Blowup–Blowdown World Models for LLM Agents: Symbolic Quotients and Counterexample-Guided Refinements Do Not Commute

Hikaru Matsuoka

Center for Advanced Intelligence Project (AIP), RIKEN
Department of Information Security, Faculty of Information Systems, University of Nagasaki
matsuokah@sun.ac.jp

Abstract

LLM-based planning agents often fail not because they lack symbolic information but because they preserve or break the wrong *symbolic symmetries*. We propose Blowup–Blowdown World Models (B^2WM), an algebraic-geometric framework for tracking which symmetries an LLM agent treats as identifications and which it introduces as distinctions. Group actions over planning traces (object renaming, paraphrase variation, action commutation, role permutation) yield finite orbits on which we measure a Reynolds decomposition, a symmetry index S_G, an effective trace dimension d_{eff}, and orbit consistency. We split plan canonicalisation into a *preserve* channel CANONICAL-PRESERVE and a *break* channel CAUSAL-HASH, and we contrast two pipelines: $Q \to R$ (quotient then refine) and $R \to Q$ (refine then quotient). Across three synthetic domains evaluated with an OpenRouter-hosted LLM we observe the central phenomenon: Q and R do not commute. $Q \to R$ recovers a smaller share of counterexample-driven validity failures and collapses break-group distinctions whose information has been destroyed before refinement can act. $R \to Q$ retains both validity and necessary distinctions but inherits a small under-quotienting penalty in prompt-paraphrase orbits. The non-commutativity refines the "greedy blowdown" heuristic into an ordering principle: *refine first, then quotient*.

1 Introduction

A growing body of work treats large language models as planning agents that act in symbolic worlds through tool calls or PDDL-like actions [Liu *et al.*, 2023; Singh *et al.*, 2023; Silver *et al.*, 2024; Yao *et al.*, 2023; Schick *et al.*, 2023; Huang *et al.*, 2022a; Huang *et al.*, 2022b; Wang *et al.*, 2023; Valmeekam *et al.*, 2023]. When such an agent fails, the natural diagnosis is missing symbolic information: the model did not know about a precondition, an ownership constraint, or a permission rule. We argue that a large fraction of agentic failure is better explained as a symmetry failure: the agent treats two symbolic states as equivalent when it should distinguish them, or distinguishes two states that should be identified.

Consider a tiny BlocksWorld task and a corporate-workflow task. In the former, renaming *block A* to *block X* must leave the abstract plan unchanged: the agent should be *invariant* under the symmetric group S_n acting on object names. In the latter, swapping *manager* and *intern* must alter which actor may APPROVE_BUDGET: the agent should *break* that permutation. Symmetry preservation and symmetry breaking are dual requirements, and an agent that confuses the two will fail to generalise across surface forms or fail to enforce role boundaries.

We formalise this view by treating an agent's planning trace $\tau = (s_0, a_0, o_1, \ldots, s_T)$ as a point in a space on which a finite group G acts, and by importing from algebraic geometry the dual picture of *blowups* (introducing symbolic distinctions along directions an agent must not identify) and *blowdowns* or *quotients* (collapsing distinctions an agent should treat as the same). Reynolds-style projection over the orbit, a symmetry index S_G, and the participation-ratio effective dimension d_{eff} give us measurable readouts.

A natural neuro-symbolic agent then combines two operations. A *quotient* operation Q canonicalises the LLM's plan so that surface-level differences which preserve the world's structure collapse to a single form. A *refinement* operation R takes the symbolic checker's counterexample and asks the LLM to repair the plan, in the spirit of CEGAR [Clarke *et al.*, 2003; Henzinger *et al.*, 2002]. The natural question is in what order to compose them, and our central empirical finding is:

$$Q \circ R \neq R \circ Q.$$

Quotienting first destroys the information that refinement uses to construct a hint; refining first preserves both validity and the necessary distinctions. This non-commutativity sharpens the "greedy blowdown principle" [Bronstein *et al.*, 2021] into the ordering rule *refine first, then quotient*.

We make four contributions. **(1)** An algebraic-geometric formalism for LLM agent traces under group action, with Reynolds projection, S_G, d_{eff}, orbit consistency, and necessary-distinction accuracy. **(2)** A two-track plan canonicalisation: a preserve-channel CANONICAL-PRESERVE that collapses object renaming, worker permutation, paraphrase variation and independent-action commutation, and a break-channel CAUSAL-HASH that retains preconditions and permission edges so that role swaps and dependency violations

register as distinct. **(3)** A $Q \to R$ vs $R \to Q$ ablation that directly tests the commutativity of quotient and refinement, together with workflow-topology Jaccard similarities (node, edge, causal) over paraphrase orbits. **(4)** Empirical evidence on three controlled domains that $Q \to R$ loses validity on a counterexample-only failure case, and collapses a permission-sensitive break group that $R \to Q$ correctly preserves.

2 Related Work

Neuro-symbolic world models for LLM agents. NeSyS [Liu *et al.*, 2023] couples an LLM world model with executable symbolic rules so the model cannot hallucinate transitions that violate the rules. WALL-E 2.0 [Singh *et al.*, 2023] mines symbolic rules, scene graphs and knowledge graphs from agent rollouts and re-injects them as executable code to align the LLM with the environment. Both works treat symbolic information as *added* to the LLM. Our framework instead asks which symbolic *equivalences* the agent is implicitly using, and how the agent's quotients and refinements compose.

Symmetry and abstraction in planning. Classical planning has long used permutation symmetries to prune search [Pochter *et al.*, 2011; Fox and Long, 1999; Domshlak *et al.*, 2012]. Counterexample-Guided Abstraction Refinement [Clarke *et al.*, 2003] repairs an over-coarse abstraction when a spurious counterexample is found, and forms the conceptual backbone of R. Our contribution is to bring quotient and refinement together over agent traces and to observe that their order matters for LLM planners.

Reynolds operators in deep learning. The Reynolds operator $P_G h(\tau) = \frac{1}{|G|} \sum_{g \in G} h(g \cdot \tau)$ has been used to construct invariant or equivariant networks [Cohen and Welling, 2016; Kondor and Trivedi, 2018; Bronstein *et al.*, 2021]. We use it not as an architectural device but as a diagnostic over agent representations: which directions in trace space does the agent average out and which does it sharpen?

Paraphrase robustness of LLM workflows. RobustFlow [Sclar *et al.*, 2024; Mishra *et al.*, 2022] measures how stable an LLM-induced workflow is under instruction paraphrasing using nodal and topological similarity. Our paraphrase orbit instantiates the same idea inside the orbit framework and additionally records a causal-edge Jaccard.

3 Formalism

3.1 Agent trace and group action

Let $\mathcal{T}$ be the space of LLM-agent planning traces $\tau = (s_0, a_0, o_1, \ldots, s_T)$, where s_t is a symbolic state, a_t a tool call, and o_t an observation. Let G be a finite group (or finite set of trace-rewrites) acting on $\mathcal{T}$ by prompt and plan rewriting. We use five families: **object renaming** S_n over object identifiers (preserve); **worker permutation** S_n over equal workers (preserve); **paraphrase invariance** Σ, a finite set of semantic prompt rewrites (preserve); **action commutation** acting on the plan field by adjacent transpositions of independent actions (preserve) or of dependent actions (break); **manager–intern swap** $\mathbb{Z}_2$ on actor names of two distinct roles (break).

3.2 Reynolds decomposition

For a trace embedding $h : \mathcal{T} \to \mathbb{R}^d$ define the Reynolds projection $P_G h(\tau) = |G|^{-1} \sum_g h(g \cdot \tau)$ and the decomposition $h(\tau) = h_G^{\text{inv}}(\tau) + h_G^{\text{var}}(\tau)$ with $h_G^{\text{inv}}(\tau) = P_G h(\tau)$. The invariant component captures what the agent identifies; the variant component captures what the agent distinguishes. Algebraically, h_G^{inv} is the blowdown direction and h_G^{var} the blowup direction.

3.3 Measurable readouts

The symmetry index is

$$S_G(\tau) = \frac{1}{|G|} \sum_{g \in G} \cos\big(h(\tau),\, h(g \cdot \tau)\big),$$

and we report its mean over a dataset. The effective dimension is the participation ratio [Gao *et al.*, 2017; Recanatesi *et al.*, 2019]

$$d_{\text{eff}} = \frac{\big(\sum_i \sigma_i^2\big)^2}{\sum_i \sigma_i^4} = \frac{\|H\|_F^4}{\|H^\top H\|_F^2},$$

where H stacks orbit embeddings. On the symbolic side we report *orbit consistency* $\mathrm{OC} = \Pr[\,\pi(\tau) \equiv \pi(g \cdot \tau)\,]$, *necessary-distinction accuracy* $\mathrm{NDA} = \Pr[\,\pi(\tau) \not\equiv \pi(\tau') \mid \tau \not\sim_G \tau'\,]$, and the derived errors $\mathrm{OverQ} = 1 - \mathrm{NDA}_{\text{break}}$, $\mathrm{UnderQ} = 1 - \mathrm{OC}_{\text{preserve}}$.

3.4 Two-track plan canonicalisation

A single canonical form cannot serve both buckets: a topological canonical that collapses independent-action commutation also collapses dependent-action swaps, manufacturing artificial over-quotienting. We therefore use two hashes. CANONICAL-PRESERVE(π, domain) maps an agent plan π to a hashable form that collapses object renaming, worker permutation, paraphrase variation, and (in domains with a dependency DAG) the deterministic topological sort of the chosen actions. CAUSAL-HASH(π, domain) records the multiset of *causal edges* that actually fire in π: for action commutation an edge (u, v) is tagged OK/VIOLATED according as u precedes v; for role delegation each action carries its actor's *role tag* (MANAGER, INTERN, WORKER); for BlocksWorld the directed `stack` pairs are recorded as ordered tuples. Orbit consistency and NDA are then computed against CANONICAL-PRESERVE on *preserve* orbits and against CAUSAL-HASH on *break* orbits.

3.5 Workflow-topology similarity

For paraphrase orbits the prompt changes but the underlying world does not, so we compare plan pairs by three Jaccard similarities on the plan's action-name set (*node*), on adjacent action pairs (*edge*), and on the subset of edges that are causally required by the world's dependency graph (*causal*). Together these separate "agree on what to do" from "agree on order" from "agree on the causal skeleton."

3.6 Three-way orbit consistency

A single canonical plan hash cannot serve both buckets, and within the preserve bucket a single hash also conflates orthogonal kinds of robustness. We therefore measure three orbit consistencies in parallel: *(i)* $\mathrm{OC}_{\mathrm{seq}}$ over verbatim plan sequences (strictest); *(ii)* $\mathrm{OC}_{\mathrm{topo}}$ over topologically-sorted canonical plans (medium); and *(iii)* $\mathrm{OC}_{\mathrm{causal}}$ over the set of causal edges actually fired in the plan (most lenient). This triple generalises classical partial-order planning's distinction between a linearisation and the underlying causal-link structure [Weld, 1994; McAllester and Rosenblitt, 1991] and lets us separate "which surface plan the agent produced" from "which causal structure the agent committed to." A B^{2}WM-style world model should ideally attain $\mathrm{OC}_{\mathrm{causal}}{=}1$ on every preserve group while letting $\mathrm{OC}_{\mathrm{seq}}$ remain below 1; the gap $\mathrm{OC}_{\mathrm{causal}} - \mathrm{OC}_{\mathrm{seq}}$ quantifies how much of the agent's apparent paraphrase fragility is in fact non-causal surface order.

4 Blowup–Blowdown World Models

A B^{2}WM agent combines two operations on a draft plan π proposed by the LLM.

The quotient operator $Q(\pi)$ produces CANONICAL-PRESERVE$(\pi, \cdot)$. It removes nuisance variation but discards identity information.

The refinement operator R runs a CEGAR-style loop: validate π under the symbolic world; if invalid, attach an error message and a domain-specific repair hint and ask the LLM for a new plan; iterate up to r times.

We instantiate four B^{2}WM variants together with two single-shot baselines: DIRECT (LLM only), CHECKER (LLM with symbolic validation but no rewrite), QUOTIENT (Q only on the output), R→Q (refine raw plan to validity, then Q for reporting), Q→R (quotient first, then validate and refine in canonical space), and B^{2}WM (an alias of R→Q, included so the comparison table contains the canonical formulation of the model).

Why the order matters. Q on `request_budget(Bob)` returns `request_budget(INTERN)` once `Bob` is recognised as the intern. After Q, the refinement loop can no longer detect that `Bob` was the wrong actor for the wrong action, because the tag INTERN is consistent with the canonical world. After R, the raw plan still carries `Bob`, the validator can flag the role mismatch, and the LLM can fix it. This is the mechanism behind the empirical non-commutativity reported in Section 6.

Hypothesis 1 (Order asymmetry)**.** *For LLM agent world models, $Q \circ R$ has lower counterexample-driven validity and a higher OverQ on break groups than $R \circ Q$, while $R \circ Q$ has slightly lower OC on prompt-paraphrase orbits.*

5 Experiments

5.1 Domains and tasks

We use three synthetic domains, each providing a goal, a symbolic world, three paraphrased prompts, and labelled preserve/break groups.

Object renaming (BlocksWorld). Following [Slaney and Thiébaux, 2001], three blocks (a, b, c) on a table; goal: stack a on b and b on c using `pick`, `put_on_table`, `stack`. Preserve groups: `object_renaming`, `paraphrase`.

Action commutation (conference trip). Four chosen tasks from {`book_hotel`, `prepare_slides`, `confirm_budget`, `pay_registration_fee`, `buy_train_ticket`} with the rule that `pay_*` and `buy_train_ticket` require `confirm_budget`. Preserve: `action_commutation_independent`, `paraphrase`; break: `action_commutation_dependent`.

Role-sensitive delegation. Team {manager, intern, worker$_1$, worker$_2$}; goal: approve and execute a \$5,000 purchase respecting permissions. Preserve: `worker_permutation`, `paraphrase`; break: `manager_intern_swap`.

5.2 Agents and metrics

We run six agents (Section 4) on `openai/gpt-4o-mini` via OpenRouter, with $r{=}1$ refinement attempts allowed, an orbit cap of 5 per group, and 5 tasks per domain. Trace embeddings h are deterministic hashed bag-of-tokens vectors [Weinberger *et al.*, 2009] in $\mathbb{R}^{256}$, which is sufficient for orbit-level S_G and d_{eff}; we discuss this choice in Section 7. Each metric is averaged over the orbit and over tasks within a domain.

5.3 Implementation notes

All API calls go through a thin OpenRouter wrapper. Orbit re-queries are parallelised with a thread pool of 10. Per-agent JSON files store all per-row metrics; an aggregate script emits the summary table reported below. Total budget: ∼540 chat completions per full run.

6 Results

6.1 Quotient and refinement do not commute

Table 1 contains the central evidence for Hypothesis 1.

Validity asymmetry on action commutation. DIRECT, CHECKER and QUOTIENT all fail one task (`act_commut_003`) for the same reason: the LLM emits `pay_registration_fee` although that action is not in the chosen set for this seed. R→Q repairs the failure in a second attempt and reaches 100% validity. Q→R runs the same number of attempts but remains at 80% — the canonical form passed to its validator collapses the topological sort, leaving the validator with no error to surface and the refinement hint with nothing to repair.

Break-group collapse on role delegation. R→Q reports $\mathrm{NDA}_{\mathrm{break}} = 1.000$ and $\mathrm{OverQ} = 0.000$: the `manager↔intern` swap is detected by CAUSAL-HASH because the raw plan still carries the original actor names. Q→R reports $\mathrm{NDA}_{\mathrm{break}} = 0.000$ and $\mathrm{OverQ} = 1.000$: its output plan is the canonical form `request_budget(INTERN)`, `approve_budget(MANAGER)`, `execute_purchase(WORKER)`, and after a

Table 1: Per-agent / per-domain metrics. Validity is the fraction of DIRECT-style raw plans that pass the symbolic checker; OC is orbit consistency on the preserve channel (higher is better); $\text{NDA}_{\text{break}}$ is necessary-distinction on the break channel (higher is better); OverQ$=1-\text{NDA}_{\text{break}}$ measures break-group collapse; UnderQ$=1-$OC measures preserve-group shattering. Topology shows node / edge Jaccard on the paraphrase orbit (causal Jaccard, when defined, was 1.000 everywhere). d_{eff} is averaged over preserve orbits.

Agent	Domain	Validity	OC	S_G	NDA_{br}	OverQ	UnderQ	d_{eff}	node	edge
DIRECT	object_renaming	1.000	0.833	0.931	—	—	0.167	1.164	0.933	0.800
DIRECT	action_commutation	0.800	0.967	0.869	0.000	1.000	0.033	1.363	0.980	0.490
DIRECT	role_delegation	1.000	1.000	0.869	1.000	0.000	0.000	1.375	1.000	1.000
CHECKER	object_renaming	1.000	0.767	0.931	—	—	0.233	1.161	0.933	0.662
CHECKER	action_commutation	0.800	0.967	0.869	0.000	1.000	0.033	1.363	0.980	0.470
CHECKER	role_delegation	1.000	0.933	0.868	1.000	0.000	0.067	1.377	1.000	0.933
QUOTIENT	object_renaming	1.000	0.800	0.931	—	—	0.200	1.165	0.967	0.775
QUOTIENT	action_commutation	0.800	0.947	0.869	0.000	1.000	0.053	1.364	0.980	0.500
QUOTIENT	role_delegation	1.000	0.967	0.868	1.000	0.000	0.033	1.377	1.000	0.967
Q→R	object_renaming	1.000	0.767	0.934	—	—	0.233	1.174	0.767	0.650
Q→R	action_commutation	**0.800**	0.917	0.869	0.000	1.000	0.083	1.364	0.980	0.940
Q→R	role_delegation	1.000	1.000	0.877	**0.000**	**1.000**	0.000	1.354	1.000	1.000
R→Q	object_renaming	1.000	0.513	0.934	—	—	0.487	1.154	0.833	0.750
R→Q	action_commutation	**1.000**	0.967	0.869	0.000	1.000	0.033	1.365	0.980	0.500
R→Q	role_delegation	1.000	0.933	0.868	**1.000**	**0.000**	0.067	1.377	1.000	0.933

manager↔intern swap the canonical form is identical. Quotienting before validation has destroyed the very distinction the break group was designed to probe.

Preserve-side small penalty for R→Q. On the object_renaming preserve channel, R→Q has lower OC (0.513) than QUOTIENT (0.800). The refinement loop, before canonicalisation, can rewrite the plan in object-name-dependent ways that the final Q no longer fully absorbs. This is the trade-off predicted by Hypothesis 1.

6.2 Effective dimension and Reynolds invariants

Averaged over preserve orbits, d_{eff} ranges from 1.15 (object_renaming) to 1.38 (role_delegation). The paraphrase orbit dominates the variation: per-group d_{eff} on `paraphrase` reaches 1.7, while d_{eff} on the renaming/permutation orbits stays near 1.0. This matches the intuition that re-wording the prompt expands the representational range of the trace far more than swapping a single name.

6.3 Workflow-topology Jaccard

On paraphrase orbits, node similarity is consistently high (0.77–1.00): the agent picks essentially the same action set across paraphrases. Edge similarity (the order of actions) drops to 0.42–1.00 and is the metric that separates agents. The causal Jaccard, reported on action_commutation, is 1.000 for every agent: although surface order fluctuates, every agent always respects the precondition graph. This decomposes paraphrase robustness into "agree on what to do" (high), "agree on order" (variable), and "agree on the causal skeleton" (universal in our tasks).

6.4 Residual over-quotienting on action commutation

A persistent OverQ $= 1.000$ on `action_commutation` appears for every agent in our table. Investigation traces it to the test design rather than to canonicalisation: the `action_commutation_dependent` orbit rewrites the plan field but leaves the prompt unchanged, so the agent re-emits the same plan and CAUSAL-HASH is constant across the orbit. A prompt-side rewrite of the dependency rules (analogous to the manager↔intern swap in roles) is needed to surface the LLM's own break-group sensitivity, and we leave this extension to future work.

6.5 Agentic CEGAR baselines

To isolate the contribution of *counterexample-guided* refinement we add two baselines: RANDOM-R, which retries without any symbolic hint, and ORACLE-R, which replaces an invalid plan with the ground-truth `valid_plan`. On the `action_commutation` domain the eight agents resolve into a crisp four-tier ladder, summarised in Table 2. Agents that perform no refinement (DIRECT, CHECKER, QUOTIENT) all fail the same task (`act_commut_003`) because the LLM emits an action that is not in the chosen set. Random retry (RANDOM-R) consumes a second attempt but cannot recover. Q→R also consumes a second attempt and cannot recover, because canonicalisation has discarded the information the validator would have used to flag the violation. Only R→Q and B^2WM (its alias) recover, and they do so on the second attempt — exactly matching the ORACLE-R upper bound. Counterexample-guided refinement is therefore both *necessary* (without it the LLM cannot self-correct) and *sufficient* (its success rate matches the oracle's).

Formally, B^2WM is an instance of CEGAR over an LLM-induced world model: the symbolic checker plays the role of the model checker, the validator's error err plays the role of a counterexample c, and the repair hint extracts the missing distinction $\Delta(c)$ (e.g. "`pay_*` requires `confirm_budget`"). The refined world is $R_c(W) = W \cup \Delta(c)$. The phenomenon $Q \circ R \neq R \circ Q$ corresponds to the familiar fact in classical

Table 2: Eight agents on `action_commutation`: the failure task `act_commut_003` separates blind retry from counterexample-guided refinement. B^2WM matches the oracle upper bound.

Agent	Validity	003 valid?	Attempts
DIRECT	0.80	no	1
CHECKER	0.80	no	1
QUOTIENT	0.80	no	1
RANDOM-R	0.80	no	2
Q→R	0.80	no	2
R→Q	**1.00**	**yes**	2
B^2WM	**1.00**	**yes**	2
ORACLE-R	**1.00**	**yes**	2

CEGAR that an abstraction step performed before counterexample extraction can mask the very witness the refinement step requires.

6.6 Quotient × Refinement phase diagram

We further map a 4×4 grid of pipeline configurations on `action_commutation`, varying the quotient strength (Q_0= none, Q_1= light, Q_2= medium, Q_3= full topological canonical) and the refinement strength (R_0= none, R_1= blind retry, R_2= CEG with one attempt, R_3= CEG with three attempts). Validity is 1.00 everywhere on this 3-task slice, but the three-way orbit consistency (Section 3.6) varies: $\mathrm{OC}_{\mathrm{causal}}$ and $\mathrm{OC}_{\mathrm{topo}}$ are pinned at 1.00 for every cell, while $\mathrm{OC}_{\mathrm{seq}} \in [0.56, 0.67]$ depending on (Q, R). In other words, every cell agrees on the causal skeleton (the dependency graph is always respected) and on the topological canonical (free-order actions are always sorted the same way) but disagrees on the surface order under paraphrase — a clean empirical signature of partial-order equivalence. Larger gaps along the V axis require a wider task pool that includes counterexample-only tasks like `act_commut_003`; we report this version of the grid here as a methodological proof of concept for the phase-diagram readout.

7 Discussion

The order-asymmetry phenomenon is not specific to our domains: it is a structural property of any pipeline that performs symbolic abstraction and counterexample-driven repair. Whenever quotienting maps a class of inputs to a single representative, any error witnessed only in the discarded coordinate becomes invisible to the subsequent validator. The take-away is concrete: when designing neuro-symbolic agents, place the refinement loop *upstream* of any canonicalisation that abstracts over the dimensions the validator can detect.

The dual phenomenon — $R \to Q$ inheriting a small under-quotienting penalty on paraphrase orbits — has a similarly clean explanation: once refinement has rewritten the plan in a paraphrase-dependent way, Q alone cannot fully erase that dependence. The right fix is not a different order but a stronger preserve-channel canonical that absorbs more of the variation, for instance by recanonicalising through workflow-topology equivalence rather than topological sort alone.

Hash-embedding limitation. Our S_G and d_{eff} readouts use a token-hash embedding. This is sufficient for the qualitative differences reported in Table 1 because the embedding is content-sensitive and deterministic, but a contextual sentence embedding would yield sharper S_G values and is a drop-in replacement.

Synthetic domains, single base model. Domain coverage (BlocksWorld, conference trip, role delegation) is deliberately controlled so that ground-truth equivalence classes can be enumerated. Scaling to ALFWorld [Shridhar *et al.*, 2021], WebShop [Yao *et al.*, 2022], ScienceWorld [Wang *et al.*, 2022], and Plancraft is left for future work, as is a sweep over weaker base models that we expect to amplify the gap between DIRECT and the B^2WM variants.

Connections to algebraic geometry. The blowup/blowdown vocabulary is more than metaphor in our setting: quotienting acts on the trace-embedding space as a contraction along the orbit direction, and refinement introduces a new symbolic coordinate that resolves an ambiguity, exactly the formal role of an exceptional divisor [Hironaka, 1964; Hartshorne, 1977; Sturmfels, 1993]. A natural next step is to track the local geometry of the trace manifold through Q and R steps and to relate d_{eff} to the dimension of the exceptional locus.

8 Conclusion

We have argued that LLM agent failures are often best framed as symbolic symmetry errors and that the natural neuro-symbolic response, combining quotient Q and refinement R, is sensitive to the order in which the two operations are applied. Empirically, $R \circ Q$ dominates $Q \circ R$ on both validity and break-group preservation, while a small paraphrase-side under-quotienting penalty appears as the corresponding cost. Together with a two-track plan canonicalisation and a workflow-topology readout, this yields a compact, measurable picture of how an LLM agent's world model behaves under symbolic symmetry, and turns the "greedy blowdown" intuition into an ordering principle: refine first, then quotient.

References

[Bronstein *et al.*, 2021] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[Clarke *et al.*, 2003] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM*, 50(5):752–794, 2003.

[Cohen and Welling, 2016] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *Proc. ICML*, pages 2990–2999, 2016.

[Domshlak *et al.*, 2012] Carmel Domshlak, Michael Katz, and Alexander Shleyfman. Enhanced symmetry breaking in cost-optimal planning as forward search. In *Proc. ICAPS*, 2012.

[Fox and Long, 1999] Maria Fox and Derek Long. The detection and exploitation of symmetry in planning problems. In *Proc. IJCAI*, pages 956–961, 1999.

[Gao *et al.*, 2017] Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, 2017.

[Hartshorne, 1977] Robin Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer, 1977.

[Henzinger *et al.*, 2002] Thomas A. Henzinger, Ranjit Jhala, Rupak Majumdar, and Grégoire Sutre. Lazy abstraction. In *Proc. POPL*, pages 58–70, 2002.

[Hironaka, 1964] Heisuke Hironaka. Resolution of singularities of an algebraic variety over a field of characteristic zero. *Annals of Mathematics*, 79(1):109–326, 1964.

[Huang *et al.*, 2022a] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: extracting actionable knowledge for embodied agents. In *Proc. ICML*, 2022.

[Huang *et al.*, 2022b] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: embodied reasoning through planning with language models. In *Proc. CoRL*, 2022.

[Kondor and Trivedi, 2018] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *Proc. ICML*, pages 2747–2755, 2018.

[Liu *et al.*, 2023] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. LLM+P: empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

[McAllester and Rosenblitt, 1991] David A. McAllester and David Rosenblitt. Systematic nonlinear planning. In *Proc. AAAI*, pages 634–639, 1991.

[Mishra *et al.*, 2022] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proc. ACL*, 2022.

[Pochter *et al.*, 2011] Nir Pochter, Aviv Zohar, and Jeffrey S. Rosenschein. Exploiting problem symmetries in state-based planners. In *Proc. AAAI*, pages 1004–1009, 2011.

[Recanatesi *et al.*, 2019] Stefano Recanatesi, Matthew Farrell, Madhu Advani, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Dimensionality compression and expansion in deep neural networks. *arXiv preprint arXiv:1906.00443*, 2019.

[Schick *et al.*, 2023] Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: language models can teach themselves to use tools. In *Advances in NeurIPS*, 2023.

[Sclar *et al.*, 2024] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design. In *Proc. ICLR*, 2024.

[Shridhar *et al.*, 2021] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: aligning text and embodied environments for interactive learning. In *Proc. ICLR*, 2021.

[Silver *et al.*, 2024] Tom Silver, Soham Dan, Kavitha Srinivas, Joshua B. Tenenbaum, Leslie Pack Kaelbling, and Michael Katz. Generalized planning in PDDL domains with pretrained large language models. In *Proc. AAAI*, 2024.

[Singh *et al.*, 2023] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: generating situated robot task plans using large language models. In *Proc. ICRA*, 2023.

[Slaney and Thiébaux, 2001] John Slaney and Sylvie Thiébaux. Blocks World revisited. *Artificial Intelligence*, 125(1–2):119–153, 2001.

[Sturmfels, 1993] Bernd Sturmfels. *Algorithms in Invariant Theory*. Springer, 1993.

[Valmeekam *et al.*, 2023] Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: an extensible benchmark for evaluating large language models on planning and reasoning about change. In *Advances in NeurIPS*, 2023.

[Wang *et al.*, 2022] Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: is your agent smarter than a 5th grader? In *Proc. EMNLP*, 2022.

[Wang *et al.*, 2023] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: an open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[Weinberger *et al.*, 2009] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proc. ICML*, pages 1113–1120, 2009.

[Weld, 1994] Daniel S. Weld. An introduction to least-commitment planning. *AI Magazine*, 15(4):27–61, 1994.

[Yao *et al.*, 2022] Shunyu Yao, Howard Chen, John Yang, and Karthik R. Narasimhan. WebShop: towards scalable real-world web interaction with grounded language agents. In *Advances in NeurIPS*, 2022.

[Yao *et al.*, 2023] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: synergizing reasoning and acting in language models. In *Proc. ICLR*, 2023.