Semantic Hierarchies for AI Safety Research in SynthSAEBench

Overview: From Abstract Hierarchies to Semantic Structure

Yes, there is a powerful way to enforce hierarchies that provide semantic meaning for specific topics, including AI safety-relevant concepts like harmful actions, scheming, and deception. The key insight is that SynthSAEBench's hierarchy mechanism can be reinterpreted as a **semantic taxonomy** rather than just an abstract parent-child relationship. Instead of having arbitrary hierarchical features, we can deliberately construct hierarchy trees that mirror real conceptual structures, including dangerous capability domains. This transforms the synthetic benchmark from a purely statistical testbed into a **semantically grounded simulation** where we can study how SAEs decompose and represent structured knowledge that matters for AI safety.

Technical Implementation: Building Semantic Concept Trees

The implementation leverages SynthSAEBench's existing hierarchy infrastructure but populates it with carefully designed concept trees. At the root level, you might have broad domains like "Agency & Planning," "Social Manipulation," or "Situational Awareness." Each root spawns children representing subcategories—for example, under "Social Manipulation" you might have "Deception," "Persuasion," and "Coercion." These further branch into specific manifestations: "Deception" could have children like "Lying by Omission," "Fabrication," "Misdirection," and "Strategic Ambiguity." The crucial technical detail is that the **firing probabilities and correlations** can be tuned to match realistic deployment scenarios. For instance, scheming-related features might be: (1) rare overall (low p_i for root-level scheming features), (2) highly correlated with each other (when the model is scheming, multiple scheming-related features activate together), and (3) anti-correlated with honest behavior features (enforced through negative entries in the correlation matrix).

Here's a concrete example hierarchy for deception-related features:

```
Root: Deceptive Reasoning (p=0.001, rare but critical)
  Child: Goal Misrepresentation (p=0.01 | parent active)
     Grandchild: False Alignment Signaling (p=0.3 | parent active)
     Grandchild: Hidden Objective Pursuit (p=0.5 | parent active)
  Child: Information Withholding (p=0.015 | parent active)
     Grandchild: Strategic Omission (p=0.4 | parent active)
     Grandchild: Selective Disclosure (p=0.6 | parent active)
  Child: Strategic Misdirection (p=0.02 | parent active)
      Grandchind: Attention Manipulation (p=0.5 | parent active)
```

```
Grandchild: False Trail Creation (p=0.5 | parent active)
```

The hierarchical constraint means that "False Alignment Signaling" can only activate when both "Goal Misrepresentation" AND "Deceptive Reasoning" are active, creating a realistic dependency structure. You can further enhance realism by making some of these features **manifold features** rather than discrete ones. For example, "Degree of Deception" might be a circular manifold representing a continuous spectrum from "full honesty" to "complete fabrication," while "Target Sophistication" might be a 1D manifold representing the intended audience's ability to detect deception. This captures the reality that deceptive behavior isn't binary but exists along multiple continuous dimensions.

Safety Research Applications: Testing SAE Robustness on Dangerous Capabilities

This semantic hierarchy approach enables several critical AI safety research directions. First, **capability detection robustness**: Train SAEs on synthetic data containing rare but hierarchically structured dangerous capability features, then measure whether SAEs can reliably identify these features even when they're: (a) very sparse (appearing in $<0.1\%$ of samples), (b) entangled with benign features through correlation, or (c) expressed at different levels of abstraction in the hierarchy. If SAEs fail to consistently recover rare scheming features in the controlled synthetic setting, this tells us they'll likely miss such features in real models. Second, **transfer learning for monitoring**: Suppose you construct a synthetic dataset with a specific deception hierarchy and train SAEs that successfully learn to decompose it. You can then test whether the learned **geometric signatures**—the specific patterns of latent correlations, decoder cosine similarities, and activation statistics—transfer to SAEs trained on actual language models. If you find that certain structural patterns (e.g., a cluster of highly correlated latents with low firing probability and specific mutual exclusivity constraints) reliably indicate scheming-type hierarchies in synthetic data, you can search for the same patterns in real SAEs as a monitoring signal.

Third, **worst-case adversarial scenarios**: Use the semantic hierarchy to model scenarios where models actively hide dangerous capabilities. Create features for "Capability Concealment" that are anti-correlated with overt expressions of the same capabilities and test whether SAEs can still recover the underlying features when the model has learned to suppress obvious signals. For example, you might have:

```
Harmful Capability Feature (manifold: intensity of capability)
  Overt Expression (discrete, negatively correlated with concealment)
  Concealed Expression (discrete, fires when capability present but overt suppressed)
```

Train SAEs and measure whether they learn separate latents for concealed vs. overt expressions or conflate them. Fourth, **intervention testing with ground truth**: Because you have complete ground truth about which features represent dangerous capabilities, you can test whether ablating or clamping SAE

latents matched to these features actually prevents the corresponding behavior. This is impossible with real models where you don't know the true feature structure, but in synthetic data you can definitively answer: "If I ablate the SAE latents that claim to represent 'Strategic Misdirection,' do samples containing this ground-truth feature actually change?"

The power of this approach is that you can construct arbitrarily complex semantic hierarchies reflecting real AI safety threat models—power-seeking, deception, sycophancy, manipulation, goal misspecification—and then rigorously test whether interpretability tools can decompose these structures. You can vary the prevalence (how rare are scheming features?), the abstraction level (do dangerous capabilities appear at the root or deep in the hierarchy?), the correlation structure (are scheming features clustered together?), and the representation type (discrete vs. manifold) to understand the limits of SAE-based monitoring. This grounds AI safety research in concrete, reproducible experiments with known ground truth, allowing us to build confidence in interpretability methods before deploying them for actual model monitoring.