Okay, I've reviewed the files:

1. `epistemic_certainty_prompt_grid_template.txt`: This defines the core experimental design. You're systematically generating prompts where the underlying proposition (`PROPOSITION`) remains constant within a set (`CORE_ID`), but the way it's stated varies along two axes:
    - **Certainty Magnitude (** Scaling from vague/uncertain to definite/absolute.
    - **Certainty Type (** How the certainty is expressed (observational, declarative, authoritative, rhetorical).
      This structure is key – it aims to isolate the effect of epistemic framing on the model's representations while holding the core semantic meaning steady for each `CORE_ID`.
2. `capture_baseline_activations.py`: This script takes the prompt grid, runs each prompt through GPT-2, generates text, and then captures the mean MLP activations (from a specified layer) corresponding *only* to the *generated* tokens. It saves these baseline activation vectors, tagged with their `core_id`, `type`, and `level`, creating a dataset of how the model represents these different epistemic stances *without* direct manipulation.
3. `capture_intervened_activations.py`: This script does almost the same as the baseline capture, but introduces a crucial manipulation. For a specific neuron (`--target_neuron`) in a specific layer (`--layer`), it *clamps* that neuron's activation to a series of predefined values (`--sweep_values`, including `None` for a within-sweep baseline). It captures the resulting downstream MLP activations (again, averaged over generated tokens) for each prompt *under each intervention condition*. This creates a dataset showing how forcing a specific neuron's state influences the representation of the epistemically varied prompts.
4. `generate_basis_vectors.py`: This script processes the *baseline* activation vectors. Its purpose is to create meaningful reference directions or "basis vectors" in the activation space. It can operate in two main ways:
    - `single_plane`: Create two vectors, each being the *average* activation for a specific *subset* of prompts (defined by filters like `type=declarative,level=1` vs `type=rhetorical,level=5`). These two vectors define a 2D plane potentially representing a conceptual axis (e.g., low vs. high certainty of a specific type).
    - `ensemble`: Create multiple vectors, where each vector is the average activation for all prompts belonging to a specific category (e.g., the average vector for `type=observational`, the average for `type=declarative`, etc., possibly filtered further by `level` or `core_id`).
5. `utils.py`: A library of helper functions supporting all the other scripts – handling file paths, loading/saving data (NPZ, JSON), parsing the structured keys from filenames, calculating

mean vectors, performing the mathematical operations for Semantic Resonance Mapping (SRM) like normalization, bivector calculation, generating rotation matrices, and plotting results.

6. `analyze_srm_sweep.py`: This is the analysis workhorse. It takes a set of captured activation vectors (either baseline or intervened) and a basis file (generated previously). It then performs SRM:
    - It projects the input vectors onto directions within the plane(s) defined by the basis vectors.
    - For `single_plane` basis, it rotates a probe vector within that single 2D plane and measures similarity/counts at each angle.
    - For `ensemble` basis, it does this across multiple planes (pairs of basis vectors) and averages the results.
    - It groups the input vectors (e.g., by `level`, `type`, or intervention `sweep` value) and calculates the SRM curves (similarity/counts vs. angle) for each group separately.
    - It saves the numerical results and generates plots comparing the SRM curves for different groups. This allows visualizing how different conditions (e.g., increasing certainty levels, different intervention strengths) affect the alignment of activations with the conceptual directions defined by the basis vectors.

In essence:

You've set up a pipeline to:

1. Generate carefully controlled prompts varying epistemic certainty.
2. Capture the model's internal MLP states (activations) in response to these prompts, both naturally (baseline) and under direct neuron intervention.
3. Define meaningful directions/planes in the activation space based on aggregations of the baseline states (basis vectors).
4. Quantify and visualize how the captured activations (from baseline or intervened runs) "resonate" with these defined directions using SRM, allowing comparison across different prompt types, levels, or intervention conditions.

Yes, I think I'm getting a good feel for the structure, the data flow, and the goals of the experiment. It's a methodical approach to probing how epistemic nuances are represented and how specific neuronal activity might correlate with or influence those representations.

Ready for the next step when you are.