# ORDERED LIST REASONING using SEMANTIC LOSS

Di Franco Federico - Serratore Francesco
Visciglia Domenico

01

# Theoretical Foundations of Semantic Loss

# THEORETICAL MOTIVATION

## PROBLEM

Standard Neural Networks **ignore logical structure**

Output may violate **hard constraints** (e.g., $x_i > x_{i+1}$ in a sorted list)

## SOLUTION

**Semantic Loss ($L_{sem}$)**

- Acts as a regularizer for **logical consistency**
- Bridges **propositional logic** and **differentiable learning**

## GOAL

Inject **symbolic knowledge** into Deep Learning

Enforce **valid configurations** during training (semi-supervised signal).

# MATHEMATICAL FORMULATION

**CONCEPT:** Treat the **constraint α** as a **probabilistic event**

**FORMULA:**

$$\mathscr{L}_{\text{sem}}(p) \propto -\log \sum\_\{x \vDash \alpha\} P(x \mid p)$$

**COMPONENTS:**

- **p:** Output probabilities from the Neural Network
- **x:** A possible state (variable assignment)
- **x ⊨ α:** The set of all states where the constraint is α True

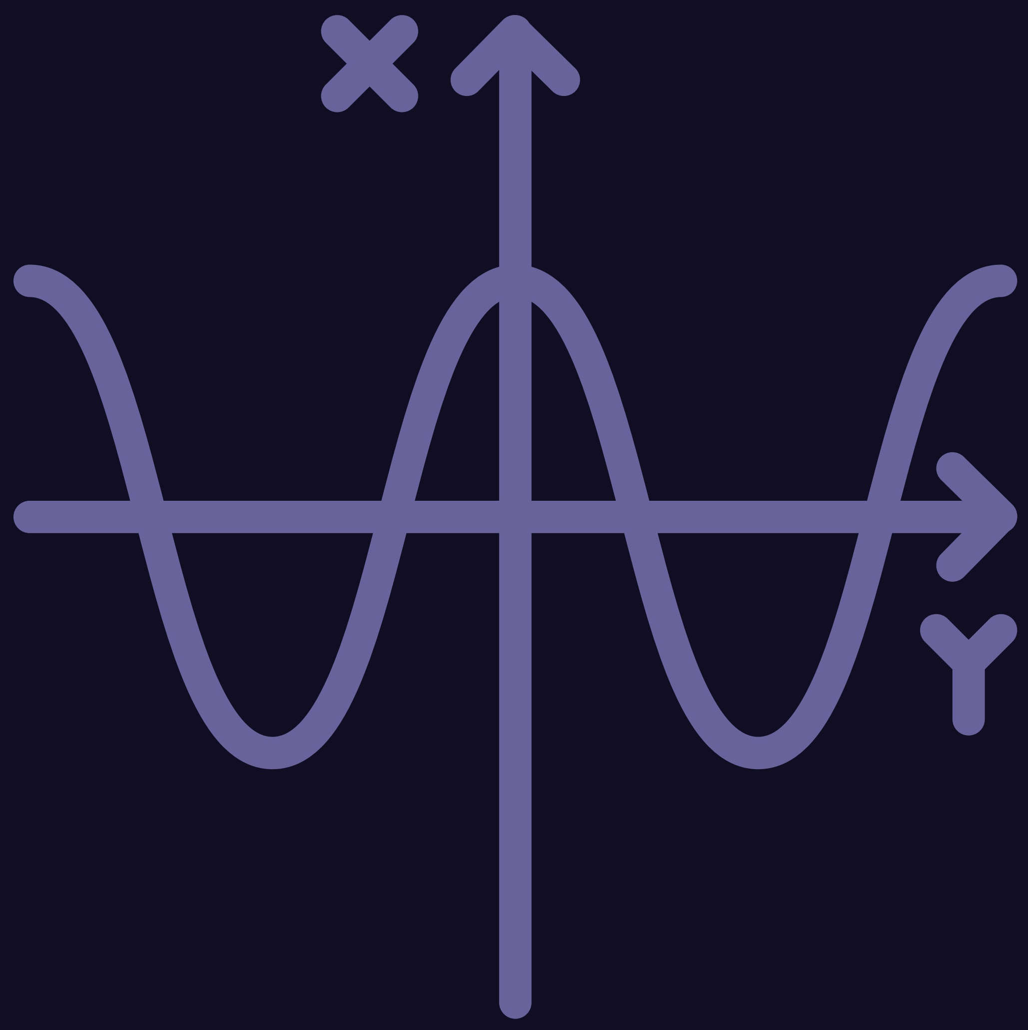
**INTERPRETATION:** Minimize the negative log-probability of generating a state consistent with α

# INTEGRATION & TRAINING

**LOSS FUNCTION:** $L_{total} = (1 - \lambda) L_{sup} + \lambda L_{sem}$

**KEY COMPONENTS:**

- ***$L_{sup}$:*** Standard Cross-Entropy (matches labels)
- ***$L_{sem}$:*** Penalizes logically invalid predictions
- **$\lambda$:** Hyperparameter controlling symbolic influence




**PROJECT CONTEXT:**

- **Constraint:** Monotonicity ($pos_i < pos_{i+1}$)
- **Outcome:** Improved robustness with limited data

02

# Connection between Propositional Logic and Differentiable Loss

# THE BRIDGE (WMC)

## PROBLEM

- **Logic is discrete** (True/False)
- **Loss functions** must **be continuous**

## SOLUTION

- Map **logical satisfaction** to **Weighted Model Counting (WMC)**
- **Calculates** the **total probability mass** of **all valid worlds**

## MECHANISM

- **Interpret NN outputs** as **variable probabilities**
- **Sum probabilities** of all **possible worlds satisfying constraint** α
- **Result** is a **differentiable scalar value** (probability mass of valid states)

# COMPUTATIONAL TRACTABILITY

## CHALLENGE

**Naive summation** over truth assignments **is #P-hard (exponential)**

## METHOD

**KNOWLEDGE COMPILATION:** Compile **Propositional Logic -> Arithmetic Circuit** (AC) (e.g., d-DNNF)

## ADVANTAGE

- AC allows **computing WMC** in **linear time** w.r.t. circuit size
- **AC operations** (Sum/Product) are **fully differentiable**
- Enables **efficient backpropagation** of **logical errors**

# FROM LOGIC TO GRADIENT

## PIPELINE:

**DEFINE:**
Logic Constraint α
(Hard Constraint)

**COMPILE:**
Generate Arithmetic
Circuit (Offline Step)

**FORWARD:**
Pass NN probabilities
into Circuit → Get
Probability of
Satisfaction

**BACKWARD:**
Compute Gradient of
$-\log(P_{sat})$ to update
NN weights

## OUTCOME:

The **logic constraint** drives the **gradient descent** towards **valid regions**

03

# Our Architecture and Training Methodology

# SEMANTIC LOSS I (SYMBOLIC CNF)

**METHOD:** Explicit construction of Propositional Logic trees (via SymPy)

**THE LOGIC:** For adjacent positions j, j+1:

- $(P_{i,j} \land P_{k,j+1}) \Rightarrow \mathrm{Val}(i) \leq \mathrm{Val}(k)$

**CONFLICT CLAUSE:** If Val(i) > Val(k), the configuration is invalid

- CNF Formula: $\neg P_{i,j} \lor \neg P_{k,j+1}$
- Probabilistic Eval: $1 - (P(i@j) \cdot P(k@j+1))$

**LIMITATION:** Requires recursive tree traversal on CPU

# SEMANTIC LOSS II (PAIRWISE OPTIM.)

**METHOD:** Matrix operations on GPU (Tensor-based)

**THE MECHANISM:** Compute **Joint Probability** for all pairs **simultaneously** using Batch Matrix Multiplication

**FORMULA:** $\mathscr{L}$_local $\propto -\log \sum_{i,k} ( P(i@j) \cdot P(k@j+1) \cdot \mathbf{M}_{i,k} )$

- $\mathbf{M}_{i,k}$: Validity Mask (1 if Val(i) ≤ Val(k), else 0)

**KEY ADVANTAGE:**

- **Transitivity:** Local checks (j < j+1) enforce global order
- **Speed:** Parallelizes the logic evaluation (O(N^2) matrix op)
- **Result:** Enables scalable training with symbolic guarantees

# MODEL & STRUCTURAL CONSTRAINTS

**ARCHITECTURE:** SortNet (MLP)

- **Input:** 5 Integers → **Hidden:** 64-32 (ReLU) → **Output:** 5x5 Matrix
- **Activation:** Softmax over positions

**CONSTRAINT 1: Exactly - One (Structural)**

- **Logic:** Every item must be in exactly one position; every position must hold exactly one item

**IMPLEMENTATION:** Differentiable penalty on probability mass

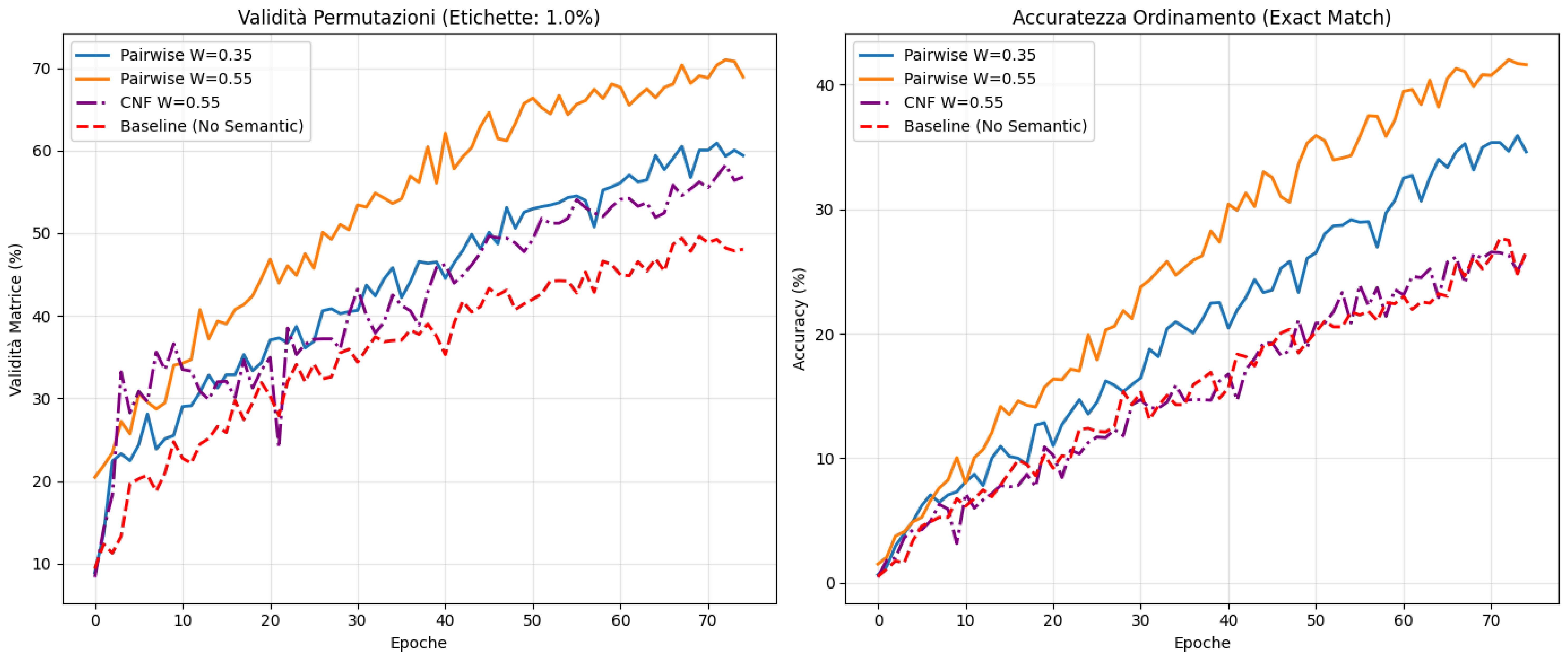
- **Rows:** $\sum_j P_{i,j} = 1$ (Item i placed once)
- **Cols:** $\sum_i P_{i,j} = 1$ (Position j filled once)

**ROLE:** Prevents the network from "cheating" (e.g., duplicating easy numbers)

04

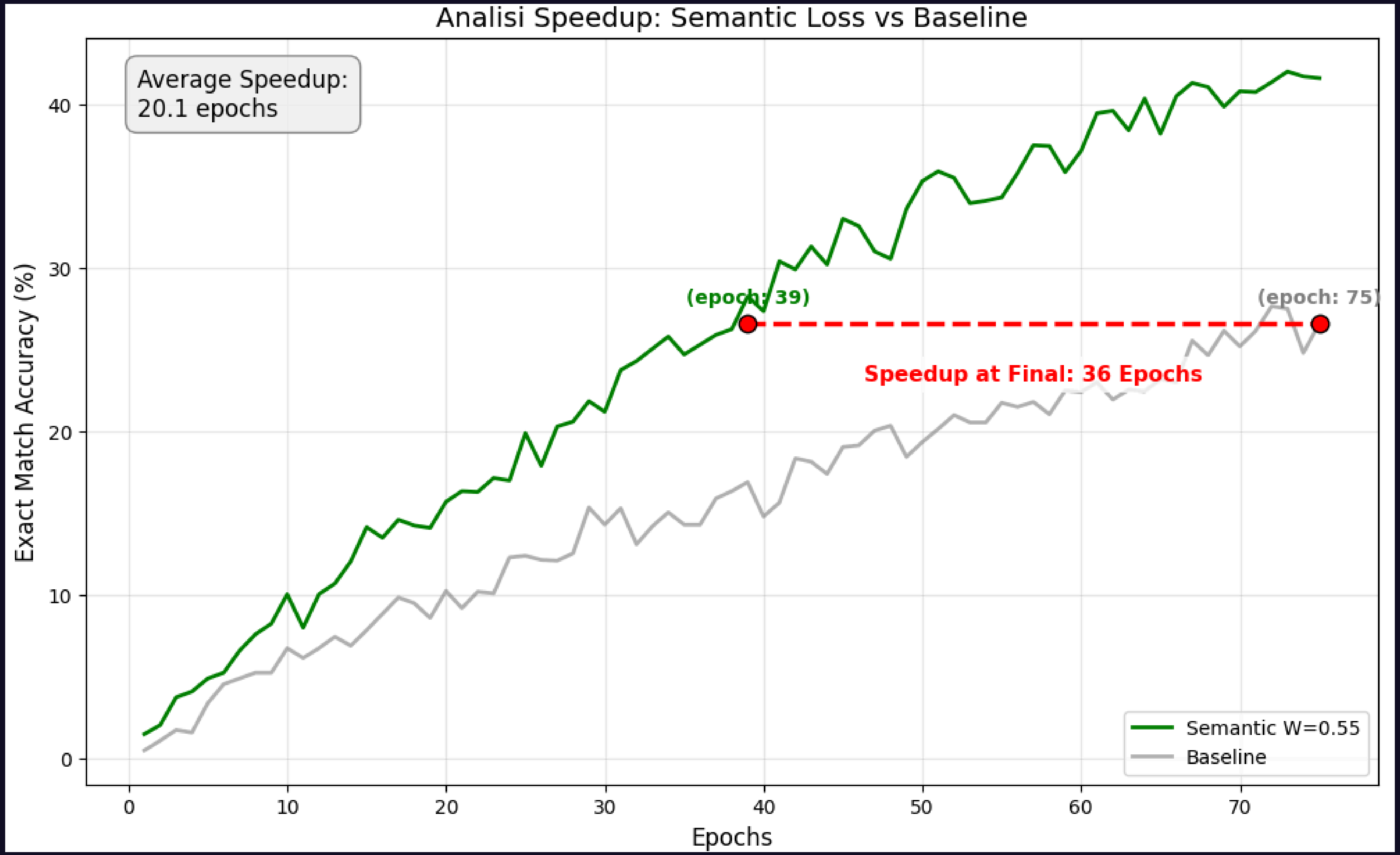
# Experimental Results and Comparisons

# LEARNING CURVE - 1% LABELS



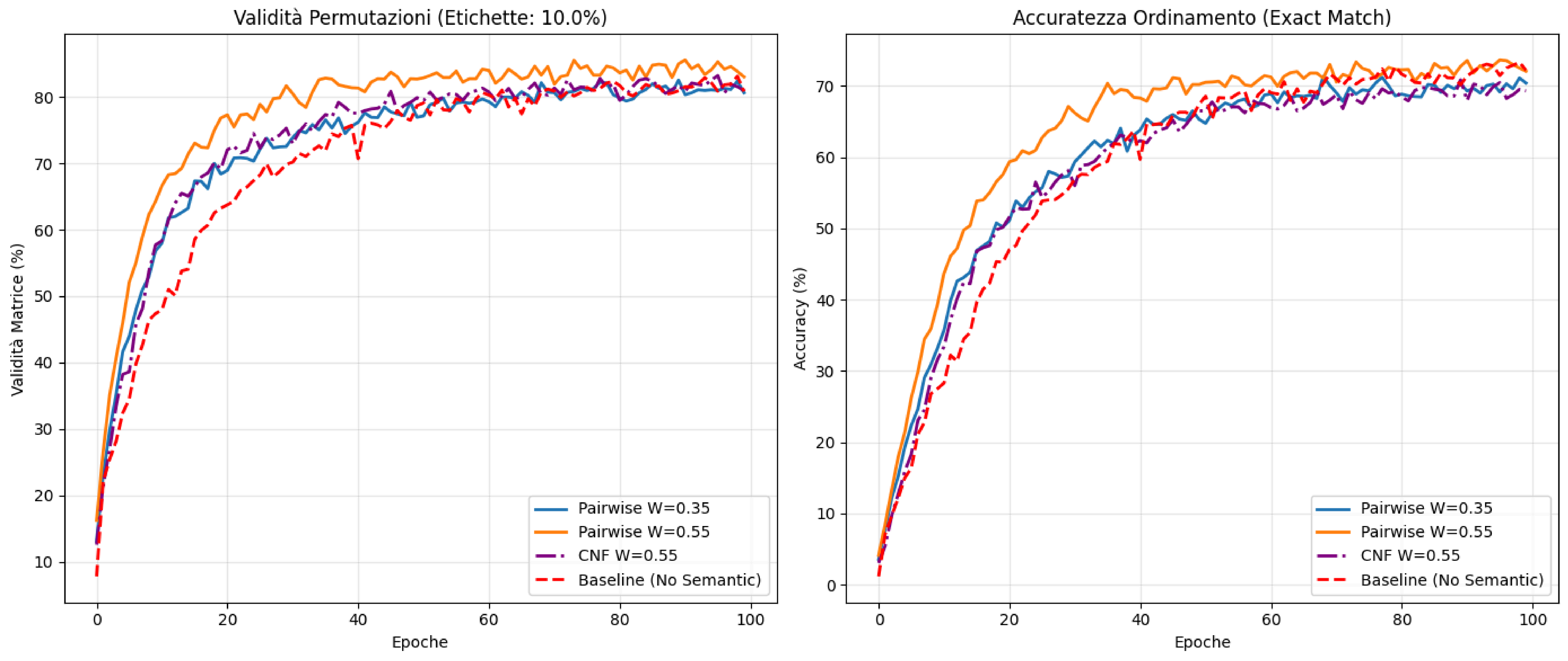

**INSIGHT:** Semantic Loss **drastically outperforms** the baseline in data-scarce scenarios, effectively **preventing overfitting** where supervised signals are insufficient

# CONVERGENCE SPEED - 1% LABELS



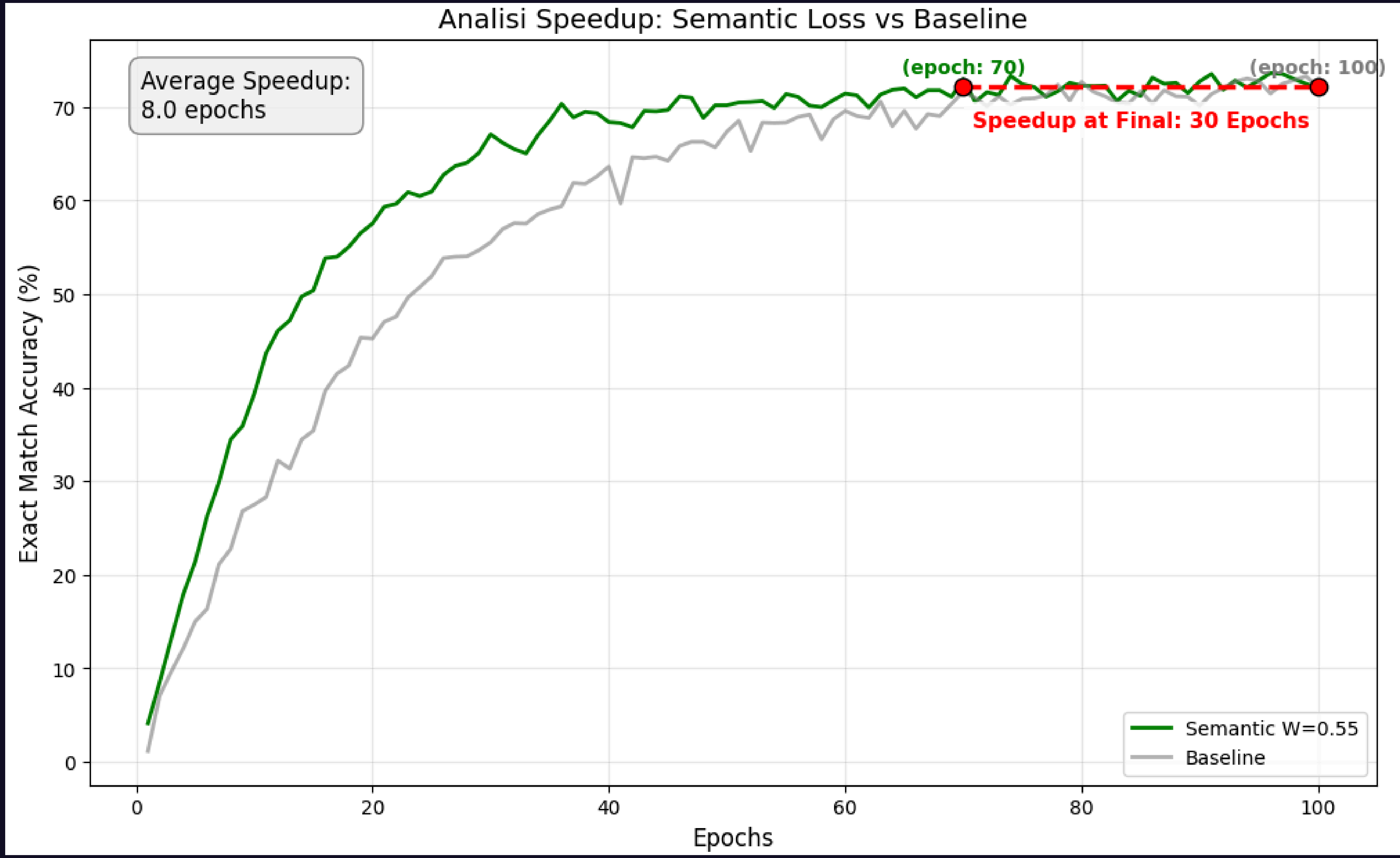

**INSIGHT:** The injection of logic acts as a **strong** "cold start" **accelerator**, allowing the model to reach **optimal performance significantly earlier** than the baseline

# LEARNING CURVE - 10% LABELS




**INSIGHT:** As labeled data **increases the performance gap** narrows, yet the Neuro-Symbolic model **maintains superior stability** and **lower variance** during training

# CONVERGENCE SPEED - 10% LABELS




**INSIGHT:** Even with sufficient data, symbolic regularization **enforces faster gradient descent, reducing the overall computational budget** required for convergence

05

# Advantages and Limitations of Semantic Loss

# PERFORMANCE MATRIX (1% LABELS)

| METRIC | SEMANTIC CNF | BASELINE | NEUROSYMBOLIC | Δ IMPACT(Baseline → Nesy) |
|---|---|---|---|---|
| EXACTLY-ONE (VALIDITY) | ~ 57% | ~ 48% | ~ 69% | +21% |
| EXACT MATCH (ACCURACY) | ~ 26% | ~ 27% | ~ 42% | +15% |
| EPOCHS-TO-27%-ACCURACY | / | 75 Epochs | 39 Epochs | -36 |

# PERFORMANCE MATRIX (10% LABELS)

| METRIC | SEMANTIC CNF | BASELINE | NEUROSYMBOLIC | Δ IMPACT(Baseline → Nesy) |
|---|---|---|---|---|
| EXACTLY-ONE (VALIDITY) | ~ 81% | ~ 81% | ~ 83% | +2% |
| EXACT MATCH (ACCURACY) | ~ 69% | ~ 72% | ~ 72% | +0% |
| EPOCHS-TO-72%-ACCURACY | / | 100 Epochs | 70 Epochs | -30 |

# ANALYSIS & KEY ADVANTAGES

**DATA EFFICIENCY (The "1% Label" Case):**

- **Observation: Massive accuracy gap** (+15%) when supervision is **scarce**
- **Why:** Semantic Loss acts as a **structural regularizer**, injecting "rules of the game" where labels are missing

**GRADIENT GUIDANCE:**

- **Observation: 2x faster convergence** despite **higher** per-epoch time
- **Why:** Logic provides **informative gradients** even for **unlabeled data**, actively steering the model away from invalid states

**GUARANTEED CONSISTENCY:**

- **Advantage:** Unlike standard MLPs, the NeuroSymbolic model **enforces strict constraints** (e.g., Exactly-One, Monotonicity), **ensuring output validity** by design

# LIMITATIONS & TECHNICAL ISSUES

**COMPUTATIONAL BOTTLENECK:**

- **Issue:** Exact logical inference (WMC) is **#P-Hard** and **scales exponentially**
- **Impact:** Pure symbolic implementation (CNF) is **slow**; **requires approximations** or **matrix-based optimizations** (Pairwise) to be **feasible on GPU**

**TRAINING OVERHEAD:**

- **Metric:** Logic calculation **increases training time per epoch** (**30x slower**)
- **Trade-off: Higher computational cost** per step vs. fewer total steps to converge

**HYPERPARAMETER SENSITIVITY:**

- **Risk:** If logic weight λ is **too high,** the model may **collapse to "Trivial Validity"** (satisfying rules but ignoring input features)

# THANKS!

Do you have any questions?

**CREDITS:** This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**