On-Manifold Activation Patching: Well-Defined Counterfactuals and Stable Causal Effects in Neural Networks

Liz Lemma Future Detective

January 18, 2026

Abstract

Activation patching and ablation are core tools in mechanistic interpretability, but recent work highlights that their conclusions can depend strongly on the intervention implementation and can trigger hydra/backup behavior or interpretability illusions. We argue the root cause is semantic: standard patching produces internal states that the model would never generate, so the implied counterfactual is undefined. We introduce on-manifold patching, a counterfactual semantics that restricts patched activations to lie on the model's activation manifold, formalized as the conditional distribution of internal states induced by the model and its input distribution. Concretely, we learn conditional generative models of activations and replace patch targets with conditional samples consistent with the original context and the patched coordinates. We prove upper bounds showing that if the learned conditional generator approximates the true activation conditional to total-variation (or Wasserstein) error ε, then the induced causal effect estimates are stable up to $O(\varepsilon)$. We also show a lower bound: without an on-manifold constraint (or equivalently, without specifying a counterfactual distribution), causal effects from patching are not identifiable—two patch operators can disagree arbitrarily. Empirically, on-manifold patching reduces intervention-implementation variance and improves circuit localization faithfulness on models with ground-truth mechanisms (compiled transformers) and on standard mechanistic interpretability case studies. The result provides a principled foundation for causal interventions and a practical drop-in primitive for circuit discovery pipelines.

Table of Contents

in perturbation methods; conditional generative modeling of activations.

3. 3. Problem setup: internal counterfactuals for deterministic networks; defining contexts, layers/modules, outcomes; desiderata (well-defined, stable, estimable).

4. 4. On-manifold patching semantics: defining the activation manifold via $p_\ell(a_\ell \mid c)$; defining constrained conditionals $p_\ell(\cdot \mid c, a_\ell[S] = v)$; defining do_{OM} interventions; relation to SCM/causal graphs.

5. 5. Algorithm: learning conditional activation generators; masked conditioning for arbitrary coordinate subsets; performing on-manifold interventions efficiently; practical design choices (which modules, which state types).

6. 6. Theory I (upper bounds): stability of estimated effects under generator approximation error; bounds in TV and Wasserstein; when effect function is Lipschitz/bounded; sample complexity for mean effect estimation.

7. 7. Theory II (lower bounds / impossibility): non-identifiability and arbitrarily wrong effects for off-manifold patching; constructions; implications for existing faithfulness metrics.

8. 8. Experiments I (ground truth): compiled/InterpBench/TRACR models with known circuits; comparing patch operators vs on-manifold patching; agreement with ground truth; stress tests across intervention implementations.

9. 9. Experiments II (real models): IOI/induction/refusal-direction case studies; measuring variance across patching schemes; hydra/backup indicators; computational overhead.

10. 10. Discussion and extensions: using on-manifold patching inside ACDC/causal scrubbing; implications for SDL/SAE-based nodes; limitations (manifold model error, distribution shift); 2026 outlook (agentic tool-use states, multimodal).

11. 11. Reproducibility and artifacts: released code, activation datasets, trained conditional generators, benchmark protocols.

1 Introduction

Mechanistic interpretability frequently appeals to *interventions* on a trained network: we modify an internal activation and observe a change in an outcome, with the intent of attributing causal responsibility to the modified coordinates. Activation patching, ablation, and their variants instantiate this principle by selecting a module index ℓ and a coordinate set $S \subseteq [d_\ell]$, overwriting $a_\ell[S]$ with some value v, and running the remaining computation to obtain an outcome Y. While this procedure is operationally clear, its *counterfactual semantics* is typically left implicit. In particular, patching specifies only the constrained coordinates $a_\ell[S] = v$ and is silent about the remaining coordinates $a_\ell[\bar{S}]$. Any concrete implementation must therefore supply a completion rule for $a_\ell[\bar{S}]$ (e.g. leave them unchanged, copy them from a donor example, set them to 0, add noise), and distinct completion rules can yield distinct estimates of the "effect" of the same intended intervention.

This under-specification matters because internal activations are highly structured objects induced by the data distribution and the network dynamics. When we overwrite $a_\ell[S]$ without respecting this structure, we typically create *off-manifold* states: vectors $\tilde{a}_\ell$ that have low or negligible probability under the model-induced conditional distribution of activations at layer ℓ. Downstream computation can be arbitrarily sensitive to such states, even when observational behavior on the original data distribution is unchanged. Consequently, intervention results can be dominated by distribution shift rather than by the causal role of the feature(s) of interest. The practical symptom is *implementation dependence*: the sign and magnitude of measured effects may vary substantially across standard patch operators, across choices of donor examples, or across seemingly innocuous normalization conventions. From a theoretical standpoint, such dependence indicates that the intervention is querying behavior outside the regime constrained by observation, and therefore outside the regime where a causal claim is identifiable without additional assumptions.

We highlight two recurring failure modes that illustrate why an explicit semantics is required. First, patching can interact with *redundancy* and *backup* computation: if multiple correlated internal representations support the same downstream behavior, then overwriting one representation off-manifold may spuriously activate alternative pathways (or suppress them) in a manner that does not correspond to any plausible counterfactual consistent with the model's typical internal states. Second, one may construct (and we empirically observe weaker analogues of) *gating-on-anomaly* behavior, in which some downstream subcomputation effectively detects implausible activation patterns (e.g. by norm thresholds or rare coordinate combinations) and routes computation differently. Such gating can be entirely irrelevant on the data distribution and yet dominate patched runs, producing large

"effects" that are artifacts of leaving the activation manifold.

To address these issues we propose a semantics, and a corresponding algorithmic approximation, for what it means to intervene on internal coordinates while remaining faithful to the model's native distribution of activations. Fix a context distribution $\mathcal{D}$ over inputs/positions/conditioning variables, and let $p_\ell(\cdot \mid c)$ denote the induced distribution of the layer state a_ℓ under $c \sim \mathcal{D}$. For a coordinate constraint $a_\ell[S] = v$ we define the target counterfactual completion as the conditional distribution

$$p_\ell(\cdot \mid c, a_\ell[S] = v),$$

and we define the corresponding on-manifold interventional expectation by sampling $\tilde{a}_\ell$ from this conditional and continuing the forward pass deterministically to obtain Y. This semantics is intentionally modest: it does not assert that the resulting counterfactual corresponds to an external structural causal model of the world; rather, it fixes a precise distributional meaning for "patch S to v" *within the model*, thereby removing ambiguity in the remaining degrees of freedom.

Because the true conditional $p_\ell(\cdot \mid c, a_\ell[S] = v)$ is not directly available, we approximate it by learning a conditional generator $q_\ell(\cdot \mid c, a_\ell[S] = v)$ from activation traces collected under $\mathcal{D}$. This choice converts patching into a conditional generative modeling problem: we learn to sample plausible full activations consistent with the imposed constraint, and we use Monte Carlo to estimate interventional expectations. Our theory formalizes the associated approximation error. Under bounded outcomes, we show that if q_ℓ is close to the target conditional in total variation distance then the induced interventional expectation is close, and under a Lipschitz condition on Y we obtain an analogous Wasserstein-based bound. These statements make precise the sense in which on-manifold patching is *stable*: improving the generator improves the causal estimate, and small distributional errors cannot produce large effect errors.

Our contributions are therefore threefold.

- We introduce an explicit distributional semantics for internal interventions in deterministic networks, defining counterfactual completions by conditioning on the model-induced activation distribution rather than by ad hoc completion rules.
- We give a practical two-phase procedure—data collection followed by conditional generation and Monte Carlo evaluation—that implements this semantics with a learned generator q_ℓ, and we propose a diagnostic based on the variance across valid implementations as a proxy for off-manifold sensitivity.
- We prove both (i) stability results showing that effect estimates depend continuously on the quality of the learned conditional generator and

(ii) an impossibility result demonstrating that common off-manifold patch operators can disagree by an arbitrarily large amount, even when observational behavior on $\mathcal{D}$ is identical.

Taken together, these results justify treating patching not merely as a coding pattern but as an inference problem with a specified target distribution, and they delineate the conditions under which patch-based causal claims can be made robust to implementation choices.

2 Background and related work

A substantial fraction of mechanistic interpretability work proceeds by treating internal activations as manipulable variables and estimating their influence on some downstream quantity. The most common template is to run a *clean* and a *corrupted* input, identify a set of internal coordinates, and then overwrite those coordinates in one run with values taken from the other, measuring the induced change in an outcome such as a logit difference or loss. This template appears under several names—*activation patching*, *activation replacement*, and *causal tracing*—and is typically instantiated at a chosen layer, attention head, or MLP block **??**. In a related vein, *ablation* studies set selected components to a baseline value (often 0) or remove their contribution and evaluate performance degradation **?**. These approaches provide operational evidence for the involvement of a component in a computation, but they leave open what counterfactual distribution over the unmodified coordinates is intended by the intervention.

Several refinements attempt to localize effects more precisely along computational routes. *Path patching* (and related "causal path" analyses) patches intermediate activations while holding fixed other parts of the computation in order to attribute an outcome change to a particular path through the network graph **?**. Similar motivations underlie *causal mediation* style decompositions in neural models, where one seeks to separate direct and indirect effects through specified internal variables **?**. *Causal scrubbing* formalizes a notion of a circuit being sufficient for a behavior by comparing a model to an abstracted computation that preserves only selected nodes/edges, with interventions used to test equivalence **?**. These methods, while conceptually distinct, share the requirement that an "intervened" run must specify a completion of the full internal state consistent with the imposed constraints.

The interpretability literature has also documented failure modes where intervention-based evidence is misleading. One class of issues can be understood as *interpretability illusions*: a procedure yields seemingly crisp attributions that are unstable, unfaithful, or sensitive to arbitrary implementation details. While the term is often discussed in the context of saliency maps and input perturbations **?**, analogous concerns arise for internal interventions: if the patched state is atypical, the model may enter regimes of computation

irrelevant to its on-distribution behavior, producing large but semantically spurious outcome changes. Empirically, practitioners have noted sensitivity to the choice of baseline in ablation, to which donor examples are used for patching, and to whether normalization layers are recomputed or frozen. Such sensitivity indicates that the intervention is confounded with distribution shift in the internal representation space rather than isolating the causal role of the intended coordinates.

Distribution shift is a general concern for perturbation methods. In input space, small-norm perturbations can leave the data manifold, and the resulting effects may reflect adversarial directions rather than meaningful feature removal **?**. In representation space, the situation is more acute: coordinates are not independent, and valid states often lie on a thin, context-dependent subset of $\mathbb{R}^{d_\ell}$. Consequently, "surgical" edits to a subset of coordinates can easily produce combinations that are never realized under the model-induced distribution. Some works attempt to mitigate this by adding noise, by using mean/variance-matched baselines, or by patching from carefully matched donor contexts; however, these heuristics still define completion rules implicitly, and they do not provide a target counterfactual distribution against which approximation quality can be assessed.

A complementary line of work studies *latent manipulation* and *model editing*, where one seeks to change behavior by modifying internal representations or parameters while preserving other behaviors **?**. Although the objectives differ, the technical obstacle is similar: unconstrained edits can create internal states (or parameter regimes) that cause unpredictable side effects. Techniques such as rank-one updates, constrained optimization, or editing within a learned subspace can be interpreted as imposing structure on allowable counterfactuals. Our focus is narrower: we do not edit parameters, but we seek a semantics for transient internal interventions that makes the allowable counterfactual completion explicit.

From the perspective of probabilistic modeling, our proposal aligns patching with *conditional generative modeling* of activations. There is an established practice of fitting generative models to internal representations, for purposes ranging from analysis to compression and sampling **?**. More directly relevant are *imputation* and *inpainting* problems: given a high-dimensional vector with some coordinates fixed, one samples the remaining coordinates from a learned conditional distribution. Modern generative model families—autoregressive transformers, masked autoencoders, normalizing flows, and diffusion models—support such conditional sampling, either by explicit factorization or by conditioning mechanisms **??**. In our setting, the conditioning additionally includes the external context c, reflecting that the activation manifold is context-dependent. The algorithmic primitive is thus: learn $q_\ell(\cdot \mid c, a_\ell[S] = v)$ from observational traces, and use it to generate full states consistent with the patch.

This framing yields two benefits relative to ad hoc completion rules.

First, it makes the target of approximation explicit: we are not merely "patching," but approximating a particular conditional distribution induced by the model and the context distribution. Second, it enables stability analysis using standard distances between distributions: if q_ℓ approximates the desired conditional in a metric such as total variation or Wasserstein distance, then expectations of bounded or Lipschitz outcomes are close. In the next section we formalize the objects needed to state these claims precisely: contexts c, layer states $a_\ell(c)$, coordinate sets S, outcomes Y, and the on-manifold interventional expectation defined by conditional completion.

3 Problem setup: internal counterfactuals in deterministic networks

We fix a trained, deterministic neural network M (e.g. a Transformer) equipped with *white-box* access that allows us to (i) run the forward computation and (ii) read and overwrite designated internal states at chosen modules. Although M is deterministic, the quantities we analyze are random variables induced by a distribution $\mathcal{D}$ over *contexts* c (prompts, conditioning variables, and any index specifying which token position is under study). Sampling $c \sim \mathcal{D}$ and executing M yields a sequence of internal activations; we write $a_\ell(c) \in \mathbb{R}^{d_\ell}$ for the activation at module/layer index $\ell \in \{1, \dots, L\}$ that we intend to manipulate.

To isolate downstream consequences of changing a_ℓ, it is convenient to factor the computation at ℓ. Let g_ℓ denote the deterministic "suffix" map from the module-ℓ state (together with the ambient context) to whatever quantity the remainder of the network produces before evaluation by a task-dependent readout. Concretely, we assume the outcome of interest can be written as

$$Y \;=\; h(g_\ell(a_\ell, c))\,, \tag{1}$$

where h is a measurable function specifying the evaluation metric (e.g. a logit difference, a loss, a class indicator, or an action). We treat Y as a real-valued random variable via $c \sim \mathcal{D}$ and $a_\ell = a_\ell(c)$; in later bounds we will impose either boundedness (e.g. $Y \in [0,1]$ after scaling) or a Lipschitz condition in a_ℓ for each fixed c.

An *internal intervention* specifies a subset of coordinates $S \subseteq [d_\ell]$ and a target value $v \in \mathbb{R}^{|S|}$ (possibly depending on c). We regard S as the object selected by an interpretability hypothesis (a head, MLP neurons, sparse features, or a low-dimensional subspace expressed in a basis), and we regard $v = v(c)$ as the operational content of "what we want those coordinates to be" (e.g. values taken from a contrast run, or values that encode a concept). The primitive operation available to us is coordinate overwrite: given a full state $\tilde{a}_\ell \in \mathbb{R}^{d_\ell}$ with $\tilde{a}_\ell[S] = v$, we run the suffix g_ℓ on input $(\tilde{a}_\ell, c)$ and record the resulting outcome $\tilde{Y} = h(g_\ell(\tilde{a}_\ell, c))$. The methodological question is: *how*

should we choose the remaining coordinates $\tilde{a}_\ell[\bar{S}]$, where $\bar{S} = [d_\ell] \setminus S$, so that $\tilde{Y}$ represents a meaningful counterfactual consequence of "setting $a_\ell[S] = v$"?

Formally, even in a deterministic network, a counterfactual intervention at layer ℓ requires specifying a distribution over full post-intervention states conditional on the context and the imposed constraint. For each context c we may consider the model-induced conditional distribution of activations $p_\ell(\cdot \mid c)$ obtained by sampling $c \sim \mathcal{D}$ and recording $a_\ell(c)$. Any proposed intervention semantics that "sets $a_\ell[S] = v$" implicitly selects a *completion rule*, i.e. a conditional distribution $r_\ell(\cdot \mid c, a_\ell[S] = v)$ supported on $\{a \in \mathbb{R}^{d_\ell} : a[S] = v\}$. The resulting interventional expectation is then

$$\mathbb{E}[Y \mid do_r(a_\ell[S] = v)] \;:=\; \mathbb{E}_{c\sim\mathcal{D}}\, \mathbb{E}_{\tilde{a}_\ell \sim r_\ell(\cdot \mid c, a_\ell[S]=v)} \left[h(g_\ell(\tilde{a}_\ell, c))\right]. \tag{2}$$

Different patching or ablation procedures correspond to different, typically unstated, choices of r_ℓ. Thus, before we can speak of causal effects of internal variables, we must fix a semantics that makes r_ℓ explicit.

We emphasize three desiderata for such internal counterfactuals. *(Well-definedness)* The semantics must yield a mathematically unambiguous target quantity for any specified $(\ell, S, v(\cdot), Y, \mathcal{D})$, so that two investigators running different implementations can in principle be judged against the same object. *(Stability)* Small changes in the completion rule within an appropriate class should not cause large changes in the estimated effect; equivalently, if two completion distributions are close in a standard metric (e.g. total variation or Wasserstein distance), then the induced expectations of Y should be close, under mild regularity conditions on Y. *(Estimability)* The semantics should admit approximation from observational traces produced by M on contexts from $\mathcal{D}$, without requiring oracle access to latent mechanisms beyond activations themselves. In particular, we seek an approach where the only learned object is a conditional generator trained on samples $(c, a_\ell(c))$, and where Monte Carlo evaluation of interventions has controlled variance under feasible computational budgets.

Finally, we define the *effect* we aim to estimate as a difference of two interventional expectations corresponding to two patch specifications v and v' (e.g. "clean" versus "corrupted" feature values):

$$\Delta \;:=\; \mathbb{E}[Y \mid do_r(a_\ell[S] = v)] \;-\; \mathbb{E}\big[Y \mid do_r(a_\ell[S] = v')\big]. \tag{3}$$

The remainder of our development chooses a particular completion semantics r_ℓ motivated by remaining on the model-induced activation manifold and then shows how to approximate it by a learned conditional generator while controlling the resulting error and variance.

4 On-manifold patching semantics

We now fix the completion rule by appealing to the distribution over activations that the model itself induces. For each layer ℓ and context c, the

forward pass of M produces a deterministic vector $a_\ell(c) \in \mathbb{R}^{d_\ell}$; randomness enters only through $c \sim \mathcal{D}$. Hence, for each c we obtain a (typically intractable) conditional distribution $p_\ell(\cdot \mid c)$ over $\mathbb{R}^{d_\ell}$, representing the variability of a_ℓ across draws from $\mathcal{D}$ sharing the same conditioning information c.[1] We informally refer to the *activation manifold at layer* ℓ (relative to $\mathcal{D}$) as the subset of $\mathbb{R}^{d_\ell}$ on which $p_\ell(\cdot \mid c)$ concentrates for typical contexts; the key point is not the topology of this set but that it is characterized by the model-induced law itself.

Given a coordinate subset $S \subseteq [d_\ell]$ and a patch value $v \in \mathbb{R}^{|S|}$, we want to define the counterfactual distribution of the entire state under the constraint $a_\ell[S] = v$ while remaining "on-manifold." The canonical object is the constrained conditional

$$p_\ell(\cdot \mid c, a_\ell[S] = v), \tag{4}$$

i.e. a regular conditional distribution supported on the affine slice $\{a \in \mathbb{R}^{d_\ell} : a[S] = v\}$. When $p_\ell(\cdot \mid c)$ admits a density (or more generally when standard disintegration hypotheses hold), (4) is well-defined as the conditional law of a_ℓ given the event $\{a_\ell[S] = v\}$. In continuous settings this event may have probability zero; we interpret (4) as the version of the conditional arising from disintegration with respect to the coordinate projection $\pi_S(a) = a[S]$, equivalently as the family of conditional measures appearing in the factorization

$$p_\ell(da \mid c) \;=\; p_\ell(da[S] \mid c)\, p_\ell(da[\bar{S}] \mid c, a[S]), \tag{5}$$

and then evaluating at $a[S] = v$. This interpretation is the one that our learned generator will approximate.

We define the *on-manifold intervention* $do_{\mathrm{OM}}(a_\ell[S] = v)$ by using (4) as the completion rule. Concretely, for each c we sample a full post-intervention state

$$\tilde{a}_\ell \sim p_\ell(\cdot \mid c, a_\ell[S] = v), \qquad \text{so that} \qquad \tilde{a}_\ell[S] = v \text{ a.s.}, \tag{6}$$

and then run the deterministic suffix computation to obtain $\tilde{Y} = h(g_\ell(\tilde{a}_\ell, c))$. The corresponding interventional expectation is

$$\mathbb{E}[Y \mid do_{\mathrm{OM}}(a_\ell[S] = v)] \;:=\; \mathbb{E}_{c\sim\mathcal{D}}\, \mathbb{E}_{\tilde{a}_\ell \sim p_\ell(\cdot \mid c, a_\ell[S]=v)} \left[h(g_\ell(\tilde{a}_\ell, c))\right]. \tag{7}$$

Thus the semantics is distributional: we intervene by imposing a coordinate constraint and otherwise sampling as the model typically would, conditional on that constraint. This is precisely what is violated by common off-manifold operators (zeroing, adding large noise, mixing independent coordinates), which generally produce states lying far outside the high-probability region of $p_\ell(\cdot \mid c)$.

[1] In implementations one chooses what information is included in c (e.g. full prompt prefix, metadata, token index). The semantics below is defined relative to this choice.

Two further remarks clarify why (4) is the natural completion. First, among all completion distributions supported on the slice $\{a[S] = v\}$, the constrained conditional is the least committal modification of the original law in an information-theoretic sense: for fixed c, it is the I-projection of $p_\ell(\cdot \mid c)$ onto the constraint set. Formally, whenever the constraint admits at least one distribution absolutely continuous with respect to $p_\ell(\cdot \mid c)$, we have

$$p_\ell(\cdot \mid c, a[S] = v) \;\in\; \arg\min_{r:\, r(a[S]=v)=1} \mathrm{KL}(r(\cdot) \,\|\, p_\ell(\cdot \mid c))\,, \tag{8}$$

which expresses that we change only what is required to enforce $a[S] = v$. Second, the choice preserves all correlations between $a[S]$ and $a[\bar{S}]$ that are present on the activation manifold, by sampling $a[\bar{S}]$ from the correct conditional law rather than from an unconditional or heuristic substitute.

It is also helpful to relate do_{OM} to standard causal formalisms. Consider an SCM in which exogenous noise is the context $c \sim \mathcal{D}$ and endogenous variables include the internal states $A_1, \ldots, A_L$ together with the output Y. Because the network is deterministic, one may write structural equations $A_\ell = F_\ell(A_{\ell-1}, c)$ and $Y = H(A_L, c)$ for suitable deterministic maps. In such an SCM, a *surgical* intervention on a subset of coordinates of A_ℓ is not uniquely specified unless we also specify how the remaining coordinates are generated after the intervention: replacing only part of a vector-valued variable does not determine a complete structural equation. The on-manifold intervention $do_{\mathrm{OM}}(A_\ell[S] = v)$ may be viewed as defining a new structural mechanism for the whole of A_ℓ under intervention, namely

$$A_\ell^{(\mathrm{int})} \sim p_\ell(\cdot \mid c, A_\ell[S] = v), \tag{9}$$

while leaving downstream mechanisms unchanged. Equivalently, do_{OM} fixes the marginal of $A_\ell[S]$ at the desired value and uses the observational conditional to supply a compatible draw for $A_\ell[\bar{S}]$, thereby avoiding counterfactual queries about off-support combinations of internal variables. In this sense do_{OM} does not posit a causal graph among individual coordinates inside A_ℓ; rather, it treats A_ℓ as a single endogenous variable and uses p_ℓ as an implicit, distributional substitute for unknown intra-layer causal structure.

Finally, we emphasize that (7) is an ideal target rather than an operational primitive: $p_\ell(\cdot \mid c, a[S] = v)$ is not directly accessible. The next section therefore constructs a learned approximation $q_\ell(\cdot \mid c, a[S] = v)$ from activation traces and uses it to implement do_{OM} approximately, with quantitative bounds controlling the induced error in effect estimates.

5 5. Algorithm: learning conditional activation generators; masked conditioning for arbitrary coordinate subsets; performing on-manifold interventions efficiently; practical design choices (which modules, which state types).

To operationalize the semantics above we require a procedure that, for a given layer ℓ, context specification c, coordinate subset $S \subseteq [d_\ell]$, and patch value $v \in \mathbb{R}^{|S|}$, can sample a full activation vector $\tilde{a}_\ell \in \mathbb{R}^{d_\ell}$ distributed approximately as the constrained conditional $p_\ell(\cdot \mid c, a_\ell[S] = v)$. Since p_ℓ is available only implicitly through forward executions of M on contexts drawn from $\mathcal{D}$, we learn a conditional generator $q_\ell(\cdot \mid c, a_\ell[S] = v)$ from activation traces and then use q_ℓ to implement approximate on-manifold interventions.

Data collection and conditioning. We first fix the instrumentation granularity: a module index ℓ and a definition of the internal state $a_\ell(c)$ to be patched (e.g. the residual stream at a specified token position, the MLP output pre-residual, or the concatenation of per-head attention outputs). We then sample contexts $c \sim \mathcal{D}$ and run M while logging the resulting states $a_\ell(c)$. The exact content of c is a modeling choice: including too little information produces a wide conditional distribution $p_\ell(\cdot \mid c)$ and makes constrained sampling difficult; including too much information (e.g. the entire prompt and position index) improves conditional predictability but may reduce effective sample size. In practice we treat c as whatever metadata will be available at intervention time (prompt prefix, token index, auxiliary conditioning variables), and we train q_ℓ to accept c as input.

Masked conditioning for arbitrary coordinate subsets. A central requirement is *arbitrary-subset conditioning*: we want a single learned model q_ℓ that can condition on *any* fixed subset of coordinates and generate the rest. To this end we represent a constraint by a triple (m, x^{obs}, c) where $m \in \{0,1\}^{d_\ell}$ is a binary mask with $m_i = 1$ indicating that coordinate i is clamped, and $x^{\mathrm{obs}} \in \mathbb{R}^{d_\ell}$ provides values on the clamped coordinates (arbitrary elsewhere). For the intervention $a_\ell[S] = v$ we set $m = \mathbf{1}_S$ and $x^{\mathrm{obs}}[S] = v$. The generator is trained to model the conditional law of a_ℓ given (c, m, x^{obs}), with the hard constraint implemented by construction: samples $\tilde{a}_\ell$ are post-processed (or parameterized) so that $\tilde{a}_\ell[S] = v$ exactly.

There are several equivalent training formulations. A convenient choice is masked conditional likelihood: we draw random masks m during training (e.g. coordinate-wise Bernoulli with rate ρ, or structured masks corresponding to blocks, heads, or learned subspaces), and maximize

$$\mathbb{E}_{c\sim\mathcal{D}}\, \mathbb{E}_{a_\ell \sim p_\ell(\cdot|c)}\, \mathbb{E}_{m\sim\mathcal{M}} \Big[\log q_\ell\big(a_\ell[\bar{m}] \mid c, m, a_\ell[m]\big) \Big], \tag{10}$$

where $\bar{m}$ denotes the complement coordinates. This objective directly matches the conditional completion rule we need at evaluation time: given (c, S, v), we sample $\tilde{a}_\ell[\bar{S}] \sim q_\ell(\cdot \mid c, m = \mathbf{1}_S, x^{\text{obs}}[S] = v)$ and then set $\tilde{a}_\ell[S] = v$. Architecturally, q_ℓ may be (i) an autoregressive model over coordinates (often expensive for large d_ℓ), (ii) a conditional normalizing flow (exact likelihood, moderate sampling cost), or (iii) a conditional diffusion/score model (high-quality samples, but iterative sampling). We emphasize that our subsequent use of q_ℓ is purely as a sampler; any family that supports conditioning on (c, m, x^{obs}) is admissible.

Efficient implementation of interventions. Given a trained q_ℓ, we implement an approximate on-manifold intervention by a two-stage run. First we execute the prefix of M up to module ℓ on context c to obtain any needed intermediate values and the base activation $a_\ell(c)$ (the latter is optional if v is supplied externally). Second we draw $\tilde{a}_\ell \sim q_\ell(\cdot \mid c, a_\ell[S] = v)$ and overwrite the internal state at module ℓ with $\tilde{a}_\ell$, then run the deterministic suffix computation to obtain the outcome Y. For Monte Carlo estimation we repeat the sampling of $\tilde{a}_\ell$ for the same c while caching the prefix computation, thereby reducing cost from $O(K\,T_M)$ to $O(T_{\text{prefix}} + K\,T_{\text{suffix}})$ per context, where T_{prefix} and T_{suffix} denote the costs of the corresponding segments. Batching is straightforward: we may sample K completions in parallel from q_ℓ and evaluate the suffix on a batch of patched states.

Patch specifications and practical design choices. The patch value may be a fixed vector v, a function of c (e.g. $v(c)$ produced by a secondary run on a contrast context), or a stochastic mapping (e.g. sampling v from an empirical distribution). Our implementation requires only that at evaluation time we can produce the desired v and the corresponding mask $m = \mathbf{1}_S$. Choosing S is likewise flexible: S may be a small set of coordinates (interpreted as features), a learned subspace (implemented by changing basis and masking in that basis), or a structured component (e.g. all dimensions of a particular attention head output). Empirically, modules with moderately sized states and clear functional roles (late residual streams at a token position, MLP activations, head outputs) are often easier to model with q_ℓ than extremely early layers or concatenations spanning many positions.

Finally, we note that training q_ℓ is an offline cost that can be amortized across many interventions at the same ℓ. Once learned, the same generator supports a family of interventions across different subsets S and values v, enabling systematic effect estimation with controlled completion semantics rather than ad hoc off-manifold replacements.

6 6. Theory I (upper bounds): stability of estimated effects under generator approximation error; bounds in TV and Wasserstein; when effect function is Lipschitz/bounded; sample complexity for mean effect estimation.

We now quantify how errors in the learned completion model propagate to errors in estimated on-manifold effects. Fix a layer ℓ, coordinates $S \subseteq [d_\ell]$, and a patch rule v (possibly depending on c). For each context c we define the ideal on-manifold interventional mean

$$\mu_v(c) \;:=\; \mathbb{E}_{\tilde{a}\sim p_\ell(\cdot|c,\tilde{a}[S]=v(c))}\big[Y(\tilde{a},c)\big], \qquad \mu_v \;:=\; \mathbb{E}_{c\sim\mathcal{D}}\big[\mu_v(c)\big], \tag{11}$$

and the corresponding quantity induced by our learned generator

$$\tilde{\mu}_v(c) \;:=\; \mathbb{E}_{\tilde{a}\sim q_\ell(\cdot|c,\tilde{a}[S]=v(c))}\big[Y(\tilde{a},c)\big], \qquad \tilde{\mu}_v \;:=\; \mathbb{E}_{c\sim\mathcal{D}}\big[\tilde{\mu}_v(c)\big]. \tag{12}$$

The object of interest is typically a contrast $\Delta_{\mathrm{OM}} = \mu_v - \mu_{v'}$, estimated by $\widehat{\Delta}_{\mathrm{OM}} = \widehat{\tilde{\mu}}_v - \widehat{\tilde{\mu}}_{v'}$ where $\widehat{\tilde{\mu}}_v$ denotes a Monte Carlo approximation of $\tilde{\mu}_v$ using J contexts and K completions per context.

Stability under total variation error. Assume throughout this paragraph that $Y(\tilde{a},c) \in [0,1]$ for all $(\tilde{a},c)$, which covers normalized losses, accuracies, and suitably scaled logit differences. For fixed c and constraint $\tilde{a}[S] = v(c)$, the map $\tilde{a} \mapsto Y(\tilde{a},c)$ is a bounded test function, hence its expectation differs by at most the total variation distance between the corresponding constrained conditionals. Writing

$$\varepsilon_v(c) \;:=\; \mathrm{TV}\big(p_\ell(\cdot \mid c, \tilde{a}[S]=v(c)),\; q_\ell(\cdot \mid c, \tilde{a}[S]=v(c))\big), \tag{13}$$

we obtain the pointwise bound $|\mu_v(c) - \tilde{\mu}_v(c)| \le \varepsilon_v(c)$. Averaging over $c \sim \mathcal{D}$ yields

$$|\mu_v - \tilde{\mu}_v| \;\le\; \bar{\varepsilon}_v \quad \text{where} \quad \bar{\varepsilon}_v := \mathbb{E}_{c\sim\mathcal{D}}[\varepsilon_v(c)]. \tag{14}$$

For a two-arm effect, the generator-induced bias is therefore additive:

$$|\Delta_{\mathrm{OM}} - (\tilde{\mu}_v - \tilde{\mu}_{v'})| \;\le\; \bar{\varepsilon}_v + \bar{\varepsilon}_{v'}. \tag{15}$$

A useful corollary concerns *intervention-implementation variance*: if two distinct generators q_ℓ and q'_ℓ both satisfy $\bar{\varepsilon}_v \le \varepsilon$ for the same intervention specification, then $|\tilde{\mu}_v - \tilde{\mu}'_v| \le 2\varepsilon$, i.e. the estimated effect is stable across implementation choices once both implementations are sufficiently on-manifold.

Stability under Wasserstein error via Lipschitz outcomes. When Y is not naturally bounded but is regular in a_ℓ, we can trade TV for W_1. Suppose that for each fixed c the function $\tilde{a} \mapsto Y(\tilde{a}, c)$ is L-Lipschitz in ℓ_2, meaning $|Y(\tilde{a}, c) - Y(\tilde{a}', c)| \le L\|\tilde{a} - \tilde{a}'\|_2$. Then by Kantorovich–Rubinstein duality,

$$|\mu_v(c) - \tilde{\mu}_v(c)| \;\le\; L \cdot W_1(p_\ell(\cdot \mid c, \tilde{a}[S] = v(c)),\, q_\ell(\cdot \mid c, \tilde{a}[S] = v(c)))\,, \quad (16)$$

and hence $|\mu_v - \tilde{\mu}_v| \le L\,\mathbb{E}_c[W_1(\cdot, \cdot)]$. In deterministic networks one may upper bound (or estimate) L by differentiating the suffix computation: if $Y(\tilde{a}, c) = \psi(g_\ell(\tilde{a}, c))$ with ψ Lipschitz and g_ℓ differentiable in $\tilde{a}$, then $L \le \sup_{\tilde{a}} \|\nabla_{\tilde{a}} Y(\tilde{a}, c)\|_2$. This yields a quantitative route to translating generator quality (in transport distance) into effect-estimation guarantees.

Monte Carlo estimation error and sample complexity. We next bound the deviation between $\widehat{\tilde{\mu}}_v$ and $\tilde{\mu}_v$. For each sampled context $c_j \sim \mathcal{D}$ we draw completions $\tilde{a}_\ell^{(k)} \sim q_\ell(\cdot \mid c_j, \tilde{a}[S] = v(c_j))$ independently and form

$$\widehat{\tilde{\mu}}_v \;=\; \frac{1}{J} \sum_{j=1}^{J} \left(\frac{1}{K} \sum_{k=1}^{K} Y(\tilde{a}_\ell^{(k)}, c_j) \right). \quad (17)$$

If $Y \in [0, 1]$, a direct Hoeffding argument (applied either to the JK pooled samples, or conditionally and then averaged) gives concentration of order $O((JK)^{-1/2})$ for $\widehat{\tilde{\mu}}_v$ around $\tilde{\mu}_v$, and therefore the same rate for the difference $\widehat{\Delta}_{\mathrm{OM}}$ around $(\tilde{\mu}_v - \tilde{\mu}_{v'})$. A more informative decomposition uses the law of total variance:

$$\mathrm{Var}(\widehat{\tilde{\mu}}_v) \;=\; \frac{1}{J} \mathrm{Var}_{c \sim \mathcal{D}}\big(\tilde{\mu}_v(c)\big) \;+\; \frac{1}{JK}\, \mathbb{E}_{c \sim \mathcal{D}}\Big[\mathrm{Var}_{\tilde{a} \sim q_\ell}(Y(\tilde{a}, c) \mid c)\Big], \quad (18)$$

which makes explicit the roles of J (covering context variability) and K (reducing completion noise at fixed c). In practice this suggests selecting K large enough to suppress the inner variance when q_ℓ is broad, while scaling J to control across-context heterogeneity.

End-to-end upper bound. Combining generator approximation error (TV or Wasserstein) with Monte Carlo concentration yields an additive end-to-end guarantee of the form

$$|\widehat{\Delta}_{\mathrm{OM}} - \Delta_{\mathrm{OM}}| \;\le\; (\text{generator bias}) \;+\; (\text{sampling error}), \quad (19)$$

where the bias term is controlled by $\bar{\varepsilon}_v + \bar{\varepsilon}_{v'}$ in TV (or the analogous $L\,\mathbb{E}[W_1]$ term), and the sampling error is $O((JK)^{-1/2})$ for bounded Y. This formalizes the basic promise of on-manifold patching: once we can learn constrained completions that are close to the true activation manifold, estimated causal effects become both accurate (small bias) and stable (small dependence on implementation), with the remaining uncertainty governed by standard Monte Carlo sample complexity.

Theory II: why off-manifold patching is not a semantics. We now justify, by lower bounds and explicit constructions, the claim that standard patching/ablation operators do not admit uniform correctness guarantees, even when the underlying network is deterministic and even when we have unlimited observational access on $c \sim \mathcal{D}$. The obstruction is *non-identifiability*: observational behavior constrains the model only on the support of its induced activation distribution, whereas most patch operators systematically query states that lie outside this support (or in regions of vanishing density). Any purported "causal effect" thereby depends not on the model as deployed, but on an arbitrary extension of its behavior to counterfactual activation configurations.

Activation-level counterfactuals are underspecified off-manifold. Fix a layer ℓ and coordinates S. A generic patch operator $\mathcal{P}$ specifies, for each context c, a rule producing a full patched activation $\tilde{a}_\ell = \mathcal{P}(c)$ satisfying $\tilde{a}_\ell[S] = v(c)$. Crucially, $\mathcal{P}$ also implicitly specifies a distribution over the remaining coordinates $[d_\ell] \setminus S$; e.g. zero ablation sets them to their unmodified values but changes only S, whereas cross-example patching copies a full state from a different context. Absent an explicit conditional distribution (such as $p_\ell(\cdot \mid c, a_\ell[S] = v(c))$), there is no canonical choice: multiple distinct completions are consistent with the same observational traces, and the downstream computation may behave arbitrarily differently on each completion.

Arbitrary disagreement between common patch operators. Theorem 4 formalizes this intuition. We sketch the mechanism because it clarifies what goes wrong. We construct a network with an internal *off-manifold detector* $D(a_\ell, c) \in \{0, 1\}$ that is identically zero on all activations $a_\ell(c)$ encountered under $c \sim \mathcal{D}$, yet equals one on a set of patched activations produced by $\mathcal{P}_1$ and equals zero on those produced by $\mathcal{P}_2$. Downstream, the model gates its computation on D: if $D = 0$ it implements the original task (thereby matching observational behavior), while if $D = 1$ it adds an arbitrarily large offset to the outcome-relevant computation (e.g. flips a classification label or shifts a logit difference by C). Because D never fires on-manifold, no amount of observational evaluation can reveal its existence; nevertheless, a patch operator that lands off-manifold reliably triggers the alternate branch. By choosing the magnitude of the gated effect, we can force $|\Delta_{\mathcal{P}_1} - \Delta_{\mathcal{P}_2}| \geq C$ for any prescribed C, and in particular make the sign disagree.

Two aspects are worth emphasizing. First, the construction does not require stochasticity, adversarial training, or pathological activations; it requires only that the activation manifold (the support of $p_\ell(\cdot \mid c)$) is a strict subset of $\mathbb{R}^{d_\ell}$, which is generically true in high dimension. Second, the disagreement is not a quantitative instability that disappears with more sam-

ples: it is a *semantic* instability caused by querying undefined counterfactual regions. This motivates treating "intervention-implementation variance" not merely as an experimental nuisance but as evidence that the intervention itself is ill-posed.

Implications for faithfulness and circuit metrics. Many existing faithfulness metrics implicitly rely on an off-manifold operator: zeroing a subspace, replacing it with a mean activation, adding noise, or swapping in a state from another example. Our impossibility result shows that such metrics cannot, in general, be interpreted as estimating a model-intrinsic causal quantity; rather, they estimate the causal quantity of the *pair* $(M, \mathcal{P})$, where $\mathcal{P}$ supplies an arbitrary completion semantics. Consequently, comparisons across papers (or even across implementations) can be confounded by differences in patch choice, normalization, hook location, or tensor shaping, each of which changes the induced off-manifold distribution. A practical corollary is that when two patch operators disagree, it is not meaningful to ask which one is "more correct" without an external criterion specifying the intended counterfactual.

Our on-manifold semantics resolves this by specifying the counterfactual distribution explicitly as the constrained conditional $p_\ell(\cdot \mid c, a_\ell[S] = v)$, and by approximating it with a learned q_ℓ. In this view, a faithfulness metric becomes well-defined only after we commit to the conditional distribution used to complete the intervened coordinates; stability across reasonable choices of q_ℓ is then an empirical diagnostic that we are indeed sampling near the activation manifold.

Lower bounds for learning the manifold. Theorem 5 explains why learning such completions is nontrivial. In the absence of structure, estimating $p_\ell(\cdot \mid c)$ (or the constrained version) to small total variation error requires sample size exponential in d_ℓ, by classical minimax lower bounds for density estimation. Thus, there is no "free" universally correct on-manifold patcher for arbitrary layers: success must come from exploitable regularities, such as low intrinsic dimension of activations, conditional factorization, or parametric inductive bias in q_ℓ. This also clarifies why naive heuristics (e.g. Gaussian matching of marginal moments) can appear plausible yet fail catastrophically: matching low-order statistics does not control TV or W_1 in high dimension, and therefore does not control counterfactual expectations.

Takeaway for the experimental agenda. The theory therefore yields two testable predictions. First, off-manifold patching should exhibit high intervention-implementation variance and can disagree even in sign across operators. Second, when we can learn q_ℓ well enough to remain on-manifold, effect estimates should become both more stable and more aligned with

ground-truth causal structure in settings where such ground truth is available. We now turn to such settings and quantify these predictions.

Experiments I (ground truth): compiled models with known circuits. We begin with settings in which the relevant causal structure is specified externally, so that we can compare intervention estimates to a reference quantity not defined in terms of any patch operator. Concretely, we use (i) *compiled* Transformers produced by TRACR (and related RASP-to-Transformer compilers), where intermediate variables of the source program provide a natural notion of "ground-truth mediators," and (ii) *InterpBench*-style synthetic tasks in which a small, human-auditable circuit is embedded into an otherwise inert network. In both cases we know, by construction, a set of internal coordinates/features (or linear subspaces) that are intended to implement a particular computation, and we can define a ground-truth interventional effect by intervening at the level of the source variable or the embedded circuit rather than by choosing an activation completion rule.

Reference effects and circuit labels. For each task we specify an outcome Y (typically a bounded logit difference or an indicator of correctness) and a patch specification $v(c)$ corresponding to a semantic change in an intermediate variable. In TRACR models, we obtain $v(c)$ by running the compiled program on a contrast input (or by directly modifying a program variable when the compiler exposes it), and we treat the compiler's alignment map as providing a layer–subspace pair (ℓ, S) expected to encode that variable. This yields a reference effect

$$\Delta_{\text{ref}} \;=\; \mathbb{E}_{c\sim\mathcal{D}}[Y \mid \text{program-level intervention}] \;-\; \mathbb{E}_{c\sim\mathcal{D}}[Y \mid \text{no intervention}]\,,$$

which is well-defined independently of any activation patching semantics. Separately, for circuit recovery we define ground-truth labels $G \subseteq [d_\ell]$ (or a ground-truth low-dimensional subspace) via the compiler's construction (TRACR) or via the planted circuit (InterpBench). These labels allow evaluation of how well an intervention method ranks features by causal importance.

Intervention methods compared. We compare on-manifold patching to several commonly used off-manifold operators. Given a targeted coordinate set S and patch value $v(c)$, we consider: (a) *zero/mean ablation* ($a_\ell[S] \leftarrow 0$ or $\leftarrow \mathbb{E}[a_\ell[S]]$), (b) *Gaussian noise* matched to marginal moments of $a_\ell[S]$, (c) *cross-example patching* (copying $a_\ell[S]$ from a different context c'), and (d) *swap-full-state* variants that copy an entire activation vector at layer ℓ (which often implicitly changes variables other than the intended one). For on-manifold patching we train $q_\ell(\cdot \mid c, a_\ell[S] = v)$ from activation traces on $c \sim \mathcal{D}$ and sample $\tilde{a}_\ell \sim q_\ell$ with the hard constraint $\tilde{a}_\ell[S] = v$ enforced by

construction. We then estimate $\widehat{\Delta}_{\mathrm{OM}}$ by Monte Carlo as in Algorithm OM-Patch.

Agreement with ground truth and faithfulness to intended mediators. We evaluate two notions of agreement. First, *effect agreement*: we compare each operator's estimated effect to Δ_{ref} across a suite of interventions (different variables, layers, and tasks), reporting absolute error and sign agreement. Second, *circuit agreement*: for each method we score features or subspaces by the magnitude of their estimated effect when patched (holding the patch semantics fixed), and we report standard ranking metrics with respect to G (e.g. AUROC or average precision when G is a set; subspace overlap when G is a span). The salient observation across compiled tasks is that on-manifold patching yields effect estimates that track Δ_{ref} more closely than off-manifold baselines when the intervention is intended to correspond to a semantic variable change. In particular, when the intervention is localized to a compiler-identified subspace, off-manifold operators frequently confound the intended change with distributional shift in the remaining coordinates, while q_ℓ-based completions tend to preserve the correlations necessary for the downstream computation to remain in the regime observed under $\mathcal{D}$.

Stress tests: intervention-implementation variance. To probe whether a method defines a stable semantics rather than a fragile implementation, we measure *intervention-implementation variance* by repeating each experiment under variations that should be inessential: different hook points that are algebraically equivalent (e.g. pre- vs post-residual-add where applicable), different tensor reshaping conventions for the same subspace, and different random seeds or architectures for q_ℓ (e.g. masked autoregressive Transformer vs conditional normalizing flow) while keeping the training data and conditioning information fixed. For each intervention we record the empirical variance of the estimated effect across implementations. Off-manifold operators typically exhibit large dispersion under these variations, consistent with the fact that each implementation induces a different off-manifold completion of $[d_\ell] \setminus S$. By contrast, when multiple q_ℓ instances achieve similar held-out likelihood (or similar reconstruction error under masked coordinates), the resulting $\widehat{\Delta}_{\mathrm{OM}}$ concentrates tightly, and disagreements between generators serve as a diagnostic that the manifold approximation is inadequate at the chosen (ℓ, S).

Negative controls and falsification checks. We include interventions on coordinates known (by construction) to be irrelevant to Y in the planted-circuit setting. A reasonable semantics should assign near-zero effect to these negative controls, up to sampling error. Off-manifold operators can spuri-

ously produce nontrivial effects on these controls due to distribution shift triggering downstream nonlinearities, whereas on-manifold patching substantially suppresses such false positives when q_ℓ is well-trained. We also report a *constraint satisfaction* check (exact equality on S) and a *manifold proximity* check (e.g. comparing discriminator scores or held-out conditional log-likelihood under q_ℓ), since these correlate with downstream stability.

Summary of what the ground-truth setting establishes. These experiments do not claim that q_ℓ learns p_ℓ in full generality; rather, they show that in settings where we can independently specify the intended counterfactual and identify a small set of mediator coordinates, enforcing an explicit conditional completion semantics improves both (i) agreement with external ground truth and (ii) robustness across innocuous implementation choices. This ground-truth evidence motivates applying the same methodology to real pretrained models, where no program-level reference effect exists and stability diagnostics become correspondingly more important.

Experiments II (real models): case studies without external ground truth. We next evaluate on-manifold patching on pretrained language models where no program-level reference effect is available. Here our target is not agreement with Δ_{ref} but rather (i) *semantic coherence* of the induced counterfactuals (the suffix computation remains in-distribution relative to $\mathcal{D}$), (ii) *stability* of estimated effects under innocuous implementation choices, and (iii) *diagnostics* for redundancy or "backup" mechanisms that complicate single-site causal attribution. In all case studies we fix a layer ℓ and intervention set S (either an attention-head output subspace, an SAE latent subset, or a hand-selected direction), define a patch value $v(c)$ from a contrast construction, and estimate $\widehat{\Delta}_{\text{OM}}$ with K generator samples per context and J contexts from $\mathcal{D}$, keeping Y bounded (logit differences clipped to $[0, 1]$ when required).

Indirect object identification (IOI): mediator-local interventions and variance. In the IOI task we consider prompts of the form "A and B went to the store. A gave a gift to _," with Y the logit difference between the correct indirect object token and the distractor. We define a contrast by swapping the roles of the two names while holding the rest of the prompt fixed, and we set $v(c)$ to the activation coordinates on S obtained under the contrast context (so $v(c)$ is a *semantic* patch value tied to a well-defined alternative input). For S we study (a) the output subspace of previously identified IOI-relevant attention heads and (b) subsets of SAE latents whose decoding directions align with name-referent features. Off-manifold baselines (zero ablation, mean ablation, cross-example copying of $a_\ell[S]$) often yield effects whose sign and magnitude depend strongly

on whether the hook is placed pre- or post-residual-add and on the precise tensorization used to represent the head subspace. Under on-manifold patching, we observe markedly reduced dispersion across these choices: if two implementations produce generators q_ℓ with comparable held-out conditional log-likelihood, then the corresponding $\widehat{\Delta}_{\mathrm{OM}}$ agree to within Monte Carlo error. When they do not agree, this typically coincides with clear failures of the manifold diagnostics (poor conditional reconstruction of held-out coordinates or elevated discriminator detectability), which we treat as evidence that (ℓ, S) is too large or that the conditioning metadata is insufficient (e.g. missing position information).

Induction: patching heads versus directions under controlled distribution shift. We also study induction behavior on synthetic repetition prompts sampled from a distribution $\mathcal{D}$ that controls sequence length and token entropy, with Y measuring the probability mass assigned to the repeated token at the induction position. We intervene on (i) induction-head outputs and (ii) known "induction directions" in the residual stream obtained by linear probes trained on whether a token is part of a repeated bigram. The salient phenomenon is that off-manifold operators can artificially suppress induction by pushing the residual stream into regions where downstream MLPs saturate, yielding large apparent effects even when the patched coordinates are not uniquely responsible for the behavior. By contrast, on-manifold completions preserve the co-activation structure between the patched head output and the surrounding residual coordinates (e.g. positional and frequency features), so the induced counterfactual more closely resembles "induction with a modified mediator" rather than "induction under distributional corruption." As a further stress test we vary $\mathcal{D}$ (different alphabet sizes and repetition rates) while keeping the generator trained on the original $\mathcal{D}$; the resulting degradation in stability aligns with the theoretical requirement that q_ℓ approximate the relevant conditional distribution. Empirically, effect estimates become less stable precisely when manifold diagnostics indicate covariate shift (e.g. sharp drops in conditional likelihood).

Refusal-direction interventions: isolating safety behavior from corruption. For refusal we use instruction-following models and define $\mathcal{D}$ over benign and harmful instruction prompts with a fixed template. We take Y to be a bounded refusal score (either a classifier probability or a logit difference between refusal and compliance tokens). We consider a patch set S corresponding to a "refusal direction" (a low-dimensional subspace in the residual stream) and define $v(c)$ by replacing the projection onto this subspace with that of a contrasting prompt class (harmful $\leftrightarrow$ benign), holding the orthogonal complement to be sampled by q_ℓ. A recurring failure mode of off-manifold ablations is to elicit incoherent outputs (degenerate repeti-

tion, sudden topic shifts) that inflate refusal scores for reasons unrelated to the intended safety mechanism. On-manifold patching substantially reduces these artifacts: completions $\tilde{a}_\ell$ sampled from $q_\ell(\cdot \mid c, a_\ell[S] = v(c))$ tend to preserve fluency and instruction adherence, allowing Y to more directly reflect changes in refusal-relevant features rather than generic corruption. We emphasize that this does not certify any particular normative interpretation; it only provides a more stable counterfactual semantics at the internal-state level.

Hydra and backup indicators: diagnosing redundancy via conditional heterogeneity. Real models often exhibit redundancy: multiple distinct internal pathways can sustain the same behavior. We operationalize "hydra" effects by measuring conditional heterogeneity of outcomes under the same patch constraint. Concretely, for fixed (c, ℓ, S, v) we examine the distribution of $Y(\tilde{a}_\ell, c)$ over $\tilde{a}_\ell \sim q_\ell(\cdot \mid c, a_\ell[S] = v)$, not only its mean. Large conditional variance (beyond Monte Carlo noise) indicates that the patched mediator does not uniquely determine the downstream behavior, consistent with backups that can be toggled by different completions of $[d_\ell] \setminus S$. We also compare single-site interventions to joint interventions on unions $S_1 \cup S_2$; superadditivity or sign reversals in $\widehat{\Delta}_{\mathrm{OM}}$ across these choices serve as further evidence of interacting redundant pathways rather than a single sparse circuit.

Computational overhead and practical regimes. Training q_ℓ is the dominant additional cost relative to standard patching. In our implementations we amortize data collection by logging a_ℓ for multiple ℓ in the same forward pass and train moderate-capacity conditional generators (masked autoregressive models or small Transformers) on N activation samples. At evaluation time the overhead is essentially multiplicative in K: we require JK suffix runs plus generator sampling, but we cache prefix computations and batch the K samples per context so that wall-clock time is typically bounded by a small constant factor over running JK standard forward passes. In practice we select K by monitoring stabilization of $\widehat{\Delta}_{\mathrm{OM}}$ and of the implementation-variance diagnostic; when these plateau, additional sampling yields diminishing returns.

7 Discussion and extensions

Our central claim is not that on-manifold patching is the unique "correct" counterfactual semantics, but that it is a semantics sufficiently explicit to admit (i) approximation guarantees and (ii) meaningful diagnostics when those guarantees are violated. This viewpoint suggests a family of extensions in which we treat standard mechanistic-interpretability procedures as *pipelines*

that query counterfactuals, and we replace their implicit, underspecified patch operators by the explicit do_{OM} operator induced by $p_\ell(\cdot \mid c, a_\ell[S] = v)$ (approximated by a learned q_ℓ). We highlight three directions: integration into circuit-discovery methods, implications for feature-based "nodes" (e.g. SAE latents), and limitations and outlook.

Using on-manifold patching within ACDC and causal scrubbing. Procedures such as ACDC and causal scrubbing repeatedly evaluate whether an internal subcomputation is causally necessary (or sufficient) for an observed behavior by performing targeted interventions and measuring changes in an outcome Y. Abstractly, these algorithms require an operator that (a) clamps a chosen set of internal variables and (b) "fills in" the remaining variables in a way that is intended to preserve everything not explicitly intervened upon. Off-manifold patching implements (a) but leaves (b) ambiguous, which is precisely the failure mode captured by Theorem 4. On-manifold patching proposes a principled replacement: when ACDC considers severing or restoring an edge corresponding to coordinates S, we estimate the relevant effect by

$$\widehat{\mu}_{\mathrm{OM}}(v) \;=\; \mathbb{E}_{c\sim\mathcal{D}}\, \mathbb{E}_{\tilde{a}_\ell\sim q_\ell(\cdot\mid c, a_\ell[S]=v(c))}\big[\, Y(\tilde{a}_\ell, c)\,\big],$$

and compare it to the analogous estimate for a baseline value $v'(c)$. The algorithmic change is minimal—one replaces "overwrite and run suffix" by "overwrite, resample the complement from q_ℓ, and run suffix"—but the semantics become stable to implementation details insofar as the different implementations learn generators with similar conditional fit. In practice, one can plug OM-Patch into ACDC as a drop-in oracle for edge scoring, while using the held-out conditional likelihood (or any calibrated manifold diagnostic) to decide whether the score is trustworthy at a given (ℓ, S).

Counterfactual trace generation for scrubbing. Causal scrubbing and related "program replacement" methods often require not only scalar effects but entire counterfactual *traces*: one wishes to replace a subtrace by one computed under an alternative input, while keeping the remainder of the trace "as it would have been" under that replacement. On-manifold patching provides a way to generate such traces without appealing to unidentifiable off-manifold behavior. Concretely, if a scrubbed program prescribes a value $v(c)$ for a set of coordinates S at module ℓ, we can sample $\tilde{a}_\ell \sim q_\ell(\cdot \mid c, a_\ell[S] = v(c))$, continue the model deterministically, and (if desired) log subsequent activations for further downstream scrubbing steps. This yields a coherent notion of a counterfactual run consistent with the model-induced activation statistics on $\mathcal{D}$, at least to the extent that the sequential application of generators remains accurate. The natural technical question here is compositionality: if we intervene at multiple layers, errors in successive generators can accumulate. A conservative practice is to treat

each additional intervention as increasing approximation risk and to validate with increasingly strict diagnostics as the number of intervention sites grows.

Implications for SDL and SAE-based "nodes." Many contemporary circuit analyses define nodes not as individual neurons but as learned features (SAE latents, sparse directions, or subspaces) and then patch those nodes to test causal relevance. Our framework clarifies what it means to intervene on such objects: if S indexes SAE latents, then a patch $a_\ell[S] = v$ is best interpreted as a *constraint* on those latents, while the remaining degrees of freedom (including correlated latents and residual components) should be resampled from the conditional distribution induced by the model on $\mathcal{D}$. This matters because SAE features are generally not statistically independent; forcing a subset to atypical values while leaving the rest fixed can create implausible combinations that downstream components may respond to in arbitrary ways. By training q_ℓ to condition on subsets of latents, we can (i) test whether a feature is causally implicated in Y under a semantics that respects its typical co-activation structure, and (ii) quantify redundancy by examining the conditional distribution of Y under the same latent constraint (our "hydra" diagnostic). More broadly, this suggests a refinement of structure discovery for dictionaries: a set of latents is a robust mechanistic unit only if clamping them yields low conditional heterogeneity in downstream behavior.

Limitations: manifold model error and distribution shift. The primary limitation is that the quality of the counterfactual depends on how well q_ℓ approximates the relevant conditional distribution. Theorems 1–3 turn this into an explicit error term, but they do not remove the practical burden of training and validating q_ℓ. Two failure modes are especially important. First, *model error*: even on the training distribution, q_ℓ may be misspecified, underpowered, or inadequately conditioned (e.g. missing position, attention mask, or other metadata), yielding samples that satisfy $a_\ell[S] = v$ but drift off the true manifold elsewhere. Second, *distribution shift*: if we train q_ℓ on contexts from $\mathcal{D}$ and then evaluate interventions on a different distribution $\mathcal{D}'$, the relevant conditional $p_\ell(\cdot \mid c, a_\ell[S] = v)$ may change substantially, invalidating both stability and semantic-coherence claims. In both cases, we recommend treating manifold diagnostics as first-class outputs: if conditional likelihood, reconstruction accuracy, or detectability degrades, then the intervention result should be reported as unreliable rather than over-interpreted.

Outlook (2026): agentic tool-use and multimodal states. We expect the on-manifold perspective to become more valuable as models become more agentic and more multimodal. In tool-using systems, the "context" c

includes not only a prompt prefix but also external observations, tool outputs, and long-horizon state (memory buffers, scratchpads, or action histories). Interventions then naturally target *policy-relevant* internal variables (e.g. action-selection subspaces) while requiring on-manifold completion of the remaining state so as not to induce spurious failures unrelated to the intended mechanism. This likely necessitates sequential or stateful generators that model $p_\ell(a_{\ell,t} \mid c_t, a_{\ell,t}[S] = v_t)$ across time steps, rather than a single-step conditional at a fixed token position. For multimodal models, interventions at the interface between modalities (vision encoder outputs, cross-attention keys/values, audio embeddings) similarly demand conditional completion that preserves cross-modal consistency. We view these as natural extensions of the same semantic commitment: counterfactual internal states should be sampled from a distribution the model itself assigns non-negligible probability under the relevant operating regime.

Reproducibility and artifacts. Because our proposal replaces an underspecified intervention operator by an explicit conditional distribution, the primary reproducibility burden shifts from "how exactly did we patch?" to "what conditional distribution did we learn, and how well does it fit?" We therefore treat artifacts for (i) learning q_ℓ and (ii) evaluating $\widehat{\Delta}_{\mathrm{OM}}$ as first-class research outputs. Concretely, we release (a) code for OM-Patch end-to-end, (b) activation datasets for the layers/tasks reported in the paper where licensing permits, (c) trained conditional generators (including multiple seeds/architectures for stability checks), and (d) benchmark protocols that specify $\mathcal{D}$, intervention sites (ℓ, S), patch-value functions $v(\cdot)$, and outcome metrics Y.

Our codebase is organized as a single reproducible pipeline with two explicit phases matching Algorithm OM-Patch: *trace collection* and *counterfactual evaluation*. Trace collection scripts take as input a model identifier, a dataset sampler for $c \sim \mathcal{D}$, and a list of intervention sites; they then run M once per context and log the corresponding activations $a_\ell(c)$ together with all conditioning variables required to define $p_\ell(\cdot \mid c)$ in practice (token indices, attention masks, position encodings, modality tags, and any task-specific metadata). Evaluation scripts never re-log activations: they load a trained q_ℓ, apply coordinate constraints $a_\ell[S] = v(c)$, sample $\tilde{a}_\ell \sim q_\ell(\cdot \mid c, a_\ell[S] = v(c))$, and run the deterministic suffix to obtain Y. This separation makes it easy to rerun the same evaluation against alternative generator families (or alternative seeds) without conflating generator error with trace-collection differences.

To make comparisons meaningful, each experiment is described by a machine-readable configuration that fully instantiates the tuple

$$(\mathcal{D},\ \ell,\ S,\ v(\cdot),\ Y,\ N,\ J,\ K,\ \text{generator family, training hyperparameters}).$$

We provide canonical configs for all reported figures/tables. Each config

also pins the tokenizer version, model revision hash, and any non-default numeric precision choices. We record random seeds at three levels: dataset sampling (contexts), generator training (initialization and minibatch order), and Monte Carlo evaluation (sampling $\tilde{a}_\ell$). Where GPU nondeterminism cannot be fully eliminated, we report it: we run a small determinism check that repeats a fixed config multiple times and logs the observed variance relative to the Monte Carlo standard error.

Activation datasets are released with a stable schema designed for streaming and partial loading. Each dataset contains (i) a table of contexts c (raw text or structured inputs, plus tokenization outputs), (ii) a tensor store for $a_\ell(c)$ at each logged site, and (iii) metadata sufficient to reconstruct the forward-pass conditions (model version, preprocessing, truncation rules, and the exact definition of the indexed activation, e.g. "residual stream after MLP at position t"). We store tensors in a chunked, memory-mappable format (e.g. `zarr` or `hdf5`) and provide checksums for each shard to prevent silent corruption. For large-scale models where redistribution is restricted, we provide scripts that regenerate identical datasets from public checkpoints and public $\mathcal{D}$ specifications; in those cases we release only derived summary statistics needed for manifold diagnostics (e.g. per-coordinate means/variances and held-out conditional log-likelihood curves), not raw activations.

For conditional generators q_ℓ, we release both training code and trained checkpoints, together with a minimal "model card" per checkpoint. The card reports: the conditioning interface (exactly what is included in c), the masking/clamping mechanism used to enforce $a_\ell[S] = v$, architectural details, training set size N, and validation diagnostics. Since our theorems are phrased in terms of distributional proximity (e.g. TV or W_1), which we cannot compute exactly, we standardize a set of proxy diagnostics: held-out reconstruction error on randomly masked coordinates; calibration of constraint satisfaction (numerical equality on S to tolerance); and a two-sample detectability test between real activations and unconditional samples from $q_\ell(\cdot \mid c)$, stratified by context type. For conditional evaluation, we additionally report a "conditional realism" score comparing real samples from a_ℓ restricted to events with $a_\ell[S]$ near v (when such events exist) to samples from $q_\ell(\cdot \mid c, a_\ell[S] = v)$.

Benchmark protocols specify not only tasks but also the semantics of patch values. In particular, each benchmark fixes (i) how contrast prompts or alternative inputs define $v(c)$, (ii) how we align token positions across prompts when ℓ indexes a position-specific state, and (iii) which baseline $v'(c)$ is used in $\widehat{\Delta}_{\mathrm{OM}}$. For each protocol we provide an evaluation harness that outputs $\widehat{\Delta}_{\mathrm{OM}}$, a confidence interval (bootstrap over contexts combined with within-context Monte Carlo error), and an intervention-implementation variance diagnostic obtained by rerunning the same protocol over multiple valid q_ℓ checkpoints. The harness writes a complete provenance record: config hash, code version, checkpoint hash, and the list of contexts used for

J.

Finally, we provide a small suite of "sanity" benchmarks intended to catch common failure modes: (a) a no-op test where $v(c) = a_\ell(c)[S]$ and the estimated effect should be statistically indistinguishable from zero; (b) a symmetry test where two equivalent definitions of S (e.g. two bases for the same subspace) yield consistent effects when the generator is correspondingly reparameterized; and (c) a stress test varying $|S|$ to observe the tradeoff between intervention strength and generator fit. These artifacts are meant to make it routine to distinguish "the effect is small" from "the manifold model is unreliable," which is the practical distinction our semantics is designed to support.