Shin-Dialectics 2.0: A Dialectical Prompting Architecture for AI-Human Co-Thinking

Shinya Handa
Independent Researcher
v2025.08.11

Abstract
Shin-Dialectics 2.0 is a thinking architecture that replaces the linear "thesis→antithesis→synthesis" pattern with **Simultaneous Multiple Negation ($E_2(n)$)** and a **λ-thresholded three-layer pipeline** that co-controls divergence and convergence. It operationalizes **question dialectics**, **internal vs. external negation**, and **creative tension** to produce qualitative shifts in understanding. We provide a minimal, reproducible demo and SHA-256 hashes that fix priority at tag `v2025.08.11`.

Keywords: dialectics, prompting, $E_2(n)$, λ-thresholds, divergence, convergence, co-thinking, reproducibility

---

1. Motivation
Linear prompt patterns optimize a single axis (e.g., depth or breadth) and saturate quickly. Modern tasks require *parallel criticism* and *structured integration*. Shin-Dialectics 2.0 offers a compact OS-like procedure that scales across domains.

2. Contributions
- **$E_2(n)$:** concurrent generation of many antitheses/criticisms under multiple logics.
- **λ-thresholded pipeline:** three layers that co-control *diverge → converge → integrate*.
- **Question dialectics:** formal handling of meta-questions and premise shifts.
- **Reproducibility:** public timestamped release + SHA-256 hashes.
- **Minimal demo:** tasks and rubrics to compare against mainstream baselines (CoT/ToT/etc.).

3. Method

3.1 Simultaneous Multiple Negation ($E_2(n)$)
Given an initial thesis *T*, generate $\{A_1 \cdots A_n\}$ across heterogeneous lenses (logic, domain, value systems). Encourage **independent failure modes** rather than paraphrases.

3.2 λ-Thresholds
Let λ_p, λ_a, λ_s, λ_i be thresholds for the *quantity / diversity / synthesis quality / information sufficiency*. The process only proceeds when each λ is met—preventing premature convergence.

3.3 Three-Layer Pipeline (SIP)
1) **Diverge:** produce $E_2(n)$ with constraints (orthogonality, novelty).
2) **Converge:** cluster/score antitheses; eliminate redundancy; keep highest-tension items.
3) **Integrate:** synthesize candidates; test against counter-examples; select with rubric.

Pseudocode (sketch):
```
inputs: T, lenses L, thresholds λ
A = generate_antitheses(T, L)          # E2(n)
A' = select_by_novelty_tension(A, λa)  # converge
S  = integrate_candidates(T, A', λs)   # synthesize
return S, audit_trail
```

4. Demo & Reproducibility
We include three minimal tasks (idea generation → integration, research planning, value-conflict mediation). Each is run with (a) **SIP (ours)** and (b) **baseline** (CoT/ToT). Rubrics and prompts are provided.

- **Repo:** <https://github.com/hand-shinya/shin-dialectics-priority-2025-08-09>
- **Canonical release:** <https://github.com/hand-shinya/shin-dialectics-priority-2025-08-09/releases/tag/v2025.08.11>
- **Hashes:** `docs/HASHES.txt` (verify locally; must match)

5. Results (summary)
Across the demo tasks, SIP shows (i) higher diversity under equal token budgets, (ii) fewer mode-collapse failures, (iii) clearer audit trails. Full tables and rubrics are in `demo/`.

6. Related Work
Chain-of-Thought (Wei et al., 2022), Tree-of-Thought (Yao et al., 2023), Graph-of-Thought, Self-Ask, Debate/Multi-Critic frameworks, Reflexion. Our focus is *parallel* criticism ($E_2(n)$) with *explicit λ-gates* that govern phase transitions, lifting "multi-critic prompting" to an OS-level procedure.

7. Limitations & Future Work
Token cost increases with n and lens diversity; naive parallelism can degrade model faithfulness. Future work: adaptive lens selection, automated λ-tuning, and human-in-the-loop evaluation at scale.

8. Ethics Statement
The method can surface sensitive criticisms. We recommend red-team filters and domain-expert review in high-stakes contexts.

9. Availability & Citation
- Repo / Release: links above

- Medium overview: <https://medium.com/@handa.shinya/shin-dialectics-2-0-from-linear-dialectics-to-simultaneous-multiple-negation-069703a637d4>
- DOI (to be assigned via Zenodo): **TBD**

Acknowledgments
I thank early readers and the open-source community for feedback through public channels. Large language models were used as tools for drafting and checking; all interpretations and remaining errors are my own. This work is independent and unaffiliated.