses to pursue formal patent protection.

The existence of this written specification, including the defined grammar, examples, and design rationale, establishes prior art and a clear authorship record for PSL-C as a structured invention. Future systems that closely replicate the specific structure, tokenization pattern, or rendering pipeline described here may be evaluated against this document for potential infringement.

13. Future Work

Planned extensions to PSL-C include: finer-grained temporal sequencing (multi-shot timelines), character-level state and memory, integration with story graph tools, and higher-order macros that bundle recurring cinematic motifs into single tokens. Additional multilingual dictionaries and community-driven token packs may also be developed.

14. Conclusion

PSL-C introduces a structured, interpretable, and extensible language for cinematic prompting in text-to-video systems. By treating prompts as scenes composed of tokens rather than unstructured paragraphs, PSL-C improves reproducibility, enables multilingual adaptation, supports assistive communication use cases, and provides a clear basis for intellectual property claims. As generative video technologies evolve, structured languages like PSL-C may become an important layer between human intention and model behavior.