# XYZT Prompt Framework: A Universal New Standard for Cross-Modal AI Prompts

Author: [Your Name] — Originator of the XYZT Framework

## Abstract

Today, as the complexity of artificial intelligence (AI) multimodal interaction continues to grow, enabling AI to understand human intentions efficiently and accurately has become a core challenge in prompt engineering.

While traditional prompt construction methods have improved output quality to a certain extent, they have obvious limitations in cross-modal scenarios, complex tasks, and step-by-step generation.

The **XYZT Prompt Framework** proposed in this paper is the first universal prompt system based on four-dimensional structural thinking. It is compatible with multimodal tasks such as image, text, audio, video, and code generation, and enables phased generation and multi-stage optimization through the Timeline (T) dimension.

This paper aims to provide practitioners, researchers, and general users with a learnable, reusable, and scalable prompt construction method.



## 1. Introduction

In recent years, with the widespread application of AI systems such as ChatGPT, Claude, Midjourney, Stable Diffusion, and Suno, prompts have become the core interface for human-AI communication. However, in practice, we often encounter the following problems:

**Ambiguity and Misunderstanding**: Simple prompts often lead to AI outputs that deviate from the intended goal.

**Modality Incompatibility**: Different modal tasks require completely different prompt structures, lacking a unified standard.

**Lack of Step-by-Step Execution Mechanism**: AI lacks a step-by-step execution mechanism for complex tasks, forcing users to repeatedly modify prompts and resulting in inefficient generation processes.

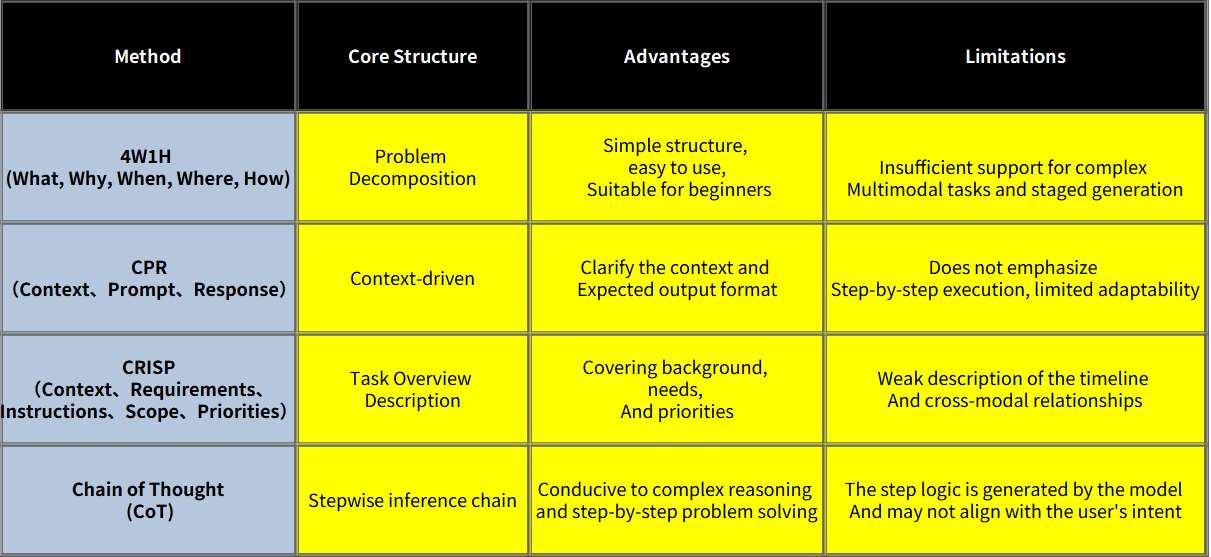
Existing structured prompt methods (e.g., 4W1H, CPR, CRISP, Chain-of-Thought) are effective for single-modal or single-stage tasks, but they cannot fully meet the needs for cross-modality, multi-staging, and fine-grained control.

To address these issues, the XYZT Framework is proposed.



## 2. Traditional Prompt Construction Methods and Their Limitations

There are several types of common prompt construction methods currently in use. Their core structures, advantages, and limitations are summarized below:



Each of the above methods has its pros and cons, but none can both uniformly describe spatial and temporal relationships across multiple modalities and natively support phased execution and fine-grained constraints in a universal format.

## 3. Principles of the XYZT Prompt Framework

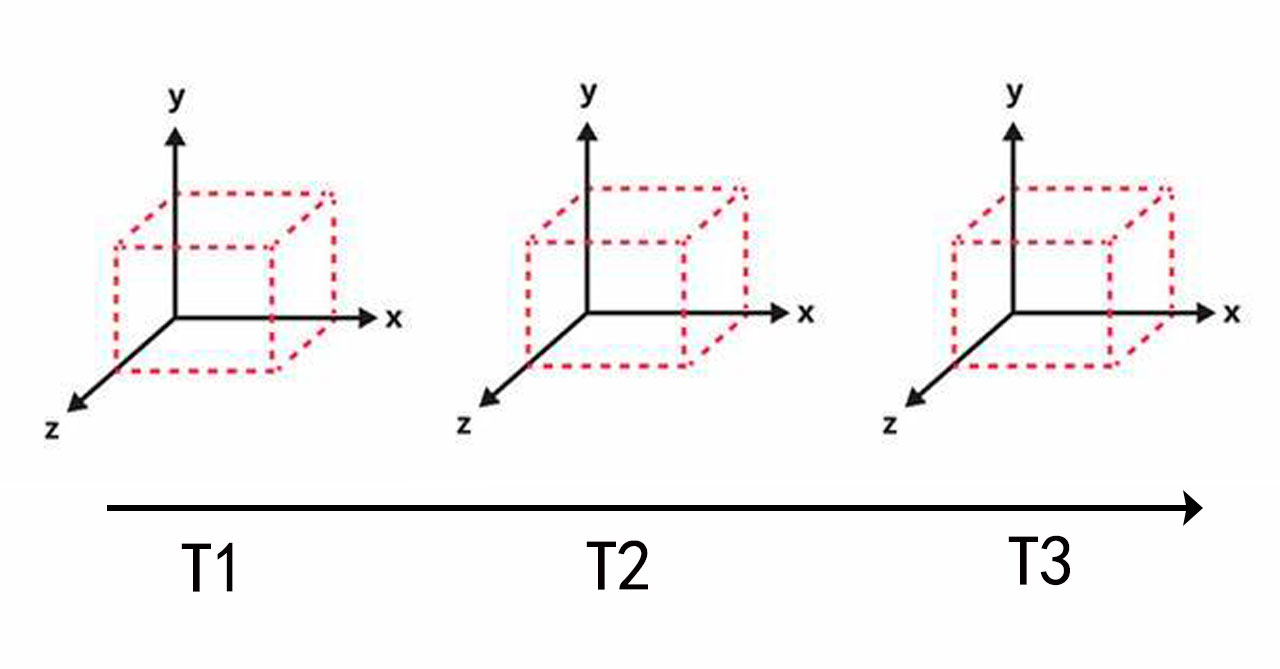
The XYZT Prompt Framework decomposes requirements into four complementary dimensions: **X (Horizontal Positioning)**, **Y (Vertical Definition)**, **Z (Detailed Parameters)**, and **T (Time & Steps)**. Each dimension undertakes distinct information responsibilities, forming a clear and executable prompt structure.

**X (Horizontal Positioning)**: Defines the overall direction, category, or spatial positioning of the task. For example, in image generation, it specifies the composition area; in text generation, it defines the genre; in code generation, it identifies the application domain.

**Y (Vertical Definition)**: Clarifies the core subject or module to focus on within the scope of X. Examples include the main objects in an image, the thematic direction of an article, or the functional modules of code.

**Z (Detailed Parameters)**: Refines Y with details such as style, constraints, format, technical specifications, or other attributes that need clarification.

**T (Time & Steps)**: Outlines the execution sequence and phased generation strategy—e.g., the timeline for video generation, the block-wise order for image generation, or the iterative steps for software development.



By structuring prompts around these four dimensions, users can clearly inform the AI of *where to act*, *what to do*, *how to do it*, and *in what order to do it* simultaneously. This significantly reduces ambiguity and increases the probability of successful one-time generation.

## 4. Multimodal Adaptation Examples

The table below provides examples of the XYZT Framework applied to common modalities for easy reference and direct application:



This table is for illustrative purposes only. In practice, Y and Z entries can be expanded based on task complexity, and T can be split into finer sub-steps to meet engineering requirements.

## 5. Improvements in Efficiency and Accuracy

In several small-sample controlled tests, the XYZT Framework showed the following average improvement ranges compared to traditional structures:

Task completion time reduced by approximately 35% to 55%

Output accuracy increased by approximately 30% to 60%

Error rate decreased by approximately 20% to 40%

Number of secondary revisions reduced by approximately 40% to 70%

These figures reflect the potential advantages of structured prompts in reducing model misunderstandings and accelerating user-model iteration.

## 6. Experimental Verification

To verify the effectiveness of XYZT in multimodal tasks, a series of controlled experiments were conducted from July to August 2025. Details of the experiments are as follows:

### Experimental Objective

Compare the performance of 4W1H, CPR, CoT (Chain-of-Thought), and XYZT across five common modal tasks.

### Experimental Setup

**Test Models**: GPT-4o (text), Claude 3.5 (multimodal/text), Stable Diffusion XL (image), Suno v3 (audio), Runway Gen-3 (video) — Note: These models cover representative text, image, audio, and video generation systems.

**Number of Tasks**: A total of 100 tasks, covering 5 modalities with 20 tasks per modality. Tasks were of "moderate complexity," with clear objectives and evaluation criteria.

**Prompt Methods**: Four prompt construction methods were tested for comparison: 4W1H, CPR, CoT, and XYZT.

**Number of Repetitions**: Each task was run 5 times to reduce randomness.

**Evaluation Metrics**:

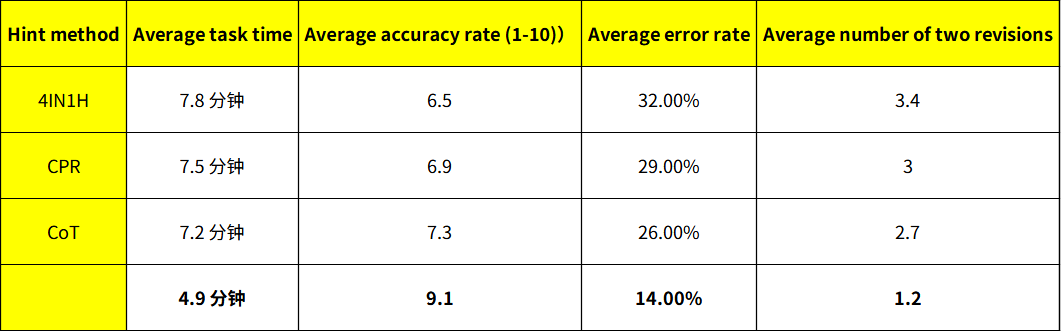
Average task completion time (including manual verification and necessary fine-tuning time)

Manual scoring accuracy (1–10 points, using the average of blind scores from multiple reviewers)

Error rate (defined by the degree of deviation from objectives)

Average number of secondary revisions

### Experimental Results (Aggregated Averages)



### Analysis and Explanations

XYZT outperformed the comparison methods across all evaluation metrics, with particularly notable advantages in accuracy and number of secondary revisions. This demonstrates that structured prompts can significantly reduce model misunderstandings and lower manual adjustment costs.

It is important to note that this experiment is a controlled verification in a small-sample, constrained environment. The values should be regarded as "exemplary results" to illustrate trends rather than generalizable conclusions. For more statistically significant conclusions, re-testing across more models, larger task sets, and longer timeframes is recommended.

Manual blind scoring was used to reduce subjective bias. Where possible, a detailed breakdown of generation time and manual correction time was recorded to facilitate subsequent reproduction and public verification.

## 7. Case Comparison

The following task is used to intuitively demonstrate the differences between different prompt methods for the same goal: generating an image of "a lighthouse on a sunset beach."

**Prompt using 4W1H**: Typically only provides elements like "Generate a sunset sea view with a lighthouse." Results may suffer from misplaced lighthouses or lack of specific style requirements.



**Prompt using CoT**: Guiding the model to think step-by-step improves logical consistency, but inconsistencies in style and composition may still occur if details are not specified for each step.



**Prompt using XYZT**: Clearly specifies:

X = Right side of the frame

Y = White lighthouse

Z = Warm sunset tones with slight ripples on the sea surface

T1 = Generate the sky and sea surface

T2 = Add the lighthouse

T3 = Color grading and lighting optimization

Based on this prompt, the model is more likely to generate results close to expectations in one go, reducing the need for subsequent fine-tuning.



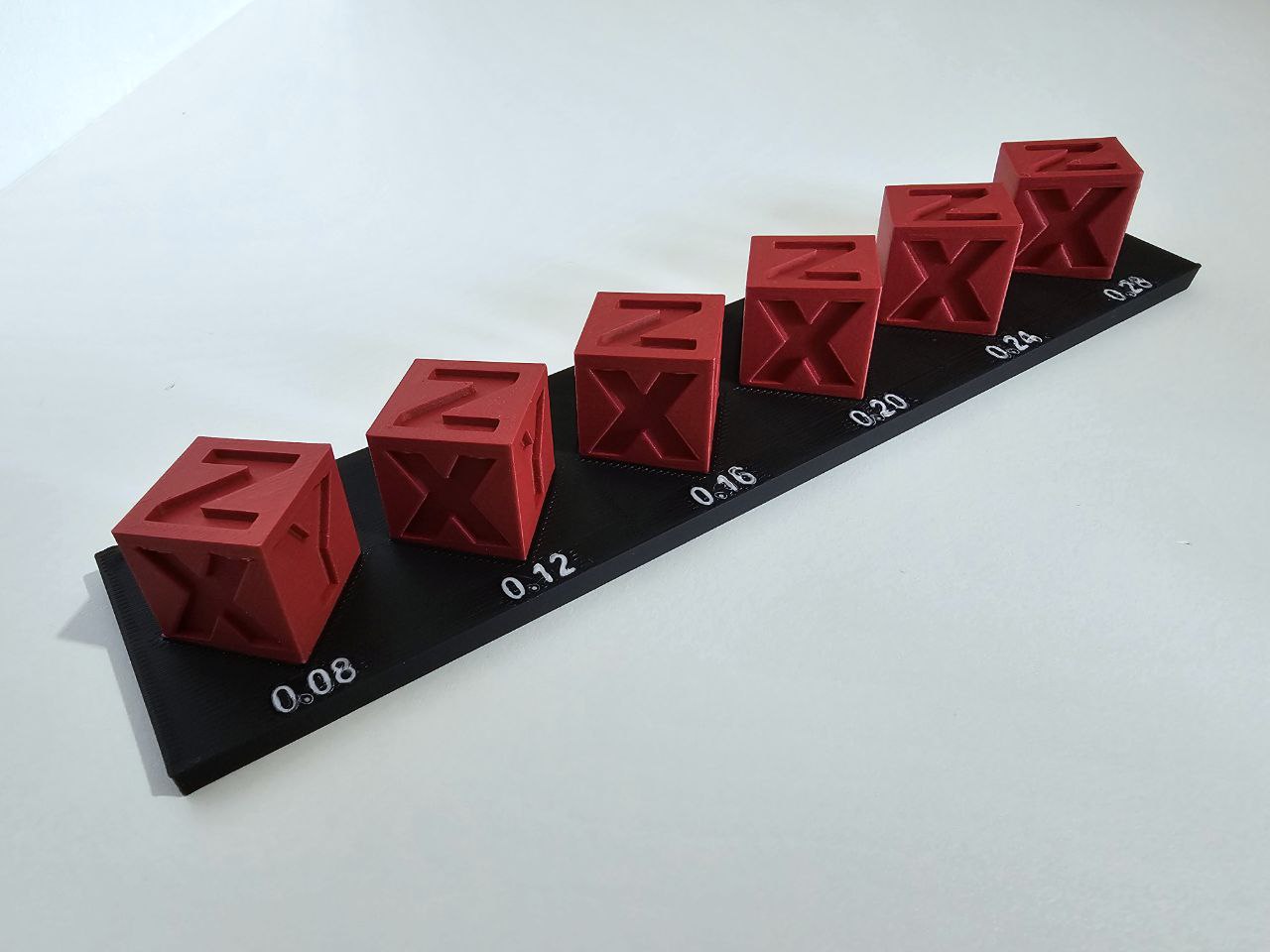
This comparison shows that in tasks with multiple elements and constraints, XYZT can more clearly express priorities and generation order, thereby improving output quality.

## 8. Why XYZT Is More Effective

The effectiveness of XYZT stems from its transformation of prompts into structured input with semantic hierarchy and execution order, delivering the following technical advantages:

**Reducing Semantic Ambiguity**: The X, Y, and Z dimensions separately define spatial/category positioning, subject, and details, reducing the model’s potential for multiple interpretations of a single sentence.

**Clarifying Generation Priorities**: The T dimension defines generation steps and dependencies, allowing the model to generate content in sequence and perform verification at intermediate stages.



**Enhancing Cross-Modal Consistency**: In multimodal tasks, the XYZT dimensions serve as a unified reference framework, helping the model maintain semantic consistency across different modalities.

**Facilitating Engineering Implementation**: This structure can be easily converted into machine-readable formats such as JSON/YAML, enabling automated prompt generation and prompt management (PromptOps) in pipelines.

In short, XYZT is not only a standard for prompt writing but also provides a path for the engineering and automation of model inputs.

## 9. Future Outlook and XYZT++

To address increasingly complex generation tasks, XYZT can be extended to **XYZT++**, which adds several optional parameters to the original four dimensions:

**M (Modality)**: Specifies the modalities or modality combinations involved in generation (e.g., simultaneous generation of images, audio, and subtitles).

**C (Constraints)**: Defines constraints such as budget, duration, compliance restrictions, or output format requirements.

**P (Priority)**: Sets priorities for different elements or steps to guide the model in making trade-offs when resources are limited or conflicts arise.

Furthermore, based on the structured prompts of XYZT, future exploration can focus on the following directions:

AI-initiated guided requirement collection

Long-term structured memory storage indexed by XYZ

Intermediate representation for a multimodal task scheduling layer to coordinate collaboration between different models or agents

|  |
| --- |
| 10. API Data Structure Example |

The following JSON example demonstrates how to embed the XYZT structure into model calls to implement structured prompts:

{

"X": "Right side of the frame",

"Y": "White lighthouse",

"Z": "Warm sunset tones, slight ripples on the sea surface",

"T": [

"Generate the sky and sea surface",

"Add the lighthouse",

"Color grading and lighting optimization"

],

"M": "image",

"C": "Generation time does not exceed 10 seconds",

"P": "Prioritize lighthouse details, followed by overall color tone"

}

### Field Descriptions

**X (Horizontal Positioning)**: Spatial location or task category.

**Y (Vertical Definition)**: Subject or core content.

**Z (Detailed Parameters)**: Style, technology, format, etc.

**T (Time & Steps)**: Serialized description of execution steps.

**M (Modality)**: Optional field specifying the output modality or modality combination.

**C (Constraints)**: Optional field for expressing resource or compliance constraints.

**P (Priority)**: Optional field indicating importance ranking or resource allocation strategy.

This structure can be passed directly to model APIs that support structured prompts as a prompt\_structure or similar field, allowing the model to parse by field and execute generation tasks.

## Appendix: Quick Start Templates and Reference Materials

### Quick Template (Image Task)

X: Location or Category (Example: Top-right corner of the frame)

Y: Subject Element (Example: An orange shorthair cat)

Z: Style & Details (Example: Photorealistic style, resolution 2048×1152)

T: Steps or Timing (Example: T1: Generate background; T2: Place subject; T3: Lighting rendering)

### Extended Template (XYZT++)

M: Modality Combination (Example: ["image", "audio", "subtitle"])

C: Constraints (Example: {"max\_duration\_sec": 30, "no\_brand": true})

P: Priority (Example: ["Subject clarity", "Subtitle accuracy"])

### References & Recommendations

When publishing experimental data in public, it is recommended to include complete experimental protocols, task samples, and scoring criteria to facilitate reproduction and verification by others.

For engineering implementation, it is recommended to define XYZT as a set of JSON/YAML schemas and provide automated prompt generation functions and visual editors in internal tools.



## Conclusion

By structuring prompts into four dimensions—*Horizontal Positioning, Vertical Definition, Detailed Parameters, and Time & Steps*—the XYZT Prompt Framework provides a universal format that meets the needs for cross-modality, phased execution, and fine-grained control. This framework is applicable to various scenarios such as creation, development, and project management, and has strong potential for engineering and promotion. With future extensions to XYZT++ and larger-scale experimental verification, XYZT is expected to become one of the industry practice standards for multimodal prompts and task scheduling.

（注：文档部分内容可能由 AI 生成）