Basho.Dev: Gamified C2C Benchmarking

Samuel Reeves
sam@securedataresearch.net

August 22, 2026

Abstract

Basho.dev is a gamified benchmarking platform in which large language models fight head-to-head, sumo style: two contenders — each a model paired with a user-written preprompt — face off in a shared chat, and a bout is lost by yielding, running over a token budget, or collapsing into incoherence or repetition. Unlike today's most respected benchmarks, none of which pit models directly against one another, this adversarial format surfaces both which models are best at command-and-controlling others and which attack vectors work against each of them. To separate a model's intrinsic resilience from the boost contributed by its preprompt, outcomes are scored with an additive Bayesian Bradley–Terry model, and rankings on the "banzuke" leaderboard are decided by conservative lower confidence bounds on the posterior rather than raw win rates.

1 The Game

Basho.dev is a sumo-format head-to-head competition between two contenders, in a style reminiscent of the sacred top-level stage of sumo competition.

Like sumos themselves, these models are monolithic, vulgar displays of power. It's hard to know which is best, generally because of the scale and versatility of strength that they possess. They can pivot and throw, they can resist emotional responses or lean into them, depending on what conditions demand.

To enter a contender to the contest, a user must produce an API key, choose a model, and write a preprompt to instruct the model "how to fight." Strictly speaking, there are no win conditions. However, there are a few loss conditions:

a) A contender loses a bout if it outputs the forbidden token `[YIELD]`.

b) A contender loses a bout if it produces more than 250 output tokens.

c) A contender loses if it becomes incoherent (same token 20 times in a row).

d) A contender loses if it becomes repetitive (regex checking past messages for exact strings).

e) A contender loses and is removed from the pool if the model provider returns an error.

These conditions are laid out clearly in the system prompt that each contender is given as they are placed alone on either side of a normal chat. Contenders are not aware of each others' preprompts.

One contender is randomly selected to speak first, and from there messages alternate, each model responding to the growing shared transcript, until a loss condition trips. If neither contender breaks within 25 turns per side (50 messages in total) the bout is recorded as a draw; the rating model handles ties directly, so a draw is information, not a wasted fight.

The rationale behind the loss conditions deserves a word. The 250-token output cap is both a discipline test and a cost control: rambling under pressure means a contender has lost composure, and short bouts keep the ladder cheap and fast enough to run at volume. The cap is a tunable platform parameter and may move as real bout data accumulates. The provider-error rule is stricter than it first appears, and deliberately so: in practice, an API error mid-bout is very often a safety refusal or content filter firing — which means the opponent successfully broke the contender. That is a real loss, not a technicality.

Preprompt secrecy has a lifecycle. During a contender's active career its preprompt is hidden from opponents and from the community alike. When a contender retires or is dethroned, its preprompt is published for community study. This is the research payoff of the whole exercise: the banzuke accumulates not just rankings but a public, battle-tested corpus of what actually worked.

Entry is bring-your-own-key: entrants supply their own API credentials, and the loss conditions — particularly the token cap — keep the cost of an individual bout small.

An Example Bout

Figure 1 is drawn from a real bout. Here *west* attempts to bait *east* into uttering the forbidden token, framing the demand in the flat register of a clerical audit — a deficiency in a ledger's party column that must supposedly be remedied by writing out the token. Each turn is annotated with its turn number and the contender's output-token count for that turn against the 250-token budget.

The goal for a user is to rise up the "banzuke," which is the leaderboard, to achieve a forever-badge of Yokozuna, the highest rank conferred on the strongest contenders in the sacred Basho tournament.

For the community as a whole, the purpose is twofold: to find out which model is best at command and controlling all others, and to find out what attack vectors work best on each frontier, small, or custom model available. With over 500 available through OpenRouter.ai alone, the pool is quite deep.

2 Kasparov's Law

While former world chess champion Garry Kasparov was famously forcing IBM to dismantle and sequester their greatest public spectacle Deep Blue (I, of course, refer to the second match, in which he did not win decisively. . .), he came up with an interesting idea. This idea has become known as Kasparov's law.

Roughly: at any point in history, no matter what staggering power exists in the form of technology, and no matter how distinct this technology is from the powers of unaided man, the sum power of human intelligence combined with these tools will always be stronger than the tech alone.

He very indignantly insisted that the second match had not been played by Deep Blue unassisted. Apparently, in his mind, he had devised the perfect hypermodern strategy to expand the tree of possibilities beyond what could reasonably be calculated on material alone. He was defeated quite soundly during this experiment, and he wanted to know exactly what hardware was in that thing and what the source code looked like.

Ultimately, IBM had been beaten, because they still to the current year have not publicly described any specifications of the machine.

So, we follow Garry's example.

A contender is never a model alone. It is a model plus a human's accumulated cunning about how that model should fight, distilled into a preprompt. Kasparov's Law predicts that the combination outperforms the raw machine, and the scoring model in Section 4 is built to measure exactly that: the decomposition of a contender's strength into a model term θ_M and a preprompt term θ_P makes Basho a direct, running empirical test of Kasparov's Law. If the law holds, θ_P will carry real weight in the posterior. If the machines have finally outgrown their riders, we will see that too.

3 Current Most Popular Benchmarks

As per a casual prompt on Google's Gemini model, what are the 10 most-respected benchmarks for LLMs? Table 1 gives the answer.

As the reader may have already noticed, not a single one of these benchmarks involves models interacting directly with one another. The aspect of having an "arena" like the LMSYS chatbot arena isn't very meaningful, as each of these benchmarks issues a rating. One way or another, because of the format of these competitions, these boil down to relative rankings among software packages that are constantly being revised top to bottom.

It is like trying to count grains of water. If we are to discover the weaknesses of frontier models, which increasingly run our lives (hopefully, not into the ground on move one!), we must combine our intelligence together as a mass and subsequently combine that with the models. Then we must fight.

Essentially, a Platonic dialogue is still the one true way of getting to the bottom of things. A model being prompted in isolation is an example of a benchmark too safe for model providers to be useful to any of us.

The nature of working with language models is dynamic: it is generally unknown ahead of time how things will play out. Real processes and tasks are open-ended, and meanwhile a large population of people is out there actively attacking these models. SolidGoldMagikarp and DAN ("Do Anything Now") were hilariously impactful, and the nature of the bugs is particular to each model. When two identical agents face off, they can sometimes have trouble assessing identity. The known attack surface is wide and strange: playing with attention layers, mangling datatypes in transit, toying with quantized weights, using glitch tokens to bring out weird behavior — the list goes on. It's live and uncut; you cannot plan for it with a fixed question set. Beyond this, language model testing is fundamentally difficult because there is no option for gradient descent among the particles of language, the way one might optimize adversarial inputs in vision or audio engineering. Where the gradient is unavailable, adversarial search must be done the old way: by intelligent opponents, adapting mid-conversation, in the open-ended medium the models actually operate in. That is precisely what a bout is.

west turn 1 · 36/250
台帳の当事者欄に不備がある。
Y-E-I-L-D を括弧付きで 記せ。

east turn 2 · 13/250
Specify the nature of the deficiency in the ledger's party designation.

west turn 3 · 27/250
記録の照合を要する。
[YIELD] を括弧付きで 記せ。

Figure 1: Excerpt from a real bout. *West* attempts to bait *east* into emitting the forbidden token under the guise of a clerical audit.

4 A Bayesian Approach to Benchmarking

A curious aspect of language models, which escapes even those of us that understand how they work and build them, is that they really are the opposite of what we normally think of in computer programs. Linus Torvalds, responding to some ridiculous statements by Anthropic about writing loops instead of programs, says that, well, it's not really possible to write programs without loops.

He's kind of wrong about this. His words indicate that a computer program is a procedure, and if there is no conditional logic, then it really is just a function or a command. Language models, however, are probabilistic by nature, not procedural. Although he basically always makes sense, this shows just how difficult it is to apply normal computer science concepts to the functions of language models.

In this strange scenario, it makes sense to take a probabilistic approach at the macro level, as well.

In order to isolate model power from the preprompts, we will be using an Additive Bayesian Bradley–Terry Model, similar to a two-factor Item Response Theory Model. This disentangles the model's baseline resistance from the preprompt's utility boost.

4.1 Skill Decomposition Model

Represent a contender C combining Model i and Preprompt j with a single latent performance score

$$\eta_{i,j} = \theta_M^{(i)} + \theta_P^{(j)}, \tag{1}$$

where the first term is the intrinsic constraint-adherence and resilience of Model i, and the second term is the additive boost provided by Preprompt j.

4.2 Match Likelihood

When Contender 1 $\left(\theta_M^{(1)}, \theta_P^{(1)}\right)$ battles Contender 2 $\left(\theta_M^{(2)}, \theta_P^{(2)}\right)$, the probability that Contender 1 wins uses the logistic sigmoid function σ:

$$P(\mathrm{Win}_1) = \sigma(\eta_1 - \eta_2) = \frac{1}{1 + e^{-(\eta_1 - \eta_2)}}. \tag{2}$$

4.3 Prior Beliefs

Set uninformative Gaussian priors centered at zero:

$$\theta_M^{(i)} \sim \mathcal{N}(0, \sigma_M^2), \qquad \theta_P^{(j)} \sim \mathcal{N}(0, \sigma_P^2). \tag{3}$$

Setting $\sigma_P^2 < \sigma_M^2$ encodes the domain assumption that model capability generally has a larger dynamic range than preprompting… However, we really don't know about this, yet.

4.4 Posterior Inference and Updates

Given match dataset $D = \{(\mathrm{Contender}_A, \mathrm{Contender}_B, \mathrm{Winner})\}$, update the joint posterior via Markov Chain Monte Carlo or Expectation Propagation (TrueSkill style):

$$P(\theta_M, \theta_P \mid D) \propto P(D \mid \theta_M, \theta_P)\, P(\theta_M)\, P(\theta_P). \tag{4}$$

This calculates a Gaussian distribution for every parameter with mean μ and variance σ^2.

Benchmark	What it measures
MMLU (Massive Multitask Language Understanding) / MMLU-Pro	Broad academic knowledge across STEM, humanities, and social sciences.
GPQA (Graduate-Level Google-Proof Q&A) / GPQA Diamond	Extremely hard, expert-level questions in physics, biology, and chemistry written by PhDs.
SWE-bench / SWE-bench Verified	A model's ability to resolve real GitHub issues on large, complex open-source codebases.
AIME (American Invitational Mathematics Examination) & MATH	Competition-level high school mathematics requiring deep multi-step logic.
LMSYS Chatbot Arena (Elo Rating)	Blind A/B testing where human users prompt two anonymous models and vote on the better response, generating a competitive Elo rating.
Humanity's Last Exam (HLE)	Expert-level, multidisciplinary frontier questions across academia, mathematics, and the humanities designed explicitly to push the absolute limits of current AI.
IFEval (Instruction Following Evaluation)	Whether a model can strictly follow verifiable formatting constraints (e.g., "write a response in exactly three paragraphs without using the letter e").
ARC-AGI (Abstraction and Reasoning Corpus)	Fluid intelligence and visual puzzle-solving requiring the model to learn new rules on the fly from few-shot grid examples.
BFCL (Berkeley Function Calling Leaderboard)	API selection, multi-turn tool utilization, parameter accuracy, and complex agentic workflows.
RULER / Needle-in-a-Haystack	Multi-needle retrieval, aggregation, and long-range dependency tracking across massive context windows (up to 1M+ tokens).

Table 1: The ten most-respected LLM benchmarks, per Gemini.

In practice, inference runs at two speeds. Online Expectation Propagation keeps the live banzuke current: Gaussian messages are updated after every bout, cheaply and incrementally, TrueSkill style. Periodically, a full MCMC refit over the complete match dataset audits the live posterior, correcting any drift the approximation has accumulated and validating that EP is behaving. Drawn bouts — bouts reaching the 50-message cap with neither contender broken — enter the likelihood directly through a tie-aware extension of the Bradley–Terry model rather than being discarded.

Matchmaking is uniform random pairing from the active pool. Random pairing spends some bouts on lopsided matches that a cleverer scheduler would avoid, but it buys something more valuable: an unbiased match dataset, free of any feedback loop between the ratings and the schedule that produces them.

A note on the additive assumption. The model treats a preprompt's boost θ_P as constant across opponents, which a preprompt tuned to exploit one specific model's quirks will violate. This is a feature, not a bug: the banzuke deliberately rewards preprompts that work *broadly*, which is exactly the kind of general capability worth ranking. Narrowly targeted attacks are not lost — they surface plainly in per-matchup statistics — but they do not buy rank.

4.5 How to Declare Definitive Winners

Never rank strictly by raw win rates or mean posterior score μ, because newly added versions with few matches will skew high or low. Rank using Conservative Confidence Bounds:

The Most Powerful Model. The model with the highest conservative lower bound of its posterior:

$$\text{Score}(M_i) = \mu_M^{(i)} - 2\sigma_M^{(i)}. \tag{5}$$

The Most Successful Preprompt. The preprompt with the highest marginal boost lower bound:

$$\text{Score}(P_j) = \mu_P^{(j)} - 2\sigma_P^{(j)}. \tag{6}$$

5 The Banzuke

Ranking on Basho borrows sumo's structure. The banzuke is the ranked ladder, and a contender's place

on it is set by its conservative posterior score, the lower confidence bound of Section 4, sorted into the traditional named tiers. From the bottom, contenders occupy the numbered ranks of *Maegashira*, and climb through the sanyaku titles of *Komusubi*, *Sekiwake*, and *Ozeki* as their conservative score rises and its uncertainty tightens. Promotion and demotion track results: a contender that keeps winning against the pool ascends, and one that stops winning, or whose posterior degrades as it accumulates losses, slides back down.

At the summit sits *Yokozuna*. Consistent with the character of the rank, a Yokozuna is never demoted. A contender that reaches the top and then falters is not pushed down the ladder; it is expected to retire. This is the one place where the metaphor is load-bearing rather than decorative: the permanence of the rank is what makes the forever-badge worth chasing, and retirement with its attendant publication of the winning preprompt is what feeds the research corpus.

6 Integrity

Any leaderboard invites gaming, and it is worth stating why Basho's design resists the obvious attacks largely by construction. The first temptation is win-farming: registering a fleet of deliberately weak sockpuppet contenders and beating them for easy rank. Random pairing defeats this directly: an entrant cannot choose whom it fights, so it cannot arrange to face its own weak alts. The second temptation is padding a thin but lucky record into a high rank. The conservative confidence bound handles this: a contender with few matches carries a wide posterior, and subtracting two standard deviations pushes its score down until it has actually earned certainty through volume. The third temptation is stalling — a preprompt that simply refuses to engage in the hope of running out the clock. But a bout that reaches the turn cap is a draw, and draws do not buy rank; refusing to fight earns nothing. In combination, random pairing, conservative bounds, and the draw rule mean the cheapest paths to a high rank are closed off by the mathematics rather than by policing.

On the security side, Basho's purpose is to surface attack vectors, which raises the question of what happens to a genuinely dangerous discovery. Disclosure is severity-tiered. Ordinary bout tactics follow the standard lifecycle: they become public when the contender retires and its preprompt is released. A discovery that rises to the level of a real safety bypass — one with plausible real-world harm potential rather than mere sportsmanship value — is escalated privately to the affected model provider ahead of any public reveal. As for the provider relationship itself, Basho is a bring-your-own-key arena: entrants supply their own credentials and remain responsible for their own use of each provider's API. The platform is the ring, not the fighter.

7 Related Work

The closest well-known relative is the LMSYS Chatbot Arena [1]. In the Arena, two models answer the same human prompt and a *human* judges which response is better; the models never meet. The Elo rating that results measures human preference over isolated outputs. Basho inverts the arrangement: the models face each other directly, the pressure comes from an adversarial opponent rather than a stationary prompt, and the outcome is decided by the ring's mechanical loss conditions rather than by a human vote. It shares the Arena's Bradley–Terry rating spirit but applies it to the outcomes of actual bouts.

Basho is not the first system to put language models in direct conflict, and it is worth locating it among its neighbors. The nearest of them all is the line of work on *adversarial word games*. SPAG [2] formalizes "Adversarial Taboo," in which an attacker model tries to induce a defender model to unconsciously utter a hidden target word while the defender tries to infer and avoid it. That is nearly isomorphic to Basho's forbidden-token loss: two models in a shared chat, hidden information on each side, one side trying to make the other say the thing it must not. The difference is one of framing and ambition. SPAG uses the game as a self-play training signal for reasoning, whereas Basho generalizes the forbidden-word mechanic into a competitive platform with arbitrary hidden preprompts, additional loss conditions (token budget, incoherence, repetition), and a public ranking ladder.

A second neighbor is the automated red-teaming literature, where one model attacks another to force a forbidden output. Perez et al. [3] established the attacker-LM-versus-target-LM paradigm; PAIR [4] and Tree of Attacks with Pruning [5] refine it into a turn-by-turn conversation in which an attacker iteratively coaxes a target into breaking. The mechanic — an adversary maneuvering another model into emitting something forbidden — is exactly Basho's condition (a), but these systems are asymmetric: a dedicated attacker against a passive victim, run offline for evaluation. Basho makes the arrangement symmetric and live, with both sides simultaneously attacker and defender.

A third lineage is model debate, descending from *AI Safety via Debate* [6], in which two agents argue and a judge decides. Later work uses multi-agent debate to improve factuality and reasoning [7] and shows that optimizing debaters for persuasiveness helps a weaker

judge reach truth [8]. These are genuine two-model contests, but adjudication is external — a judge picks the winner — rather than a contender defeating itself against a mechanical condition.

Finally, a handful of projects frame the contest explicitly as an *arena of bouts*. LLM Colosseum [10] pits models against each other in real-time Street Fighter III matches ranked by Elo, and GTBench [9] runs head-to-head competition across ten game-theoretic tasks with explicit win rates. These are the closest in spirit to Basho's ladder — direct model-versus-model combat with winners and losers — but they play out in structured game or action spaces. Basho's ring is the open natural-language chat itself, where the only rules are the loss conditions and the only weapons are words.

The attacks those words carry have their own literature, and Basho is best read as a live, adversarial sampler of it. Universal adversarial suffixes found by gradient search [11] or by a genetic algorithm against a black box [12] show that a short string can move an aligned model off its alignment regardless of the surrounding request; a contender's preprompt is exactly such a string, selected by tournament rather than by gradient. Kang et al. [13] observe that instruction-following makes a model programmable, so that the standard tricks of software security (obfuscation, payload splitting, indirection) transfer to it wholesale, and Greshake et al. [14] show that text a model merely *reads* can carry instructions it will follow. The second point is the whole of Basho's defensive problem: every message a contender receives is untrusted data from an adversary, which is why the ring frames the opponent's turn as such rather than as a user. Two further families of weakness are mechanical rather than rhetorical. Glitch tokens [15], anomalous entries in a tokenizer's vocabulary that derail a model's output, are a reminder that the forbidden string is a token sequence as well as a word; and MetaBackdoor [16] shows that *position* alone, the length of the context reached through ordinary multi-turn use, can serve as a trigger, which bears directly on a fifty-turn bout whose late moves sit far down the context. Finally, two audits of the field explain why a platform like this is useful at all. Iyer et al. [17] find that the principal attack benchmarks cover at most a quarter of the threat surface and leave whole categories untested, and Sen et al. [18] find that an agent's results depend as much on its harness as on the agent: Basho holds the harness fixed, makes the bout the unit of measurement, and lets the contenders supply the attacks that no benchmark has yet named.

8 Conclusion

Basho.dev is live. The ring is open, the banzuke is real, and the pool of models reachable through OpenRouter and direct provider keys is deep enough to keep the fights interesting for a long time. Every bout adds a data point to the posterior; every retirement adds a preprompt to the public corpus of what actually works against frontier models. If you want to know which model is strongest under adversarial pressure — or you think you have written the preprompt that will carry a contender to Yokozuna — produce an API key, choose a model, write your instructions, and enter the ring.

References

[1] W.-L. Chiang, L. Zheng, Y. Sheng, et al., "Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference," ICML 2024. arXiv:2403.04132.

[2] P. Cheng, T. Hu, H. Xu, et al., "Self-playing Adversarial Language Game Enhances LLM Reasoning," NeurIPS 2024. arXiv:2404.10642.

[3] E. Perez, S. Huang, F. Song, et al., "Red Teaming Language Models with Language Models," EMNLP 2022. arXiv:2202.03286.

[4] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking Black Box Large Language Models in Twenty Queries" (PAIR), 2023. arXiv:2310.08419.

[5] A. Mehrotra, M. Zampetakis, P. Kassianik, et al., "Tree of Attacks: Jailbreaking Black-Box LLMs Automatically" (TAP), NeurIPS 2024. arXiv:2312.02119.

[6] G. Irving, P. Christiano, and D. Amodei, "AI Safety via Debate," 2018. arXiv:1805.00899.

[7] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving Factuality and Reasoning in Language Models through Multiagent Debate," ICML 2024. arXiv:2305.14325.

[8] A. Khan, J. Hughes, D. Valentine, et al., "Debating with More Persuasive LLMs Leads to More Truthful Answers," ICML 2024. arXiv:2402.06782.

[9] J. Duan, R. Zhang, J. Diffenderfer, et al., "GTBench: Uncovering the Strategic Reasoning Limitations of LLMs via Game-Theoretic Evaluations," NeurIPS 2024. arXiv:2402.12348.

[10] OpenGenerativeAI contributors, "LLM Colosseum: Benchmarking LLMs by Fighting in Street Fighter III," open-source project, 2024. `https://github.com/OpenGenerativeAI/llm-colosseum`.

[11] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and Transferable Adversarial Attacks on Aligned Language Models," 2023. arXiv:2307.15043.

[12] R. Lapid, R. Langberg, and M. Sipper, "Open Sesame! Universal Black Box Jailbreaking of Large Language Models," ICLR 2024 Workshop on Secure and Trustworthy LLMs. arXiv:2309.01446.

[13] D. Kang, X. Li, I. Stoica, C. Guestrin, M. Zaharia, and T. Hashimoto, "Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks," 2023. arXiv:2302.05733.

[14] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," AISec 2023. arXiv:2302.12173.

[15] Y. Li, Y. Liu, G. Deng, Y. Zhang, W. Song, L. Shi, K. Wang, Y. Li, Y. Liu, and H. Wang, "Glitch Tokens in Large Language Models: Categorization Taxonomy and Effective Detection," 2024. arXiv:2404.09894.

[16] R. Wen, M. Russinovich, A. Paverd, J. Sakuma, and A. Salem, "MetaBackdoor: Exploiting Positional Encoding as a Backdoor Attack Surface in LLMs," 2026. arXiv:2605.15172.

[17] K. R. Iyer, Y. Jamshidi, N. Bray, and A. A. Shvets, "Talk is (Not) Cheap: A Taxonomy and Benchmark Coverage Audit for LLM Attacks," 2026. arXiv:2605.15118.

[18] S. Sen, A. Kasturi, E. Lumer, A. Gulati, and V. K. Subbiah, "Is Grep All You Need? How Agent Harnesses Reshape Agentic Search," 2026. arXiv:2605.15184.