**Step 1: Data Preparation and Preprocessing**

* *Data Cleaning:* Ensure your hourly and daily returns are free from missing values, outliers, or anomalies. Consider winsorizing or robust scaling if necessary.
* *Scaling:* Scale returns by a constant factor (e.g., multiply by 100 or 1000) if numerical stability is needed.
* *Data Structuring:* Clearly structure your data into tensors with dimensions (weeks, trading\_days, hourly observations) to match your model's expected input shape.

**Step 2: Model Sanity Check and Initialization**

* *Prior Predictive Checks:* Before inference, perform prior predictive sampling to ensure your priors are reasonable and produce realistic simulated data.
* *Initialization (Optional):* Initialize parameters carefully. Poor initialization can lead to slow convergence or divergence. Consider initializing parameters close to empirical estimates (e.g., GARCH parameters from historical volatility estimates).

**Step 3: Inference Setup (SVI)**

*Optimizer Choice*: Use Adam optimizer with a small learning rate (e.g., 0.005 or 0.001) initially. Adjust learning rate dynamically if needed.

*Guide Design*: Your guide (variational distribution) should closely match the complexity of your model. Ensure all parameters have appropriate constraints and initializations.

*Elbo Loss Monitoring*: Track the Evidence Lower Bound (ELBO) loss during inference. ELBO should decrease smoothly. If it fluctuates or diverges, revisit initialization, priors, or guide structure.

Step 4: Incremental Complexity and Warm-Start

Start Small: Initially, run inference on a smaller subset of your data (e.g., 3-6 months) to quickly identify potential issues.

Warm-Start: Once inference converges on smaller data, use the learned parameters as initialization for inference on the full dataset. This significantly improves convergence speed and stability.

Step 5: Convergence Diagnostics and Validation

Parameter Stability: Check parameter stability across multiple inference runs with different random seeds.

Posterior Predictive Checks: Generate posterior predictive samples and compare them to actual observed returns. Evaluate visually and statistically (e.g., posterior predictive p-values, QQ-plots).

Regime Stability: Examine inferred regime probabilities and transitions. Ensure regimes are interpretable and stable over time.

Step 6: Model Simplification (if needed)

If inference remains challenging, consider simplifying the model temporarily:

Reduce the number of regimes (e.g., from 3 to 2).

Temporarily remove or simplify GARCH or AR(1) components.

Reduce the number of hourly observations per day (e.g., aggregate hourly returns into fewer intervals).

Step 7: Posterior Analysis and Out-of-Sample Validation

Posterior Summaries: Extract posterior means, credible intervals, and regime probabilities.

Out-of-Sample Testing: Use the posterior predictive distribution to forecast the next week's returns. Compare these forecasts with actual observed returns to assess predictive accuracy.

Hierarchical Clustering: After inference, use posterior latent factors or regime probabilities as features for hierarchical clustering to determine your long-short portfolio.

Step 8: Scaling to Multi-Asset

Once inference is stable and validated for a single asset or small subset, extend the model to multiple assets:

Introduce asset-specific latent factors or hierarchical structures.

Consider dimensionality reduction techniques (e.g., PCA on latent factors) to manage complexity.

Carefully monitor inference stability and computational efficiency as you scale.

Step 9: Computational Efficiency and Scalability

GPU Acceleration: Utilize GPU acceleration (PyTorch backend) to speed up inference.

Mini-Batching: If computationally feasible, consider mini-batching your data during inference.

Parallelization: Run inference for different assets or asset groups in parallel.

Step 10: Continuous Monitoring and Iteration

Regularly monitor model performance, inference stability, and predictive accuracy.

Iteratively refine your model, priors, and inference strategy based on ongoing validation and performance metrics.