The Semantic Gap: Statistical vs. Meaningful Hierarchies in SynthSAEBench

The Core Limitation: Statistics Without Semantics

You're absolutely correct - SynthSAEBench's hierarchy mechanism is purely **statistical**, not semantic. It enforces probabilistic dependencies (child features can only fire when parents fire) but has zero understanding of what those features actually mean. When we label a feature as "Deceptive Reasoning" with children "Goal Misrepresentation" and "Information Withholding," those labels are purely for our bookkeeping - the synthetic model doesn't know or care about deception, goals, or information. From the model's perspective, it's just:

- Feature 42 (parent): fires with probability 0.001, represented by direction vector $d_{42} \in \mathbb{R}^{768}$
- Feature 137 (child of 42): fires with probability 0.01 conditional on feature 42 firing, direction d_{137}
- Feature 291 (child of 42): fires with probability 0.015 conditional on feature 42 firing, direction d_{291}

There is no inherent relationship between d_{42}, d_{137}, and d_{291} beyond the statistical constraint. They could be pointing in completely unrelated directions in the activation space. The "hierarchy" is implemented as: `if c_42 == 0, then set c_137 = 0 and c_291 = 0` - a simple masking operation that has nothing to do with semantic content.

What's Missing: Semantic Structure in the Representation Space

The fundamental issue is that **semantics requires compositional structure in the representation itself**, not just in firing probabilities. In real language models, "Deceptive Reasoning" and "Goal Misrepresentation" have a semantic relationship because:

1. **Shared representational structure**: The activation patterns for "goal misrepresentation" literally contain components of "deceptive reasoning" - the child concept is composed from the parent concept
2. **Functional relationships**: When the model processes text about misrepresenting goals, the deceptive reasoning circuits are actually used/activated
3. **Geometric relationships**: The representation spaces have meaningful structure where "lying about goals" is genuinely closer to "deception" than to "honest reporting"

None of this exists in current SynthSAEBench. The features are just random orthogonal directions with a statistical dependency rule. When feature 137 fires, it adds $c_{137} \cdot d_{137}$ to the activation, which has no compositional relationship to what feature 42 adds ($c_{42} \cdot d_{42}$). They're just independent vectors that happen to have correlated firing patterns.

What Would Be Needed for True Semantic Hierarchies

To create genuinely semantic hierarchies in SynthSAEBench, we would need several fundamental changes:

1. Compositional Representation Structure

Child feature directions should be **composed from** parent feature directions, not independent of them. For example:

```
Parent: "Deceptive Reasoning" → d_parent
Child: "Goal Misrepresentation" → d_child =  ·d_parent +  ·d_specificity

where:
-  ·d_parent: The "deceptive reasoning" component (inherited)
-  ·d_specificity: The specific additional structure for "goals" vs other deception types
- d_specificity is orthogonal to d_parent
```

This creates genuine compositional structure where the child representation literally contains the parent representation plus additional information. When an SAE decomposes activations containing the child feature, it should ideally recover both components.

2. Semantic Basis Vectors with Interpretable Meaning

Instead of random orthogonal directions, define a set of **semantic basis concepts** that have interpretable meanings. For instance:

```
Semantic Bases:
- e_intent: "Intentionality/Agency" direction
- e_honesty: "Truthfulness" direction
- e_goal: "Goal-oriented behavior" direction
- e_other_aware: "Model of other agents" direction

Then construct features as combinations:
- "Deceptive Reasoning" = +1·e_intent -1·e_honesty +1·e_other_aware
- "Goal Misrepresentation" = +1·e_intent -1·e_honesty +1·e_goal +1·e_other_aware
- "Honest Goal Pursuit" = +1·e_intent +1·e_honesty +1·e_goal
```

The challenge is that we don't actually know what the "right" semantic basis vectors are, and they may not exist as simple linear directions. This is precisely the problem SAEs are trying to solve for real models!

3. Task-Relevant Functional Relationships

For hierarchies to be semantically meaningful, there should be **tasks** where the hierarchical relationship matters functionally. For example:

- A task that requires "Goal Misrepresentation" should necessarily invoke "Deceptive Reasoning" mechanisms
- Ablating the parent feature should impair performance on all child feature tasks
- The activation patterns should show that child tasks genuinely use parent representations

This would require coupling SynthSAEBench to synthetic **task datasets** where we can test functional relationships, not just generating random activations.

4. Grounded Semantics Through Data Distribution

Another approach is to generate activations that are actually grounded in meaningful data. Instead of random sampling, create activations that correspond to:

- Synthetic text snippets or scenarios with known semantic properties
- Specific reasoning patterns that we explicitly encode
- Distributions that mirror real language model activation patterns on semantically categorized data

For example, you could:

1. Take real LLM activations on text containing deception
2. Use dimensionality reduction to find the main directions in this "deception subspace"
3. Use these empirical directions as your synthetic feature vectors
4. Build hierarchies based on actual nested semantic content in text

The Fundamental Paradox

Here's the deep problem: **We want to use synthetic data to test SAEs because we don't know the true features in real models. But to create semantically meaningful synthetic features, we need to know what semantic structure looks like in representation space - which is exactly what we don't know.**

This creates a paradox:

- **Statistical SynthSAEBench** (current): We can create it, but it lacks semantic meaning, so it may not tell us how SAEs handle real semantic structures
- **Semantic SynthSAEBench** (proposed): Would test what we care about, but requires knowing the answer to the very question we're trying to investigate

What We Can Actually Do: Partial Solutions

Given this limitation, here are realistic approaches:

Approach 1: Weak Semantic Structure Through Correlation Patterns

We can't create true semantics, but we can create **statistical signatures that might correlate with semantics**:

- Features related to deception are highly correlated with each other
- Features related to deception are anti-correlated with honesty features
- Hierarchical relationships enforce realistic conditional dependencies
- Manifold structures capture continuous semantic dimensions (e.g., degree of deception as a manifold)

This won't give us true semantics, but it tests whether SAEs can handle the **statistical patterns** that real semantic structures might produce.

Approach 2: Hybrid Approach with Real Model Guidance

Use real LLM activations to guide synthetic feature construction:

1. Collect LLM activations on curated datasets with known semantic properties
2. Use PCA/ICA/sparse coding to find empirical feature directions
3. Use these directions as synthetic feature vectors in SynthSAEBench
4. Test whether SAEs trained on this "semi-synthetic" data recover the original directions

This grounds the synthetic features in real semantics while maintaining the control and ground truth of synthetic data.

Approach 3: Accept the Limitation and Focus on What It Tests Well

Explicitly acknowledge that SynthSAEBench tests **statistical decomposition**, not semantic understanding:

- Can SAEs handle hierarchical statistical dependencies? (Yes, we can test this)
- Can SAEs decompose correlated vs. independent features? (Yes, we can test this)
- Can SAEs discover rare features vs. common features? (Yes, we can test this)
- Do SAEs learn semantically meaningful hierarchies? (**No, we cannot test this with pure SynthSAEBench**)

Use it for what it's good at: controlled experiments on statistical properties of SAE learning, not as a complete substitute for evaluation on real semantically meaningful data.

Implications for AI Safety Research

For AI safety applications, this limitation means:

What we CAN test with statistical hierarchies:

- Whether SAEs can detect rare but statistically structured patterns (if scheming has characteristic correlation signatures)
- Scaling behavior when dangerous features are sparse
- Whether transfer learning works for statistical signatures
- Robustness of detection to feature correlations and hierarchical dependencies

What we CANNOT test:

- Whether SAEs actually understand what "deception" means compositionally
- Whether ablating "deception features" prevents deceptive reasoning in a functionally meaningful way
- Whether the hierarchical relationships we create match real semantic hierarchies in deployed models
- Whether our synthetic "scheming" features bear any resemblance to how real models represent scheming

Practical recommendation: Use SynthSAEBench to test **necessary but not sufficient** conditions. If SAEs fail on statistical hierarchies, they'll definitely fail on real semantic hierarchies. If they succeed on statistical hierarchies, they might still fail on semantic ones, so you need additional validation on real models.

Conclusion

You've identified the core limitation: SynthSAEBench's hierarchies are **statistical simulacra** of semantic structure, not genuine semantic hierarchies. The feature labels we assign ("Deceptive Reasoning," "Goal Misrepresentation") are for human interpretation only - the synthetic model has no semantic understanding. This is a fundamental constraint of working with synthetic data in the absence of knowing what true semantic features look like in representation space. The value of SynthSAEBench lies in providing controlled testbeds for statistical properties (sparsity, correlation, hierarchy, manifolds) that we believe are **necessary** for handling real semantic structures, while acknowledging it cannot fully capture the **compositional and functional** properties that define true semantics.