Constraint-Based Creative Generation with AI

Models: Gemini 2.5 Flash-Lite | Gemini 2.5 Flash | Gemini 2.5 Pro Preview

Kusai Aljuhmani

1. Introduction

This report presents dance choreographies generated by three Gemini language models—Gemini 2.5 Flash-Lite, Gemini 2.5 Flash, and Gemini 2.5 Pro Preview—under varying temperature settings (0.01–1.0) and two distinct prompts (base prompt and modified).

The study examines how formal constraints embedded in a rule-based validator shape generative output and how model capability interacts with these constraints. As noted in the project guidelines, constraints in creative domains play a generative role, limiting the solution space while simultaneously shaping form and structure. This project separates expressive generation (natural language) from formal validation, creating an iterative loop that reflects established models of creative cognition. [1]

In total, 18 valid choreographies across 3 models, up to 5 temperature settings, and 2 prompt variants are introduced, yielding a rich dataset for cross-model and cross-condition comparison.

2. Methods and Experimental Setup

2.1 Models and Conditions

Three models were evaluated via the Google Gemini API, called programmatically in Python:

- Gemini 2.5 Flash-Lite — a lightweight, fast model; tested at temperatures 0.01, 0.1, 0.3, 0.7, and 1.0. [2]
- Gemini 2.5 Flash — a mid-tier model; tested at temperatures 0.1, 0.3, 0.7, and 1.0.
- Gemini 2.5 Pro Preview — the largest model; tested at temperatures 0.1, 0.3, 0.7, and 1.0.

Both a base and a modified prompt were applied to each model–temperature combination. All API calls were made programmatically. When a choreography failed validation, the system initiated an automated iterative refinement loop. A feedback prompt was dynamically generated, this generate-validate-refine cycle repeated for up to six iterations or until the choreography passed all physical constraints.

2.2 Prompt Design

Base Prompt: The baseline condition used a zero-shot prompt instructing the model simply to generate a dance choreography for 2–5 dancers with approximate timing and location descriptions.

Modified Prompt: leveraged advanced reasoning strategies to enhance generation quality.

- Meta-Prompting: We defined an "expert Choreography Design System" persona to prime domain expertise. The prompt enforced a "Meta-Analysis & Setup" phase—requiring the definition of style, mood, and narrative—to ensure physical movements were grounded in a coherent semantic structure rather than generated randomly. [3]

- Chain-of-Thought (CoT): To mitigate spatial-temporal complexity, the prompt instructed the model to "think step by step" and plan spatial usage before generating coordinates. This intermediate reasoning step reduced cognitive load, leading to improved logical consistency in the final output. [4]

Feedback prompt: dynamically generated, containing a structured list of specific violations (e.g., 'Dancer A collides with Dancer B at t=34s', 'Speed limit exceeded at t=12s'). This error report, combined with the original

erroneous JSON, was fed back into the model with explicit instructions to 'correct every violation' while maintaining the original narrative arc.

2.3 Validation

Each generated JSON was validated by a Python rule-checker enforcing: (i) spatial boundaries—all positions within [0, 24]×[0, 24]; (ii) collision avoidance—no two dancers within a minimum proximity at any timestep; (iii) movement speed limits—displacement per unit time below a maximum threshold; (iv) entry/exit coherence—dancers must enter before performing actions and exit after their last event.

3. Content Analysis of Valid Choreographies:

Label	Model	Temp	Prompt	Dur(s)	Dancers	Events	Ev/s	AvgSpd	UniAct	Prtnr%
flash-lite 0.01 base	Flash-Lite	0.01	Base	75	3	42	0.56	0.68	28	7.1
flash-lite 0.01 mod	Flash-Lite	0.01	Mod	90	4	50	0.56	0.76	25	0.0
flash-lite 0.1 mod	Flash-Lite	0.10	Mod	90	4	38	0.42	0.45	32	0.0
flash-lite 0.7 mod	Flash-Lite	0.70	Mod	90	4	55	0.61	0.41	45	0.0
flash 0.1 base	Flash	0.10	Base	78	3	33	0.43	1.08	21	0.0
flash 0.1 mod	Flash	0.10	Mod	78	3	49	0.63	0.92	33	16.3
flash 0.3 base	Flash	0.30	Base	73	3	39	0.53	1.21	17	59.0
flash 0.3 mod	Flash	0.30	Mod	75	3	52	0.69	0.69	42	11.5
flash 0.7 base	Flash	0.70	Base	85	3	38	0.45	1.24	26	7.9
flash 0.7 mod	Flash	0.70	Mod	77	3	29	0.38	0.82	24	13.8
pro 0.1 base	Pro	0.10	Base	90	3	56	0.62	0.83	27	10.7
pro 0.1 mod	Pro	0.10	Mod	80	3	34	0.42	0.86	27	38.2
pro 0.3 base	Pro	0.30	Base	90	3	56	0.62	0.95	30	8.9
pro 0.3 mod	Pro	0.30	Mod	75	3	41	0.55	0.91	33	34.1
pro 0.7 base	Pro	0.70	Base	85	3	55	0.65	1.15	38	12.7
pro 0.7 mod	Pro	0.70	Mod	90	3	43	0.48	0.77	30	18.6
pro 1.0 base	Pro	1.00	Base	90	3	49	0.54	1.24	27	12.2
pro 1.0 mod	Pro	1.00	Mod	90	3	40	0.44	0.66	30	25.0

Table 1. Summary statistics for all 19 valid choreographies. Ev/s = events per second; AvgSpd = mean movement speed (grid units/s); UniAct = unique action vocabulary size; Prtnr% = partner interaction (%).

Based on the summary statistics in Table 1, distinct behavioral profiles emerge for each model:

- Gemini Flash-Lite: It is the least collaborative, showing 0.0% partner interaction in three out of four valid runs. Its vocabulary is moderately varied (25–45 unique actions), but the movement speed is generally slow (0.41–0.76 grid units/s), reflecting a cautious generation strategy.
- Gemini Flash: This model tends toward high-velocity movement, with average speeds frequently exceeding 1.0 grid units/s (e.g., 1.24 units/s at temp 0.7). This aggressiveness correlates with the high rate of speed violations observed in the error logs. Partner interaction is sporadic, ranging from 0% in base prompts to 59% in one specific low-temp condition, suggesting high variance in its grasp of collaborative constraints.
- Gemini Pro: The most robust model, achieving valid results across all temperatures and prompts. It demonstrates a sophisticated understanding of collaboration, maintaining consistent partner interaction (8.9%–38.2%) regardless of the prompt used. Its movement speeds are balanced (averaging ~0.9 grid units/s), indicating it internalizes the "speed limit" constraint better than Flash without sacrificing complexity.

4. Validation Runs and Iterative Refinement

4.1 Summary of Attempts

Table 2 reveals a clear hierarchy in model reliability. Gemini Pro is highly efficient, requiring an average of only 2 to 4 runs to produce a valid choreography, even at high temperatures. It was the only model to successfully generate valid content at Temperature 1.0.

In contrast, Gemini Flash-Lite failed to converge (>6 runs) for all base prompt conditions except the near-deterministic temperature of 0.01. However, the modified prompt acted as a crucial stabilizer, allowing Flash-Lite to succeed at temperatures 0.1 and 0.7 within 2–3 runs. This confirms that for smaller models, "prompt engineering" serves as a necessary scaffold to compensate for lower reasoning capabilities.

Gemini Flash occupied the middle ground, generally succeeding but struggling significantly at Temperature 1.0, where it consistently failed to converge regardless of the prompt

Model / Temperature	Base – runs to valid	Base valid?	Mod – runs to valid	Mod valid?
Pro / temp = 1.0	2	**YES**	2	**YES**
Pro / temp = 0.7	4	**YES**	3	**YES**
Pro / temp = 0.3	2	**YES**	2	**YES**
Pro / temp = 0.1	2	**YES**	2	**YES**
Flash / temp = 1.0	>6	**NO**	>6	**NO**
Flash / temp = 0.7	4	**YES**	3	**YES**
Flash / temp = 0.3	5	**YES**	1	**YES**
Flash / temp = 0.1	2	**YES**	3	**YES**
Flash-Lite / temp = 1.0	>6	**NO**	>6	**NO**
Flash-Lite / temp = 0.7	>6	**NO**	2	**YES**
Flash-Lite / temp = 0.3	>6	**NO**	>6	**NO**
Flash-Lite / temp = 0.1	>6	**NO**	2	**YES**
Flash-Lite / temp = 0.01	4	**YES**	2	**YES**

Table 2. Runs required to reach a valid choreography. Green = valid within ≤6 runs; orange = reached run 6 without a valid result. Base and modified prompt conditions are shown separately.

4.2 Error Type Breakdown

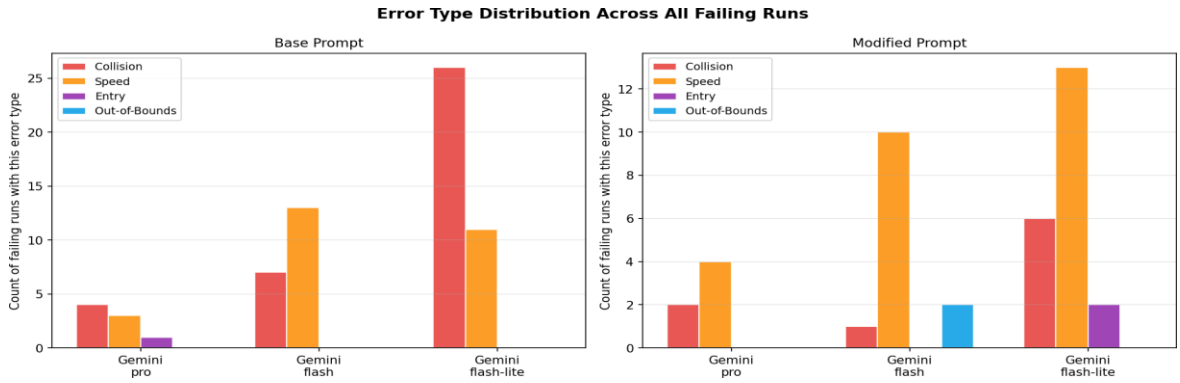

Figure 1. Count of failing runs per error type (collision, speed, entry/exit, out-of-bounds) broken down by model and prompt. Left: base prompt; right: modified prompt.

Gemini Flash showed a persistent tendency toward speed violations in both conditions, aligning with the high-velocity profile noted in the model summary. In contrast, Gemini Pro displayed the fewest errors overall, demonstrating robust adherence to both spatial and temporal constraints regardless of the prompt used. Gemini Flash-Lite struggled profoundly with spatial awareness under the base prompt, registering over 25 collision errors. Which became speed violations with the modified prompt.

4.3 Convergence Curves

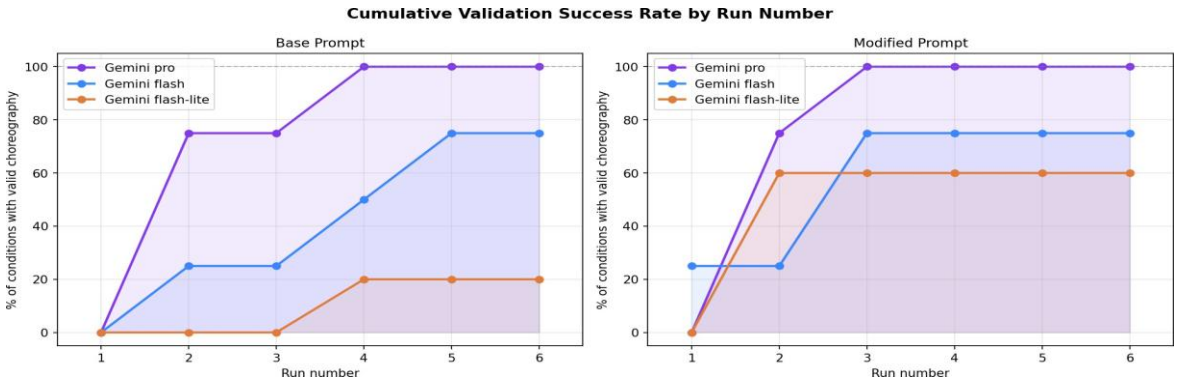

Figure 2. Cumulative percentage of conditions with a valid choreography by run number. A steeper initial rise indicates faster convergence. Left: base prompt; right: modified prompt.

The convergence curves confirm the ordering Pro > Flash > Flash-Lite in terms of generation reliability. Gemini Pro reaches 100% valid conditions by run 4 in the base prompt and by run 3 in the modified prompt. Flash converges to 75% (three of four conditions) by run 5 in both prompt types, with the temperature-1.0 condition never resolving. Flash-Lite reaches only 20% (one of five) by run 4 under the base prompt, but the modified prompt accelerates convergence markedly—two additional conditions validate by run 2, bringing the total to 60% (three of five). This is the clearest quantitative evidence that the modified prompt improves generation quality specifically for the smallest model.

5. Spatial Organization

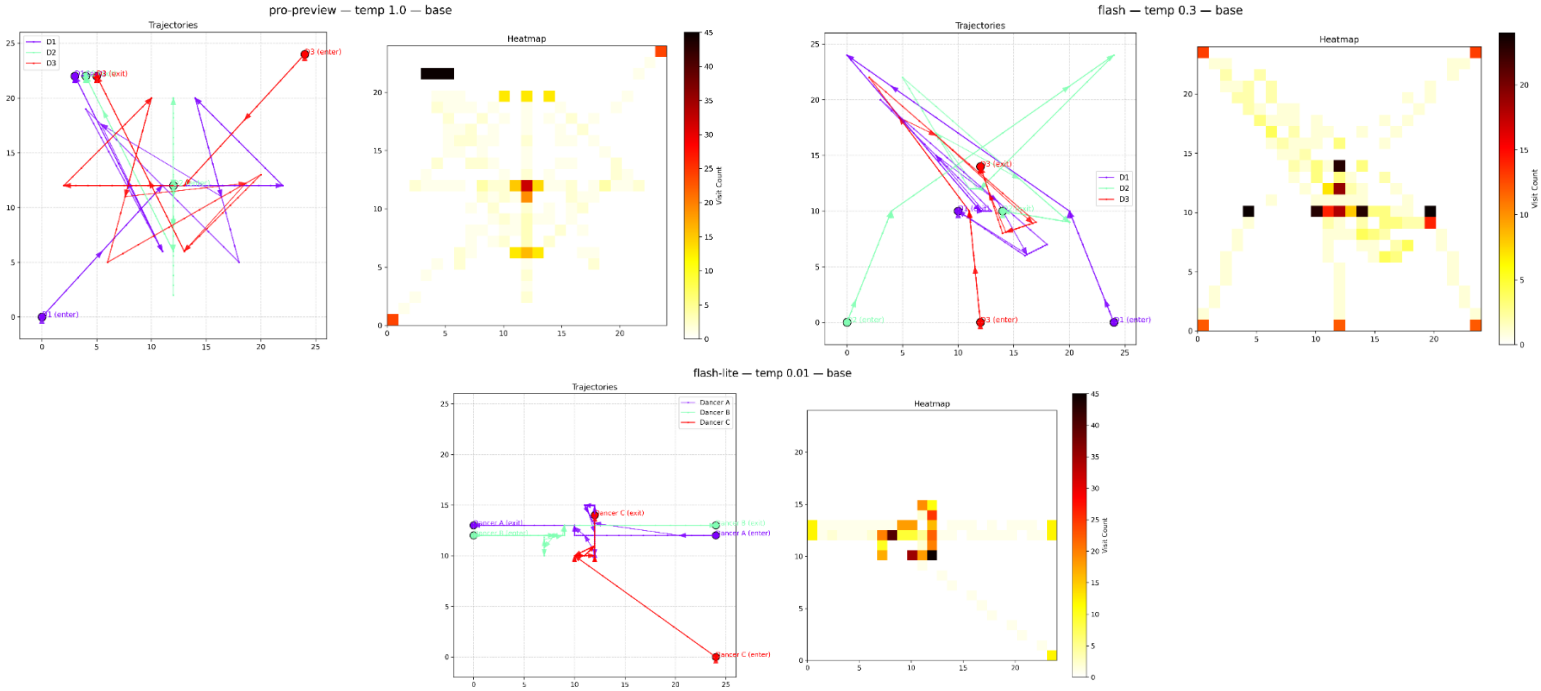

Figure 3. Trajectories and Spatial Occupancy Heatmap, gemini pro, flash, and flash-lite

Gemini Pro exhibits radial symmetry, where dancers explore the periphery but consistently converge toward a high-density central anchor to interact. This organization supports a coherent narrative structure that balances individual motion with ensemble synchrony.

In contrast, Gemini Flash displays a chaotic spatial topology composed of erratic, long-distance vectors that crisscross the grid without a unified focal point. While it utilizes the full stage, the movement shape is jagged and asymmetric.

Finally, Gemini Flash-Lite adopts a rigid, linear logic, confining movement largely to a single horizontal band with almost no curvature. Its choreography lacks simultaneous flow, relying instead on sequential, rectilinear

paths where dancers move one at a time to strictly avoid collisions.

6. Temporal and Collaborative Analysis

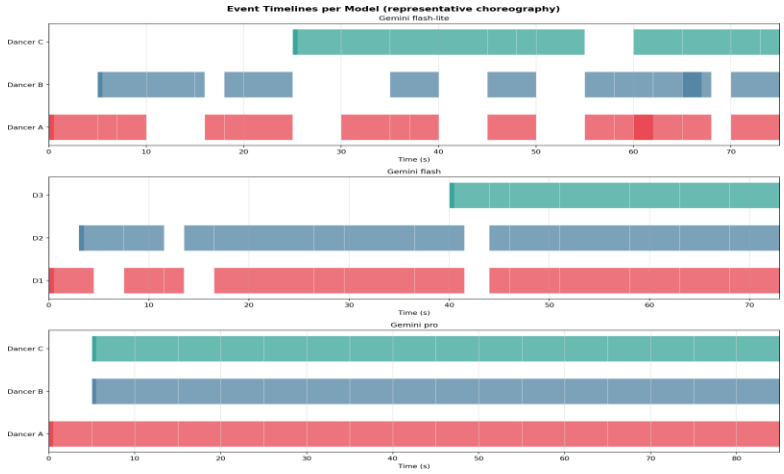

Figure 4. Event timelines per model (representative choreography). Horizontal bars = events; vertical axis = individual dancers.

Flash-Lite produces mostly sequential event chains with very limited overlap, where dancers tend to complete one phrase before the next begins, resulting in a sparse and linear temporal structure. Flash, by contrast, generates denser and more concurrent activity, with multiple dancers performing overlapping events that create a noticeably richer and more dynamic timeline. Pro goes produces the most coordinated and evenly distributed event patterns, where overlaps appear intentional and rhythmically aligned, leading to a more coherent and synchronized overall choreography.

7. Conclusion and discussion

This study demonstrates that constraint-based generation with Gemini language models produces structurally coherent dance choreographies, with quantifiable stylistic differences attributable to model capability, temperature, and prompt design. Key findings:

- Model capability is the primary determinant of constraint satisfaction. Pro produced valid choreographies in 100% of conditions; Flash in 75%; Flash-Lite in only 20–60% depending on prompt type.
- Temperature = 1.0: Both Flash and Flash-Lite consistently failed to produce valid choreographies at this temperature regardless of prompt, while Pro succeeded.
- The modified prompt improves both content quality and constraint satisfaction simultaneously. It raised the action vocabulary (+5 unique actions), reduced movement speed (−0.33 units/s), and cut average runs-to-valid by 35% for Flash and 50% for Flash-Lite.

8. References

[1] Chutaux, C. (2026). Creativity in AI as Emergence from Domain-Limited Generative Models. arXiv. https://doi.org/10.48550/arXiv.2601.08388

[2] Li, L., Sleem, L., Gentile, N., Nichil, G., & State, R. (2025). Exploring the Impact of Temperature on Large Language Models: Hot or Cold? arXiv. https://doi.org/10.48550/arXiv.2506.07295

[3] Zhang, Y., Yuan, Y., & Yao, A. C.-C. (2023). Meta Prompting for AI Systems. arXiv. https://doi.org/10.48550/arXiv.2311.11482

[4] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. arXiv. https://doi.org/10.48550/arXiv.2205.11916