**Unsupervised / Self-Supervised Pattern Learning**

**1. Autoencoders / VAEs (Variational Autoencoders)**

**🔹 How it works:**

* An autoencoder tries to **compress** and **reconstruct** the input time series.
* It learns a **latent representation** (a bottleneck vector) that captures key structural features of each segment.

**🔹 In your use case:**

* Feed your multi-timeframe window (Trigger + Pattern + Structure) into an autoencoder.
* Collect the latent vectors.
* Use **dimensionality reduction (e.g., UMAP/t-SNE)** or **clustering** (e.g., KMeans) on these vectors to **group similar pattern types**.

**✅ Benefits:**

* Uncovers **hidden structural similarities** between time segments (e.g., volatility contraction, stair-step accumulation).
* Enables **pattern library discovery** without any labels.

**2. Contrastive Learning (e.g., SimCLR-style)**

**🔹 How it works:**

* The model is trained to **map similar inputs closer** together in latent space, and **dissimilar ones further apart**.
* Unlike classification, it **learns what “looks similar” means**, without needing hard labels.

**🔹 Application to your case:**

* Create positive pairs: e.g., same pattern type with small distortions (jitter, scaling).
* Create negative pairs: e.g., breakout vs. choppy consolidation.
* Train the model to **pull positive pairs closer**, **push negative pairs away**.

**✅ Benefits:**

* Learns a **robust pattern embedding** that reflects structural relationships.
* Can later be used for **clustering or downstream forecasting**.

**3. Clustering + Human Labeling Loop**

**🔹 How it works:**

* Once you have embeddings from autoencoder or contrastive models:
  + Run clustering (e.g., KMeans, HDBSCAN).
  + Visualize representative samples from each cluster.
  + A human analyst **inspects and labels clusters**: e.g., “Breakout + Volume Spike”, “Falling Wedge”.

**✅ Benefits:**

* Minimal human effort → huge labeling gain.
* Enables you to **curate label definitions** based on actual market behavior, not theory alone.
* Resulting classes can be used to **train a supervised classifier**.

**📈 Strategy Recap: Discovery-to-Deployment Flow**

text

CopyEdit

1. Unsupervised / Self-Supervised Learning

└─ Use autoencoder or contrastive learning

└─ Get pattern embeddings per window

2. Clustering & Visual Inspection

└─ Group embeddings

└─ Inspect cluster samples

└─ Label the useful ones (optionally discard the rest)

3. Supervised Model Training

└─ Train CNN-LSTM/Transformer to classify new windows into labeled pattern types

4. Real-Time Forecasting

└─ Use pattern predictions (or embeddings) as input to forecasting model