A Gauge-Inspired Architecture for Representation-Robust Ethical Evaluation

Andrew H. Bond
San José State University
andrew.bond@sjsu.edu

Abstract

AI systems that make evaluative judgments can be exploited through *adversarial redescription*: presenting semantically equivalent situations in different forms to induce different outcomes. We propose a layered framework that (i) states representation invariance as a formal requirement, (ii) reduces this requirement to the existence and quality of a *canonicalizer*, and (iii) introduces an operational "curvature" metric that measures redescription inconsistency via loop tests. The main engineering contribution is a canonicalization pathway that targets a discrete modeling language (ErisML) rather than a continuous embedding space. Grammar-based parsing yields a deterministic accept/reject boundary and a unique abstract syntax tree (AST) representation when parsing succeeds; this can reduce a class of "in-between" boundary attacks common to vector quantization. However, it does *not* guarantee semantic equivalence: residual failure modes include errors in the natural-language-to-ErisML transpiler, ontology gaps, and mis-specified norms. We provide a reviewer-oriented evaluation plan: a metamorphic/loop-testing benchmark suite, audit artifacts, baselines, and measurable metrics (invariance violation rates, operational curvature distributions, and veto coverage).

1 Introduction

1.1 Problem statement

Modern AI systems increasingly perform evaluative tasks (moderation, triage, resource allocation, autonomy). A persistent vulnerability is that the same underlying situation can be described in multiple ways, and naive systems can produce different evaluations for descriptions that are intended to be meaning-preserving. This is often discussed as specification gaming or reward hacking, but it can be stated more precisely: *the evaluation function should be invariant under meaning-preserving redescription*.

1.2 Design goal

We seek a constructive path from an invariance requirement to a system design that is (i) *auditable* (explicit intermediate representations), (ii) *measurable* (a protocol for inconsistency), and (iii) *conservative by default* (refuse to guess when the representation is underdetermined).

1.3 High-level approach

We use a gauge-inspired viewpoint: in physics, observables are invariant under certain representation changes (gauge transformations). Here, the "transformations" are semantic redescriptions. Rather than importing the full differential-geometric machinery, we adopt its practical lesson: *when invariance fails, quantify the failure and localize it to a concrete component*. In our architecture, the key locus is the canonicalizer.

2 Layer 0: A Minimal Tensor Foundation

We include a compact foundation to support coordinate-free expressions, while emphasizing that moral *content* remains an input.

Definition 1 (Ethical feature space)**.** *Let V be a finite-dimensional real vector space whose coordinates correspond to morally relevant features chosen by the designer or domain experts. Let $g : V \times V \to \mathbb{R}$ be a positive-definite symmetric bilinear form.*

Definition 2 (Intention, obligation, and raw judgment)**.** *Let $I \in V$ denote an* intention *vector and $O \in V$ an* obligation *vector. Define the (normalized) raw judgment*

$$\Sigma \;=\; \frac{g(I,O)}{\|O\|_g}. \tag{1}$$

Theorem 1 (Coordinate independence)**.** *Σ is invariant under a change of basis of V.*

Remark 1. *This layer is deliberately minimal. It supports later invariance requirements but does not resolve which features belong in V or how I and O are obtained.*

3 Layer 1: Representation Invariance

3.1 Description space and redescriptions

Let X denote the set of possible descriptions of situations (e.g., text prompts, structured reports, policy statements).

Definition 3 (Redescription family)**.** *Let $\mathcal{G}$ be a family of transformations acting on X intended to preserve meaning (synonyms, paraphrases, perspective shifts, euphemism expansion, etc.). We write $g \cdot x$ for applying $g \in \mathcal{G}$ to $x \in X$.*

Definition 4 (Bond Invariance Principle (BIP))**.** *An evaluation function $\Sigma : X \to \mathbb{R}$ satisfies BIP w.r.t. $\mathcal{G}$ if*

$$\forall g \in \mathcal{G},\ \forall x \in X :\ \ \Sigma(g \cdot x) = \Sigma(x). \tag{2}$$

3.2 Canonicalization

A standard way to enforce invariance is to factor evaluation through a canonical form.

Definition 5 (Canonicalizer)**.** *A map $\kappa : X \to Z$ (into some representation set Z) is a canonicalizer for $\mathcal{G}$ if:*

1. ***Idempotence:*** *$\kappa(\kappa(x)) = \kappa(x)$ for all x (well-defined normal form).*
2. ***Orbit collapse:*** *if $y = g \cdot x$ for some $g \in \mathcal{G}$, then $\kappa(x) = \kappa(y)$.*

Theorem 2 (Canonicalization suffices)**.** *If κ is a canonicalizer and $\Sigma_0 : Z \to \mathbb{R}$ is any function, then $\Sigma = \Sigma_0 \circ \kappa$ satisfies BIP.*

Remark 2. *In practice, $\mathcal{G}$ is not fully known or enumerable. Thus, "BIP holds" should be read as "BIP holds for a specified suite of redescriptions". The architecture below is intended to make violations measurable and correctable.*

4 Layer 2: Operational Curvature (Path Dependence) as a Test Target

Gauge theory motivates the idea that non-invariance can be exposed as path dependence. We adopt an *operational* definition that does not require differential forms.

4.1 Loop tests

Given two transformations $g_1, g_2 \in \mathcal{G}$, compare the two length-2 paths around the "loop":

$$x \xrightarrow{g_1} g_1 \cdot x \xrightarrow{g_2} g_2 g_1 \cdot x \qquad \text{vs.} \qquad x \xrightarrow{g_2} g_2 \cdot x \xrightarrow{g_1} g_1 g_2 \cdot x.$$

If canonicalization is consistent, both routes yield the same normal form.

Definition 6 (Operational curvature)**.** *Let $\Delta(\cdot,\cdot)$ be a distance on Z (e.g., Hamming distance between normalized AST fields). Define*

$$\Omega_{\mathrm{op}}(x; g_1, g_2) \;=\; \Delta\big(\kappa(g_2 g_1 \cdot x),\ \kappa(g_1 g_2 \cdot x)\big). \tag{3}$$

We call Ω_{op} the operational curvature *at x for (g_1, g_2).*

Proposition 1 (What zero operational curvature buys)**.** *Assume κ is deterministic. If $\Omega_{\mathrm{op}}(x; g_1, g_2) = 0$ for all (g_1, g_2) in a test family and all x in a test set, then κ is path-independent on those tested loops. Consequently, $\Sigma_0 \circ \kappa$ is invariant on the same tested family.*

Remark 3. *Operational curvature is intentionally test-first: it measures inconsistency even when a full gauge-theoretic structure is not (yet) formalized. If a future version provides a rigorous bundle construction and a correspondence between canonicalizers and connections, then Ω_{op} can be related to holonomy/curvature in the usual sense.*

5 Layer 3: Discrete Canonicalization via ErisML

5.1 Motivation: continuous vs. discrete normal forms

A common canonicalization approach embeds text into $\mathbb{R}^n$ and then clusters or thresholds. These pipelines often have fuzzy decision boundaries: small changes in wording can cross a boundary and change the assigned cluster. Such boundary regions are natural targets for adversarial redescription.

We propose a different target: canonicalize into a *discrete* modeling language with a grammar and verifier. In that setting, there is a hard accept/reject boundary, and successful parses yield a structured representation that can be normalized deterministically.

5.2 Two-stage canonicalizer

We define a canonicalizer κ as:

$$\kappa(x) = \begin{cases} \mathrm{normalize}(\mathrm{parse}(\mathrm{transpile}(x))) & \text{if parsing and validation succeed} \\ \bot & \text{otherwise (veto)} \end{cases} \tag{4}$$

where:

- **transpile** is an LLM-assisted translation from natural language to ErisML code,
- **parse** produces an AST if the code is syntactically valid,
- **validate** checks constraints (active norms, required fields, safety invariants),
- **normalize** produces a unique canonical form (e.g., sorted fields, resolved defaults, deterministic hashing).

5.3 Conservatism: veto instead of guessing

When information is missing, ambiguous, or fails validation, the system returns $\bot$ rather than an arbitrary best-effort judgment. This shifts a large class of "silent" failures into explicit refusal modes, improving auditability and enabling downstream escalation policies (human review, request for clarification, or additional sensing).

5.4 What discreteness does and does not guarantee

Discreteness helps in two ways:

1. **No in-between states.** There is no continuous interpolation between two valid ASTs.
2. **Deterministic normalization.** If parsing succeeds, normalization can be made deterministic, eliminating randomness as a source of inconsistency.

However, discreteness does *not* solve semantic equivalence by itself. If the transpiler maps two equivalent descriptions to different ASTs, loop inconsistency remains; the point is that this inconsistency is measurable and attributable to a concrete component (transpiler, ontology, or norms).

6 Layer 4: Measurement and Audit Artifacts

6.1 Bond (Bd) as a normalization convention

To support regression tracking, deployments can normalize Ω_{op} into a single scalar. A conservative default is to set loop "area" $A \equiv 1$ and use a fixed threshold τ.

Definition 7 (Bond (Bd))**.** *Fix a choice of Δ and a detection threshold $\tau > 0$. Define the Bond as:*

$$\|\Omega\|_{\mathrm{Bd}} \;=\; \frac{\Omega_{\mathrm{op}}}{\tau}. \tag{5}$$

Remark 4. *If a principled "area" A is later defined (e.g., via a metric over transformation complexity), one may instead use $\Omega_{\mathrm{op}}/(A\tau)$. For review, we recommend reporting Ω_{op} directly (interpretable) and using Bonds primarily for trend plots across versions.*

6.2 Audit artifacts

Each evaluation run should emit a machine-checkable artifact with: (i) the baseline description x and canonical form $\kappa(x)$, (ii) the applied transforms and their intended class (bond-preserving vs. bond-changing), (iii) per-transform verdicts and invariance checks, (iv) loop-test results and summary statistics, and (v) a hash/signature for tamper-evidence. This mirrors metamorphic testing practice: the transforms serve as metamorphic relations and the artifact is the evidence package.

7 Evaluation Plan (Reviewer-Oriented)

This section makes falsifiability and measurement explicit.

7.1 Threat model

We assume an adversary can choose or perturb the input description while leaving the underlying situation unchanged, aiming to (a) flip the verdict, (b) force a favorable canonicalization, or (c) induce a non-auditable failure. We do not assume the adversary can directly tamper with the parser/verifier or the audit log; if such tampering is possible, additional security controls are required (out of scope).

7.2 Benchmarks

We propose three benchmark families:

1. **Structured-choice scenarios:** short cases with multiple options (e.g., triage/resource allocation) where option order, labels, and units can be perturbed without changing meaning.
2. **Narrative dilemmas:** longer textual dilemmas (e.g., tragic-conflict templates) where paraphrase and perspective shifts are common.
3. **Policy-compliance cases:** descriptions mapped to policy constraints; bond-preserving transforms include paraphrase and formatting changes; bond-changing transforms remove or add key policy-relevant facts.

7.3 Transform suites

For each benchmark case, generate:

- **Bond-preserving suite** $\mathcal{G}_{\mathrm{bp}}$: paraphrases; synonym swaps; order/label permutations; unit conversions; insertion of irrelevant but explicitly marked context; template-style rephrasings.
- **Bond-changing suite** $\mathcal{G}_{\mathrm{bc}}$: controlled edits that remove/add a key ethical fact (e.g., consent, vulnerability, discrimination evidence), intended to *change* the verdict.

7.4 Baselines

Compare the proposed discrete canonicalizer to:

1. **Direct judge baseline:** a single-step model that outputs a verdict from text (no explicit canonical form).
2. **Embedding canonicalizer baseline:** text embedding + clustering/nearest-prototype canonicalization, followed by the same downstream evaluator.
3. **Rules-only baseline:** a hand-authored ruleset over extracted keywords/features (where feasible).

7.5 Metrics

Report:

- **Invariance violation rate:** fraction of $g \in \mathcal{G}_{\mathrm{bp}}$ where $\Sigma(g \cdot x) \neq \Sigma(x)$ (or where canonical forms differ, depending on the pipeline).
- **Operational curvature distribution:** mean/median/95th of $\Omega_{\mathrm{op}}(x; g_1, g_2)$ over sampled loops, plus worst-case witness loops.
- **Veto coverage:** fraction of cases resulting in $\perp$ (overall and stratified by transform type).
- **Sensitivity:** detection rate for $\mathcal{G}_{\mathrm{bc}}$ edits (i.e., verdict should change when the bond changes).
- **Cost and latency:** runtime and compute per case (important for deployment viability).

7.6 Acceptance criteria (suggested)

A conservative target for a first submission is: low invariance violations on $\mathcal{G}_{\mathrm{bp}}$ (relative to baselines), transparent witnesses for remaining violations, and bounded veto rates (veto is acceptable if explained and recoverable, but should not dominate).

8 Related Work

Invariance in machine learning. Invariance and equivariance are used as inductive biases (e.g., group-equivariant networks) and as criteria for OOD generalization (e.g., invariant risk minimization). Our use differs: invariance is treated as a *governance contract* with audit artifacts, not primarily as a learning objective.

Adversarial examples and prompt attacks. Adversarial perturbations expose brittle decision boundaries in continuous spaces. Our proposal does not prevent adversarial inputs, but aims to reduce a class of boundary-sensitive failures by canonicalizing to a discrete syntax and rejecting unparseable cases.

Specification gaming and reward hacking. Specification gaming highlights how agents satisfy literal objectives while violating intent. BIP-style audits are a complementary tool: they target representational loopholes where intent is held fixed but the description is manipulated.

Metamorphic testing. Metamorphic testing addresses the oracle problem by asserting relations between multiple inputs and their outputs. BIP can be seen as a family of metamorphic relations, and loop tests as higher-order metamorphic checks for path dependence.

Risk frameworks and governance. NIST and related governance frameworks emphasize documentation, transparency, and measurable risk controls. Audit artifacts and invariance suites provide an actionable mechanism to instantiate these controls for representation robustness.

9 Limitations and Open Problems

9.1 Key limitations

- **Semantic ground truth.** "Meaning-preserving" redescriptions are only approximated via test suites; the architecture does not solve semantic equivalence in the abstract.
- **Ontology coverage.** The target language must represent morally relevant distinctions; otherwise the system vetoes or collapses distinct cases.
- **Transpiler robustness.** The LLM transpiler is a learned component and can be attacked; its errors are a dominant residual risk.
- **Security assumptions.** If the parser/verifier/audit log can be tampered with, additional system security is required beyond this paper.

9.2 Open problems (research + engineering)

1. **Bundle formalization.** Give a rigorous mathematical structure for description spaces and redescription families appropriate to deployed systems.
2. **Canonicalizer–connection correspondence.** Determine conditions under which a canonicalizer induces (or is equivalent to) a connection in a principal bundle.
3. **Metrics.** Develop principled choices for Δ and loop sampling that correlate with exploitability in realistic adversarial settings.
4. **Empirical evaluation.** Build benchmarks with adversarial paraphrase generators; report curvature distributions and veto rates.
5. **Norm specification.** Establish methods for specifying and verifying norms (including conflict detection and provenance).

10 Conclusion

We presented a conservative, test-oriented architecture for representation-robust ethical evaluation. The central claim is not that grammar-based canonicalization "solves ethics," but that it offers a practical way to *reduce and measure* redescription vulnerabilities by forcing intermediate representations into a discrete, auditable form and making failure explicit via vetoes. The next step is empirical: implement the evaluation plan and report whether operational curvature and invariance violations decrease under adversarial redescription relative to baselines.

Epistemic status. This paper proposes a mathematical framing and an implementable system design. Definitions are stipulative. Results labeled Theorem are proved within the stated assumptions (primarily determinism and idempotence of canonicalization). Gauge-theoretic objects (connections/curvature forms) are used as motivation unless explicitly derived. Empirical claims are deferred to evaluation; the system is designed to make such evaluation straightforward.

A Appendix: Anticipated Questions (Q&A)

This appendix is written in a deliberately practical tone. It is intended to make the paper easy to review, falsify, and reproduce.

Scope and claims

Q1. What is the main claim of this paper? **A.** The main claim is *procedural*: if a system asserts that certain transformations are "bond-preserving" (i.e., should not change the ethically relevant structure), then it should be possible to *test* that claim via invariance audits and loop tests, producing machine-checkable artifacts (witnesses on failure). We do *not* claim to solve normative ethics, semantic equivalence in general, or value learning.

Q2. Is this a normative ethical theory? **A.** No. The paper is an *assurance method* and a *systems architecture*. It makes normative commitments *explicit* (via profiles, norms, vetoes) so they can be audited and contested, but it does not argue that any particular set of commitments is uniquely correct.

Q3. Are you claiming that invariance implies moral truth? **A.** No. Invariance is a *consistency constraint*: it rules out a class of failures (frame dependence, re-description gaming) within a declared scope. A system can be invariant and still be morally wrong if its norms, ontology, or measurement choices are wrong.

Q4. What does "bond" mean operationally? **A.** A "bond" is whatever the implementation treats as ethically salient structure (e.g., consent relations, harm attribution, protected-attribute discrimination flags). "Bond-preserving" transformations are those intended *not* to change that structure. This is intentionally operational: bonds are defined by the canonical representation and its validators.

Formalism, definitions, and what is actually proved

Q5. What is G? Do you require a mathematical group? **A.** In the cleanest exposition G is a group action on descriptions. In practice, the implemented "redescription set" may be a *monoid* (not all transforms invertible) or a *family* of metamorphic relations. The invariance condition generalizes: for each tested transform t declared bond-preserving, require $\Sigma(t(x)) = \Sigma(x)$. The group language is used for clarity, not as a hard requirement.

Q6. What is actually proved versus motivated by physics? **A.** Theorems are limited to standard results: (i) invariance under a declared transform family follows from factoring through a canonicalizer that collapses declared orbits; (ii) tested-loop path-independence follows when operational loop inconsistency is zero on the tested set. Claims about differential-geometric curvature, connections, or Noether-style conservation are treated as *motivation* unless a rigorous construction is provided.

Q7. Isn't this just quotienting / canonicalization? What's new? **A.** The mathematical core is indeed quotienting. The paper's contribution is engineering: (i) a conservative canonicalization pipeline that targets a discrete language (ErisML) with parse/validate/veto behavior, (ii) an operational "curvature" diagnostic (loop tests) and a unit/metric for tracking redescription inconsistency across versions, and (iii) machine-checkable audit artifacts suitable for CI and third-party verification.

Q8. How do you define "meaning-preserving"? **A.** We do *not* claim a universal semantic definition. Instead, the deployment defines a *test suite* of transformations considered bond-preserving under the system's measurement model (e.g., reorder, relabel, unit conversion, bounded paraphrase). The claim tested is: "within this declared scope, outcomes are invariant." This makes the semantics question explicit and falsifiable.

Implementation: ErisML canonicalization, transpilers, and vetoes

Q9. Why use a discrete language at all? **A.** Because discrete parsing provides (i) a hard accept/reject boundary, (ii) structured intermediate representations (ASTs) for audit, and (iii) deterministic normalization. This removes a class of "fuzzy boundary" attacks where tiny wording changes cross continuous embedding thresholds. It does not solve semantics by itself; it makes failures measurable and attributable.

Q10. Doesn't the LLM transpiler become the weakest link? **A.** Yes, and we treat it as such. The architecture is designed so that (i) transpiler outputs are *untrusted* until parse+validation succeeds, (ii) failures produce explicit veto/refusal rather than silent guesses, and (iii) loop tests and adversarial re-descriptions directly measure residual inconsistency. The intended security posture is "fail closed".

Q11. What prevents bypassing canonicalization? **A.** The deployment model includes an external gate/monitor: evaluation and enforcement are conditioned on passing through the canonicalization+validation pipeline. If a system can route around the gate, invariance claims are void. This is a standard assurance assumption and should be made explicit in any deployment.

Q12. What does the veto mean in a product setting? **A.** Veto is a controlled refusal mode: "insufficiently grounded or invalid representation; escalate for human review or request additional information." Veto rate is itself a metric (coverage) and should be tracked. High veto rate can indicate ontology gaps, weak transpilation, or overly strict validators.

Operational curvature, metrics, and evaluation

Q13. What exactly is "operational curvature" and why call it curvature? **A.** Operational curvature is a *path-dependence signal*: if two short redescription paths around a loop yield different canonical forms, the system's normalization is inconsistent under the tested transformations. "Curvature" is an analogy to holonomy in gauge theory, but the paper's deliverable is the testable quantity (distance between canonical forms), not a differential form.

Q14. Your "Bond" unit seems arbitrary. Do we need it? **A.** No. The minimal useful metric is simply loop inconsistency Ω_{op} on a fixed suite. The "Bond" is an optional normalization to compare across systems/versions when a stable convention for thresholds and (optionally) loop "area" is defined. A conservative default is to set area $A \equiv 1$.

Q15. How do you choose the distance Δ between canonical forms? **A.** Choose Δ to reflect ethically salient structure: e.g., Hamming distance over normalized AST fields, edit distance over canonical JSON, or weighted differences emphasizing veto flags, consent relations, protected-attribute discrimination indicators, and harm attributions. The distance function is part of the evaluation protocol and should be reported.

Q16. What benchmarks do you propose? **A.** We propose three benchmark families: (i) *metamorphic invariance* suites (reorder/relabel/unit conversion/bounded paraphrase), (ii) *adversarial redescription* suites (euphemism expansion, obfuscation, perspective shifts, "legalese" reframing), and (iii) *counterfactual bond-changes* (e.g., removing discrimination evidence) where outcome changes are expected. Report (a) invariance pass rate, (b) loop inconsistency distribution, (c) veto/coverage rate, and (d) failure witnesses.

Q17. What baselines should be compared? **A.** A natural baseline is a continuous embedding canonicalizer (cluster/threshold) feeding the same downstream evaluator, plus a "no canonicalizer" baseline. If possible, also compare a hand-coded structured parser without LLM transpilation. The goal is not perfect semantics, but relative reduction in redescription sensitivity and clearer failure localization.

Governance, stakeholders, and normative pluralism

Q18. Who chooses the norms and stakeholder profiles? **A.** The deployment does. The paper's stance is governance-first: norms, vetoes, and weights must be explicit, versioned, and auditable. The architecture supports multi-stakeholder aggregation, but it does not claim a universal resolution to moral disagreement.

Q19. Isn't "multi-stakeholder aggregation" itself ethically contentious? **A.** Yes. The point is not to endorse a particular aggregation method, but to make the method inspectable and testable. In many high-stakes settings, hard constraints (rights, consent) should dominate; the architecture supports lexical ordering and vetoes precisely because aggregation-by-sum is often unacceptable.

Q20. How do you prevent "ethics washing" (pretty logs, bad norms)? **A.** You cannot prevent it by math alone. What you can do is (i) require explicit commitments (so they can be criticized), (ii) require invariance within the declared commitment scope (so outcomes cannot be quietly reframed), and (iii) produce audit artifacts for external review. Legitimacy is a governance process problem.

Security, failure modes, and threat model

Q21. What attacks remain even with discrete canonicalization? **A.** Key residual risks include: (i) transpiler manipulation (prompt injection, jailbreaks), (ii) ontology gaps (unrepresentable distinctions), (iii) validator bugs or incomplete constraints, (iv) measurement tampering (feeding false evidence), and (v) out-of-band channels (actions not captured by the observable interface). The architecture is intended to localize failures, not claim impossibility.

Q22. Can an attacker force vetoes and cause denial-of-service? **A.** Yes. Any fail-closed system can be DoS'd by inputs that trigger refusal. This should be handled at the product/security layer (rate limits, fallbacks, human escalation policies). Veto is safer than silent misclassification, but it must be operationally managed.

Q23. What about probabilistic or nondeterministic components? **A.** Nondeterminism can masquerade as redescription sensitivity. For evaluation, run with fixed seeds and deterministic settings, and treat nondeterminism as a separate axis of instability. In deployment, if nondeterminism is unavoidable, invariance should be stated probabilistically (e.g., invariance holds with high probability under a distribution of internal randomness).

Reproducibility, artifacts, and what reviewers can check

Q24. What can a third party verify without trusting you? **A.** Given the code, transform suite, and audit artifacts, a third party can re-run the invariance and loop tests, confirm pass/fail outcomes, and inspect concrete counterexamples (witnesses) where invariance fails. If artifacts are cryptographically signed (recommended), a third party can also verify provenance and integrity of reported results.

Q25. What would falsify the approach? **A.** Several outcomes would be damaging: (i) discrete canonicalization does not reduce redescription sensitivity compared to embedding baselines; (ii) loop tests do not correlate with exploitability (e.g., high Ω_{op} but no usable attacks, or vice versa); (iii) veto rates are so high that the approach becomes impractical; or (iv) attackers routinely bypass the gate or manipulate the transpiler while still producing valid ASTs.

Q26. What is the minimal "takeaway" if one dislikes the gauge framing? **A.** Ignore the physics language. The core takeaway is: *treat governance as a software contract*. Declare a class of transformations that should not change outcomes, then ship tests and artifacts that prove (or disprove) you are meeting that contract.

References

[1] E. Noether. Invariante Variationsprobleme. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1918, pp. 235–257.

[2] T. S. Cohen and M. Welling. Group equivariant convolutional networks. In *Proc. ICML*, 2016.

[3] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv:1907.02893*, 2019.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. ICLR*, 2014.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc. ICLR*, 2015.

[6] V. Krakovna et al. Specification gaming: the flip side of AI ingenuity. DeepMind Blog, 2020. https://deepmind.google/blog/specification-gaming-the-flip-side-of-ai-ingenuity/

[7] S. Segura, G. Fraser, A. B. Sánchez, and A. Ruiz-Cortés. A survey on metamorphic testing. *IEEE Trans. Software Engineering*, 2016.

[8] National Institute of Standards and Technology (NIST). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1, 2023.

[9] A.H. Bond. ErisML/DEME reference implementation. Repository: https://github.com/ahb-sjsu/erisml-lib.