TRANS

Temporal Sequence Architecture

System Handbook

Model	**eu_model.pt (V18 + Coil Focal Loss)**
Precision	**32.1% @ Top 15%**
Features	**14 temporal × 20 steps + 13 context**
Date	**January 12, 2026**

Executive Summary

PRODUCTION MODEL

eu_model.pt

V18 Architecture | Coil-Aware Focal Loss | 13 Context Features

32.1% Top 15% Precision

1.67x Lift vs Random

13.6% K2 Recall

2,416 Clean Patterns

TRANS is a production-ready consolidation pattern detection system designed to identify micro and small-cap equities poised for significant upward moves. The system employs a hybrid neural architecture combining LSTM, CNN, and attention mechanisms with a novel Coil-Aware Focal Loss function that specifically addresses the challenge of learning rare but valuable target patterns.

Key Innovations

• **Two-Registry System** — Separates pattern detection from outcome labeling, guaranteeing strict temporal integrity with no look-ahead bias.

• **13-Feature Context Branch** — Gated Residual Network processes 8 market context features plus 5 new coil features capturing pattern state at detection time.

• **Coil-Aware Focal Loss** — Custom loss function amplifying gradients for K2 (Target) patterns with high coil intensity, solving the critical K2 learning problem.

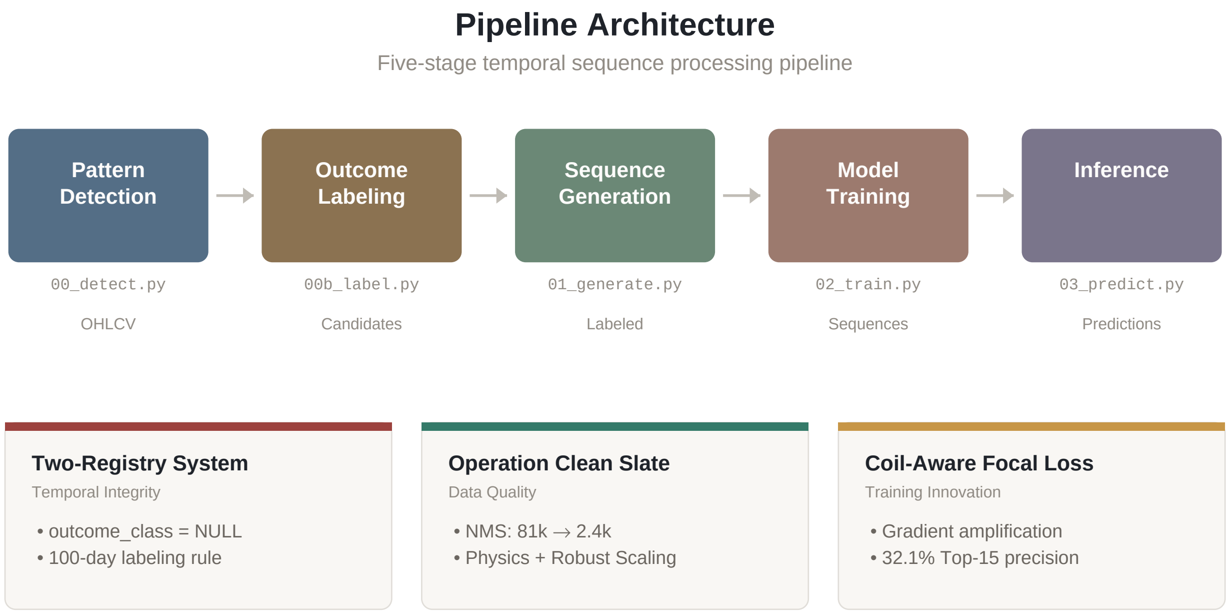
• **Operation Clean Slate** — NMS filter reduces 81,512 overlapping patterns to 2,416 unique consolidation events, eliminating train/val/test leakage.

Performance Metrics

Metric	Value	Benchmark	Status
Top 15% Precision	32.1%	> 30%	✓ Pass
Lift vs Random	1.67x	> 1.5x	✓ Pass
K2 Recall	13.6%	> 10%	✓ Pass
Clean Patterns	2,416	N/A	—

Pipeline Architecture

The TRANS pipeline processes market data through five distinct stages, each implemented as a separate module for maintainability and debugging. The architecture enforces strict temporal ordering to prevent any form of look-ahead bias.



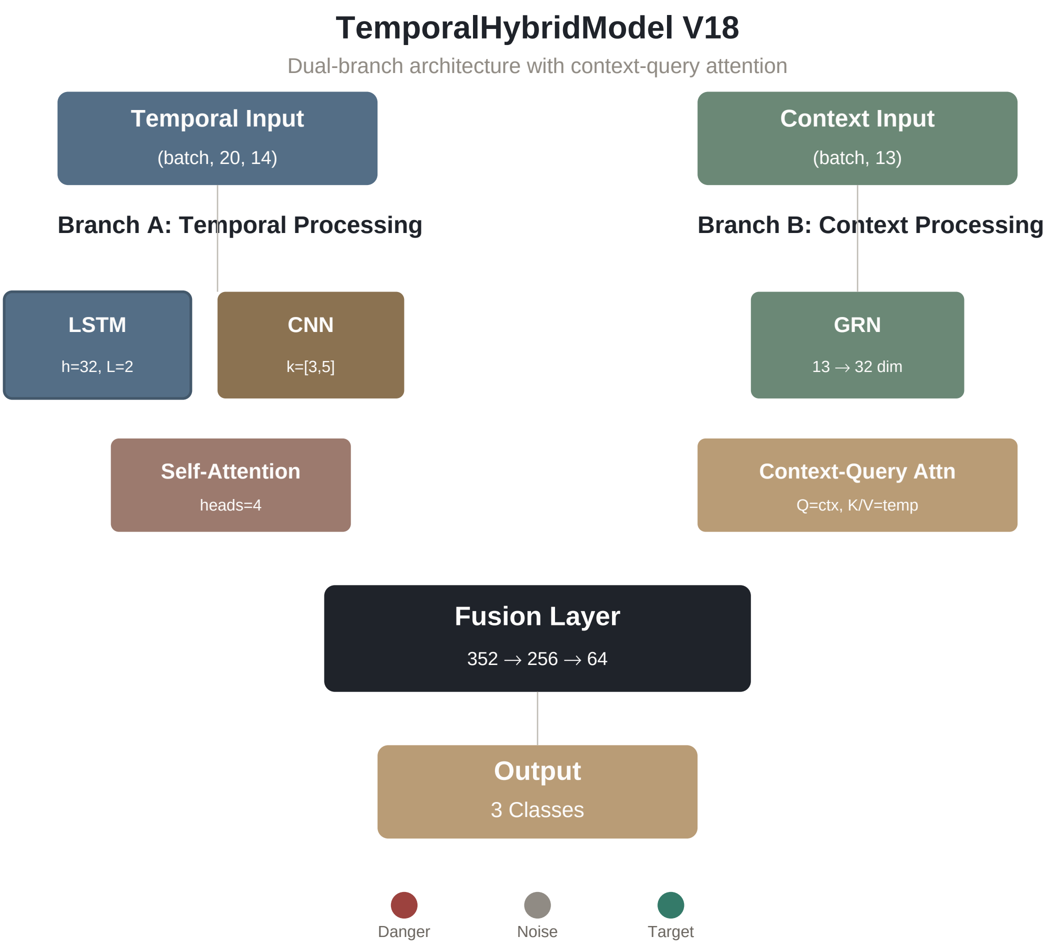
Two-Registry System

The two-registry architecture is fundamental to temporal integrity. Pattern detection and outcome labeling are completely decoupled processes:

Registry	Script	Output	Guarantee
Candidate	00_detect_patterns.py	candidates.parquet	outcome_class = NULL
Outcome	00b_label_outcomes.py	labeled.parquet	Labels only after 100 days

Neural Network Architecture

TemporalHybridModel V18 combines four processing branches: LSTM for sequence evolution, CNN for multi-scale pattern detection, Self-Attention for geometric structure analysis, and Context-Query Attention for regime-guided search.



Architecture Components

Component	Configuration	Output	Purpose
LSTM	hidden=32, layers=2	32 dim	Sequence evolution
CNN	kernels=[3,5], ch=[32,64]	96 dim	Multi-scale patterns
Self-Attention	heads=4	96 dim	Geometric structure
GRN	13 → 32 dim	32 dim	Context processing
Context-Query Attn	Q=ctx, K/V=temp	96 dim	Regime-guided search
Fusion	352 → 256 → 64	64 dim	Feature combination
Output	Linear(64, 3)	3 classes	Classification

Feature Engineering

The model processes two types of features: 14 temporal features computed for each of 20 timesteps, and 13 context features computed once per pattern. The context branch includes 5 new coil features introduced in January 2026.

Context Features

13 features processed by GRN branch

Market Context (8)

0	**float_turnover**	Accumulation activity
1	**trend_position**	200-SMA position
2	**base_duration**	Consolidation days
3	**relative_volume**	Vol vs 50-day avg
4	**distance_to_high**	% below 52w high
5	**log_float**	Log shares outstanding
6	**log_dollar_volume**	Log daily $ volume
7	**relative_strength_spy**	RS vs SPY 90-day

NEW: Coil Features (5)

Bias-free pattern state at detection

8	**price_position_at_end**	`(close-lower)/width`
9	**distance_to_danger**	`(close-stop)/close`
10	**bbw_slope_5d**	`BBW regression slope`
11	**vol_trend_5d**	`Volume trend slope`
12	**coil_intensity**	`Composite tension`

Key Insight

price_position_at_end < 0.4 shows 5.5x better K2 hit rate than >= 0.6
The coil features capture this predictive signal without any look-ahead bias.

Temporal Features (14)

Index	Feature	Normalization	Description
0-3	open, high, low, close	Relativized to day 0	OHLC prices
4	volume	Log ratio to day 0	Trading volume
5-7	bbw_20, adx, vol_ratio	Z-score (global)	Technical indicators
8-11	vol_dryup, var, nes, lpf	Robust (median/IQR)	Composite scores
12-13	upper, lower boundary	Relativized to day 0	Pattern bounds

Coil-Aware Focal Loss

The Coil-Aware Focal Loss is the key innovation enabling effective K2 (Target) pattern learning. Standard loss functions caused the model to avoid K2 predictions entirely due to precision penalties. This custom loss amplifies gradients for high-coil K2 patterns.

Coil-Aware Focal Loss

Solving the K2 learning problem through gradient amplification

Problem

Standard loss: K2 predictions = 0.04%
Model avoids K2 due to precision penalty

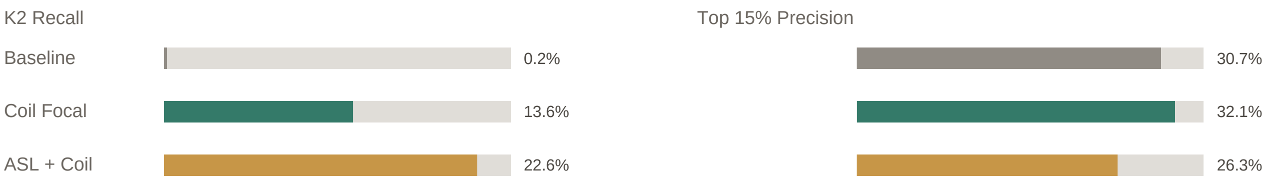
Solution

Coil Focal Loss: K2 predictions = 6.4%
Amplify gradients for high-coil K2 patterns

$$\text{loss\_K2} = \text{focal\_loss} \times (1 + \text{coil\_intensity} \times 3.0)$$

$$\text{coil\_intensity} = (1 - \text{price\_pos}) \times (1 - \text{bbw\_pctl}) \times \text{vol\_dryup}$$

Results Comparison



Implementation

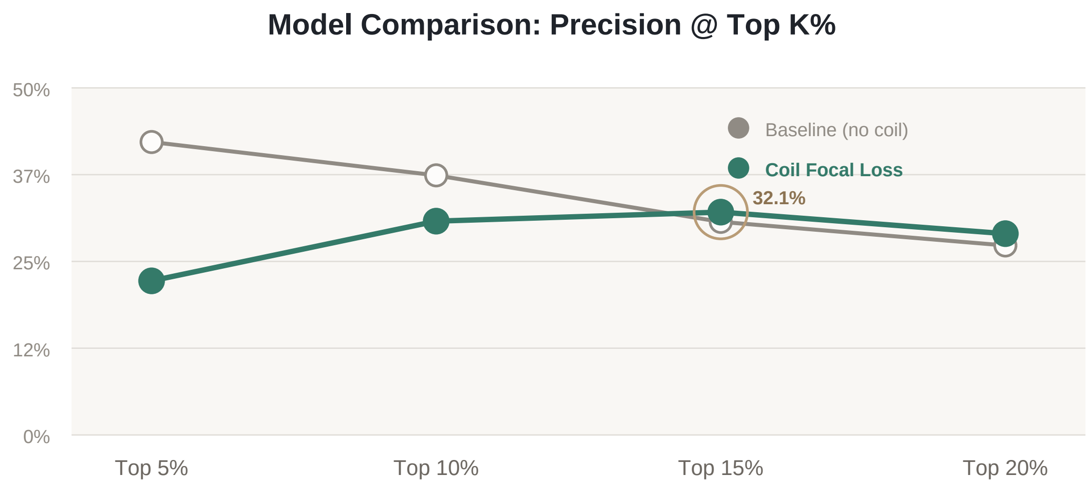
losses/coil_focal_loss.py

```
class CoilAwareFocalLoss(nn.Module):
    def __init__(self, gamma=2.0, coil_weight=3.0):
        self.gamma = gamma
        self.coil_weight = coil_weight

    def forward(self, inputs, targets, coil_intensity):
        ce = F.cross_entropy(inputs, targets, reduction='none')
        pt = torch.exp(-ce)
        focal = (1 - pt) ** self.gamma * ce

        # Amplify gradients for high-coil K2 patterns
        is_k2 = (targets == 2)
        boost = 1.0 + coil_intensity * self.coil_weight
        focal[is_k2] *= boost[is_k2]

        return focal.mean()
```


Model Comparison: Precision @ Top K%
50%
37%
25%
12%
0%
Baseline (no coil)
Coil Focal Loss
32.1%
Top 5%
Top 10%
Top 15%
Top 20%

CLI Reference

Training Pipeline

Training Commands

```
# Generate sequences with 13 context features
python pipeline/01_generate_sequences.py \
    --input output/detected_patterns.parquet \
    --apply-nms --apply-physics-filter

# Train with Coil-Aware Focal Loss
python pipeline/02_train_temporal.py \
    --sequences output/sequences/*.h5 \
    --use-coil-focal --coil-strength-weight 3.0 \
    --epochs 100 --train-cutoff 2024-01-01

# Generate predictions
python pipeline/03_predict_temporal.py \
    --model output/models/eu_model.pt
```

Key Flags

Flag	Default	Description
`--use-coil-focal`	Off	Enable Coil-Aware Focal Loss (recommended)
`--coil-strength-weight`	3.0	Coil intensity multiplier
`--apply-nms`	Off	Enable NMS de-duplication
`--apply-physics-filter`	Off	Remove untradeable patterns
`--train-cutoff`	None	Date for train/val split
`--compile`	Off	torch.compile for speedup