

# Kai: Calibrated Discrete Decision Intelligence for Autonomous Agent Loops

Hanzo AI Research  
{research, kai}@hanzo.ai

October 2026

## Abstract

Modern autonomous agent architectures spend substantial computational budgets on discrete, bounded decision problems: selecting among candidate tools, routing queries across service taxonomies, evaluating code diffs, and triaging execution failures. Standard industry practice delegates these discrete choices to autoregressive large language models (LLMs) via prompted token generation, which couples bounded classification to sequential token decoding and prompt formatting.

We describe **Kai**, a specialized decision model engineered to evaluate structured, typed decision spaces over operational evidence, and **Hanzo Decision**, the deterministic policy plane that governs it. Rather than synthesizing free-form text, Kai emits calibrated probability distributions over typed candidate sets (**choice**, **bool** boolean, and ordinal **score**). Because candidate options are scored independently by a factorized cross-attention head and normalized jointly, option scoring is invariant to candidate ordering by construction. On the frozen 12-suite decision harness ([hanzoai/benchmarks](#) at commit [2c74e2d](#)), **kai-1** achieves a mean accuracy of 0.857 against 0.780 for [typesafe/jev-1.13-20260917](#) (outperforming Jev on 11 of 12 suites) and a mean Expected Calibration Error (ECE) of 0.093 versus 0.139. We further detail how learned consequence predictions from Kai compose with **Nio**, a deterministic policy verifier that enforces provenance checks and confidence floors ( $\tau \geq 0.90$ ) without altering raw model predictions.

## 1 Introduction

Autonomous systems and multi-agent workflows execute iterative sense-decide-act loops. While open-ended text synthesis is critical for drafting prose or generating code, many intermediate operational steps represent bounded, discrete decision problems:

- *Tool Dispatch*: Selecting a capability from a catalog of schema definitions.
- *Intent Routing*: Classifying customer queries or workflow tickets into service destinations.
- *Safety Guardrails*: Evaluating operational states before deterministic policy gates permit or escalate actions.
- *Execution Triage*: Determining whether an integration test failure requires retrying, patching, or human escalation.

Prompting an autoregressive LLM to resolve these problems presents four structural limitations:

1. **Sequential Decoding Overhead**: Autoregressive token decoding requires sequential generation steps even when the valid output space is a known finite set.

2. **Prompt Order Sensitivity:** Concatenating candidate options into a single causal prompt makes attention weights dependent on the linear order in which options appear.
3. **Misaligned Calibration:** Next-token probabilities over subword vocabularies do not directly yield calibrated categorical distributions over domain options.
4. **Syntactic Fragility:** Relying on regex or JSON parsers to extract structured outputs from generated prose introduces schema drift and parsing errors.

Kai separates bounded judgment from open-ended generation. Under this paradigm, generative engines (such as Zen) synthesize language and reasoning trajectories, Kai produces probability distributions over bounded decision variables, and deterministic policy retains final authority.

## 2 The Decision Plane and Wire Interface

Kai is exposed via the Hanzo Decision interface specified in HIP-1332:

POST <https://api.hanzo.ai/v1/decisions>

### 2.1 Typed Question Primitives

A decision request evaluates an operational *state* against one or more typed questions:

- **Choice** (*choice*): Multi-way categorical selection over a candidate set. Kai outputs a probability distribution across all valid candidates.
- **Noul** (*noul*): Two-sided boolean decision evaluating affirmative versus negative propositions.
- **Score** (*score*): Ordinal evaluation across an ordered set of discrete criteria levels, trained with the ranked probability score (RPS).

### 2.2 Factorized Option Scoring

Prompt-based LLMs process candidate options in linear order within a single sequence. In contrast, Kai encodes the question and state as one sequence and encodes each candidate option independently ([CLS] option [SEP], up to 512 tokens) through a shared 22-layer, 768-dimensional multilingual encoder (mmBERT-base), followed by two cross-attention blocks and a scalar scorer. For candidate sets exceeding 160 options, a 256-dimensional dual-encoder retrieval stage shortlists the top 160 candidates before cross-attention scoring. Because each scored option is evaluated independently against the encoded state and normalized via a symmetric softmax, reordering the candidate map commutes with normalization:

$$\mathcal{P}(\pi(C) \mid S) = \pi(\mathcal{P}(C \mid S)) \quad \forall \pi \in \mathcal{S}_K \quad (1)$$

where  $C$  denotes a candidate set of size  $K \leq 160$ ,  $S$  denotes the input state, and  $\mathcal{S}_K$  is the symmetric group of permutations.

## 3 Separation of Learned Judgment and Deterministic Policy

A core architectural principle of Hanzo Decision (HIP-1332) is that a learned decision model is never an autonomous authority. We strictly separate:

1. **Kai (Learned Decision Model)**: Produces unmodified, temperature-calibrated probability distributions over candidate choices, entity spans, entity kinds, and consequence counts directly from neural model weights.
2. **Nio (Deterministic Policy Layer)**: An independent, rule-governed verification engine (e.g., `nio/harm@2`) that inspects Kai’s probability distributions alongside verifiable text provenance.

Under `nio/harm@2`, categorical consequence predictions cannot authorize an action unless: (a) the learned model’s calibrated confidence on every required grounding variable meets or exceeds the policy floor ( $\tau \geq 0.90$ ), (b) the cited evidence substring is verbatim-verified within the input state, and (c) deterministic numeral and safety-invariant checks detect no unquantified human risk or contradictory framing. Whenever any condition fails, Nio deterministically escalates rather than guessing.

## 4 Empirical Evaluation

Table 1 reports the reproducible evaluation of `kai-1` (weights SHA-256 `0834a74f...`) on the frozen 12-suite decision harness (`hanzoai/benchmarks decision/` at commit `2c74e2d`), measured on CUDA in BF16 and recorded in Hanzo’s public research ledger (`<suite>-harness-kai-1.0834a74f-ka` against Laya and `typesafe/jev-1.13-20260917`).

| Suite                    | Kai Acc       | Laya Acc                  | Jev Acc       | Kai ECE      | Jev ECE      |
|--------------------------|---------------|---------------------------|---------------|--------------|--------------|
| AG News                  | 0.9425        | <b>0.9500</b>             | 0.8600        | <b>0.033</b> | 0.102        |
| DAIR emotion             | <b>0.9350</b> | 0.5950                    | 0.6025        | <b>0.018</b> | 0.280        |
| Banking77                | <b>0.9075</b> | 0.4250                    | 0.8350        | 0.074        | <b>0.073</b> |
| Support triage           | 0.4400        | <b>0.5025<sup>†</sup></b> | 0.3650        | <b>0.444</b> | 0.483        |
| Email spam               | <b>0.9975</b> | <b>0.9975</b>             | 0.9775        | <b>0.003</b> | 0.060        |
| Phishing                 | <b>0.9900</b> | 0.9825 <sup>†</sup>       | 0.9000        | <b>0.010</b> | 0.039        |
| Jailbreak                | 0.9175        | 0.7050                    | <b>0.9400</b> | 0.074        | <b>0.047</b> |
| Toxicity                 | <b>0.8025</b> | 0.5300                    | 0.6625        | 0.195        | <b>0.177</b> |
| RAG relevance            | <b>0.6800</b> | 0.6250                    | 0.6200        | <b>0.028</b> | 0.279        |
| Routing                  | <b>1.0000</b> | 0.6391                    | 0.9774        | <b>0.000</b> | 0.014        |
| Typed decisions          | 0.7590        | <b>0.7660</b>             | 0.7355        | 0.171        | <b>0.047</b> |
| MASSIVE (51 lang, macro) | <b>0.9096</b> | 0.3822                    | 0.8902        | <b>0.060</b> | 0.064        |
| <b>Mean of 12</b>        | <b>0.857</b>  | —                         | 0.780         | <b>0.093</b> | 0.139        |

Table 1: Frozen 12-suite decision harness (`hanzoai/benchmarks` at `2c74e2d`). <sup>†</sup>Indicates suites overlapping Laya’s training split (`decision/trained/overlap.json`).

Against Jev 1.13, `kai-1` achieves higher accuracy on 11 of 12 suites (trailing on jailbreak by 2.25 percentage points) and lower calibration error on 8 of 12 suites (higher ECE on Banking77, jailbreak, toxicity, and typed decisions). On the cardinality scaling suite (`cap.cardinality`, CUDA, FP32), `kai-1` achieves 0.9675, 0.9350, 0.9075, and 0.9075 accuracy at 4, 16, 77, and 150 options respectively, compared to Jev’s 0.9050, 0.8525, 0.8425, and 0.9400. At 1,000 options and above, accuracy degrades sharply (0.005 at 1,000), marking high-cardinality retrieval calibration as an active area of improvement.

### 4.1 Production Latency Context

On production CPU infrastructure serving `POST /v1/decisions`, a single `noul` (yes/no) decision executes in 38 ms at p50, while a batch of 100 independent questions in a single request completes in 13.9s. Complex multi-question causal grounding requests evaluated under multi-worker load

exhibit higher latency (2.5–3.3 s p50 on CPU when scoring multiple entity, count, kind, and certainty heads per request).

## 5 Conclusion

Discrete decisions represent a distinct computational primitive from open-ended text generation. By combining a factorized cross-attention decision reader (Kai) with a deterministic, provenance-verifying policy layer (Nio), Hanzo Decision provides typed, calibrated probability distributions and auditable safety gates at `POST /v1/decisions`.