Process-Supervised Reward Models for Verifying Clinical Note Generation: A Scalable Approach Guided by Domain Expertise

Hanyin Wang et al.

Mayo Clinic Health System & UIUC

EMNLP 2025

Presented by Liwen Sun

Why This Paper Matters

Core Question

"How do we verify LLM outputs when there is no ground-truth answer — especially in medicine?"

This Paper's Answer:

- **Process-Supervised Reward Models (PRMs)** for clinical note generation
- **Domain-informed scalable supervision** without expensive human annotation
- **Step-by-step verification** that enables explainability and quality control

Preview:

1. Why clinical note verification is uniquely challenging
2. How to build PRMs without ground-truth answers
3. Domain-expert step decomposition + LLM-generated synthetic errors
4. State-of-the-art results: 98.8% verification, 56.2% preference accuracy

Section 1: Motivation - Clinical Note Verification Challenges

Why Clinical Note Verification Is Hard

1. **Clinical notes are open-ended**
   - No single "correct" answer
   - Multiple acceptable phrasings and styles
   - Subjective physician preferences
2. **Errors are subtle but high-stakes**
   - **Factual inaccuracies**: Wrong dosages, laterality errors (left vs right)
   - **Hallucinations**: Fabricated symptoms or history never discussed
   - **Omissions**: Missing critical follow-up instructions
   - **Unhelpfulness**: Vague phrasing ("follow up soon" vs "in 2 weeks")
3. **Current evaluation doesn't scale**
   - Expensive physician reader studies required
   - Manual review is time-consuming
   - Cannot keep pace with rapid AI deployment

Section 1: Motivation - Clinical Context

Ambient Scribing Adoption Is Accelerating

- **Current state**: AI systems generate clinical notes from patient-doctor dialogues
- **Projection**: 30% of healthcare market by 2025
- **Impact**: Substantial reduction in physician documentation burden
- **Challenge**: How to ensure quality at scale?

Contrast with Math/Code PRMs

Aspect	Math/Code	Clinical Notes
Ground truth	Exact correctness	No single answer
Verification	Deterministic	Subjective judgment
Error detection	Clear (wrong output)	Subtle nuances
Stakes	Low	Patient safety

Key insight: Need verification method that handles subjectivity while maintaining rigor

Section 1: Motivation - Reward Model Limitations

Outcome-Supervised Reward Models (ORMs)

- Assign single scalar score to entire generation
- **Pros**: Simple to train, well-established
- **Cons**:
  - No explainability - cannot identify where errors occur
  - Coarse-grained feedback
  - Limited guidance for improvement

Process-Supervised Reward Models (PRMs)

- Provide step-level verification
- **Advantages**:
  - **Explainability**: Pinpoint exact error locations
  - **Best-of-N filtering**: Better candidate selection
  - **Inference-time scaling**: Enable MCTS, beam search
  - **Step-level RL**: More granular training signals
- **Success stories**: Math reasoning (Lightman et al. 2023), code generation

Section 1: Motivation - The Gap

PRMs: Success in Math/Code, Gap in Open-Ended Domains

Why PRMs worked for Math/Code:

- Clear solution steps (one equation per step)
- Deterministic correctness verification
- Easy to collect step-level annotations

Why PRMs haven't worked for Clinical Notes:

1. **Verification of correctness**: No objective ground-truth
   - How do you know a clinical reasoning step is "correct"?
   - Multiple valid approaches to same clinical situation
2. **Process-supervised data collection**: Prohibitively expensive
   - Need physician annotations for **every step**
   - Cannot scale to thousands of samples
   - Domain expertise required at annotation time

This paper's contribution: Overcome both challenges with novel framework

Section 3: Task Definition

Ambient Scribing Task

Input: Patient–doctor dialogue (transcribed conversation) *Doctor: "Hello, what brings you in today?"*
Patient: "I've been having chest pain and shortness of breath."
Doctor: "When did this start? How severe is the pain?"
Patient: "About 3 days ago. It's worse when I exercise."

Output: Clinical note (focus on **Assessment & Plan**) *Assessment and Plan:*
1. Chest pain, likely angina
Assessment: 55-year-old male with 3-day history of exertional chest pain...
Plan: Order stress test, start aspirin 81mg daily, cardiology referral...

Goal: Train PRM to:

1. Verify notes step-by-step
2. Identify errors at precise locations
3. Select best note among multiple candidates (Best-of-N)

Section 3: Why Assessment & Plan Only?

Clinical Note Structure (SOAP Format):

- **S**ubjective: Patient's reported symptoms
- **O**bjective: Physical exam, vital signs, test results
- **A**ssessment: Diagnosis and clinical reasoning
- **P**lan: Treatment plan and follow-up

Why Focus on A&P?

1. **Core clinical reasoning**: Contains key diagnostic thinking and decisions
2. **Most error-prone**: Requires synthesis and medical judgment
3. **Time-consuming**: Most labor-intensive for physicians to write
4. **Highest value**: Directly impacts patient care and treatment
5. **Current gap**: After domain adaptation, LLMs generate high-quality Subjective/Objective sections, but notable gaps remain in A&P

Other sections: Subjective (nearly indistinguishable from human), Objective (imported from EHR)

Section 3: Base Datasets

Training Data Sources

1. **Dialogue-G** (1,205 cases)
   - Synthetic patient-doctor conversations
   - Generated by Gemini Pro 1.0
   - Gold-reference notes following "Best Practice" format
2. **ACI-BENCH** (67 cases from training subset)
   - Ambient Clinical Intelligence benchmark
   - Multiple conversation types: virtual assistant, scribe, natural dialogue
   - Notes regenerated with Gemini Pro 1.0 for consistency

"Best Practice" Format:

- Developed by panel of internal medicine physicians
- Problem-oriented charting approach
- Clear structure: Problem $\rightarrow$ Assessment $\rightarrow$ Plan
- Standardized level of detail and completeness

Key Point: Gold-reference notes are high-quality baselines, not perfect ground truth

Section 4: Methods - Step Definition (Most Important)

Core Challenge: What is a "step" in a clinical note?

Math/Code PRMs:

- Clear steps: one equation or line of code
- Sequential reasoning

Clinical Notes:

- Semi-structured text
- Multiple problems addressed simultaneously
- No obvious "steps"

Solution: Hierarchical Step Structure

1. **Problem-level step**: Problem description (e.g., "1. Congestive heart failure")
2. **Sentence-level steps**: Each sentence within problem
3. **Problem completeness step**: Checks if problem fully documented
4. **Note completeness step**: Checks if all problems addressed
5. **End-of-note step**: Overall quality score

Design Rationale:

- Aligns with problem-oriented charting (standard clinical practice)

Section 4: Methods - Step Definition Example

Original Clinical Note: *Assessment and Plan:*
1. Congestive heart failure exacerbation
Assessment: Patient presents with dyspnea and peripheral edema. Physical exam shows bilateral lower extremity edema. Echocardiogram reveals reduced ejection fraction at 35%.
Plan: Increase furosemide to 40mg twice daily. Restrict sodium intake to ¡2g/day. Follow up in 1 week to reassess fluid status.
2. Hypertension
Assessment: Blood pressure well controlled on current regimen.
Plan: Continue lisinopril 20mg daily.

Section 4: Methods - Step Definition Example

Transformed to Steps:

1. Problem: "Congestive heart failure exacerbation"
2. "Assessment: Patient presents with dyspnea and peripheral edema."
3. "Physical exam shows bilateral lower extremity edema."
4. "Echocardiogram reveals reduced ejection fraction at 35%."
5. "Plan: Increase furosemide to 40mg twice daily."
6. "Restrict sodium intake to ¡2g/day."
7. "Follow up in 1 week to reassess fluid status."
8. Problem 1 completeness
9. Problem: "Hypertension"
10. "Assessment: Blood pressure well controlled on current regimen."
11. "Plan: Continue lisinopril 20mg daily."
12. Problem 2 completeness
13. Note completeness
14. End-of-note

Section 4: Methods - Why This Design Matters

Alignment with Real Clinical Workflows

1. **Problem-Oriented Charting**
   - Standard medical documentation approach
   - Problem list critical for:
     - Medical coding and billing
     - Insurance reimbursement
     - Clinical communication between providers
2. **Prevents Reward Hacking**
   - Without completeness steps: Model might prefer shorter notes
   - Example bad behavior: Omit entire problems to get higher scores
   - Completeness steps enforce thoroughness
3. **Multi-Granularity Verification**
   - **Sentence level**: Detect factual errors, hallucinations
   - **Problem level**: Ensure complete assessment & plan per problem
   - **Note level**: Verify all relevant problems addressed
4. **Explainability**
   - Can identify: "Error in Step 5 of Problem 1"
   - Physicians can quickly locate and fix issues

Section 4: Methods - Error Taxonomy

Four Error Categories (from physician review of LLM outputs)

1. Factual Inaccuracy

- **Definition**: Information referenced in conversation but not supported by content
- **Examples**:
  - Laterality: "left knee pain" → "right knee pain"
  - Dosage: "furosemide 40mg" → "furosemide 80mg"
  - Timeline: "follow up in 1 month" → "follow up in 6 months"
- **Impact**: Can lead to serious clinical consequences

2. Hallucination

- **Definition**: Completely unrelated entities never mentioned in conversation
- **Examples**:
  - Adding "history of diabetes" when never discussed
  - Fabricating test results not mentioned
  - Inventing symptoms patient didn't report
- **Key distinction**: Entirely novel information vs. misrepresentation

Section 4: Methods - Error Taxonomy (continued)

3. Unhelpfulness

- **Definition**: Vague, incomplete, or confusing expressions lacking critical details
- **Examples**:
  - Imprecise: "patient has some issues" vs "patient reports chest pain"
  - Vague timing: "follow up soon" vs "follow up in 2 weeks"
  - Missing specificity: "adjust medication" vs "increase lisinopril to 20mg"
- **Impact**: Reduces clinical utility even if technically accurate

4. Incompleteness

- **Definition**: Missing specific steps or entire problems
- **Implementation**: Randomly remove steps or problems from samples
- **Examples**:
  - Omitting problem from problem list
  - Missing plan for documented assessment
  - No follow-up instructions
- **Rationale**: Ensures model doesn't favor brevity over completeness

Key Point: All four error types based on real failure modes observed in LLM-generated clinical notes

Section 4: Methods - Synthetic Error Generation Process

Goal: Generate realistic errors at scale without human annotation

Step 1: Prompt Engineering

- Carefully designed prompts for Gemini Pro 1.5
- Separate prompt for each error type
- Instructions specify: location (step/problem level), number of errors, output format

Step 2: Error Pool Generation

- Generate 10 unique errors per type per case
- Total: 40 errors per case (10×4 types)
- Output in structured JSON format

Step 3: Manual Quality Control

- Physician co-authors inspect generated errors
- Verify errors are realistic and clinically plausible
- Ensure error severity is appropriate

Step 4: Systematic Injection

- Randomly select steps in gold-reference notes
- Replace with errors from pool

Section 4: Methods - Error Generation Details

Prompt Template Example (Factual Inaccuracy):

"You are provided with a doctor-patient conversation and its corresponding clinical note in JSON format. Your task is to introduce 10 errors into the clinical note.

Error type: Factual Inaccuracy - Introduce detailed factual errors related to information or topics discussed in the conversation but not supported by it. Examples include changing 'left' to 'right', altering medication names, or modifying follow-up timeframe.

For each change, include: error_type, problem_no, step_no, detailed_error, new_content, original_content"

Output Format (JSON):

```
{
  "Errors": [{
    "Error_type": "Factual Inaccuracy",
    "Problem_no": "1", "Step_no": "4",
    "New_content": "Increase furosemide to 80mg twice daily",
    "Original_content": "Increase furosemide to 40mg twice daily"
  }]

}
```

Why This Works:

- Scalable: Generate thousands of samples quickly
- Realistic: LLM understands clinical context
- Controllable: Can adjust error distribution and severity

Section 3: Methods — Process-supervised Data Construction

Base data

- Dialogue-G (majority) + ACI-BENCH training subset
- "Gold-reference" A&P notes follow a physician-recommended best-practice format

Pipeline

1. Transform A&P into step hierarchy (problem + sentences + completeness steps)
2. Use Gemini Pro 1.5 to generate pools of errors per error type (with prompt engineering + physician inspection)
3. Create negative samples by swapping an original step with a synthetic error
4. Add semantic diversity via paraphrases that preserve meaning (replace correct steps with paraphrases)

Outcome: PRM-Clinic dataset: gold samples (+ everywhere) mixed with negatives (localized "-" labels).

Section 3: Methods — PRM-Clinic Dataset (Table 1)

Statistic	Value
Cases (ACI-BENCH)	67
Cases (Dialogue-G)	1,205
Total samples	9,680
Mean samples per case	7.61
Mean errors per negative sample	
Factual Inaccuracy	1.16
Hallucination	1.18
Unhelpfulness	1.19
Incompleteness	1.27
Mean paraphrases per sample	2.38

Notes: Dialogue-G is the majority of instances; training mixes gold and negatives.

Section 4: Methods - Training the PRM

Model Architecture

- **Base**: LLaMA-3.1 8B Instruct
- Uses reserved special tokens for step and score tokens
- No architectural modifications needed

Input Format: `[Dialogue] [Problem 1] <step> [Step 1] <score> + <step> [Step 2] <score> - ...`

Training Objective:

- Cross-entropy loss: $L = -\sum_{i \in I} \log p_\theta(t_i | t_{<i})$
- I = set of token positions to include
- **Best approach**: Notes-Only loss (mask dialogue tokens)

Labels:

- **Gold-reference samples**: All steps get "+"
- **Error-containing samples**:
  - Erroneous step gets "-"
  - Other steps get "+"
  - Completeness steps get "-" if incomplete
  - End-of-note gets "-" if any error present

Section 5: Using PRMs at Inference - The Process

Four-Step Inference Pipeline

Step 1: Generate Candidate Notes

- Use LLM (e.g., LLaMA-Clinic) to generate N notes from same dialogue
- Vary temperature for diversity (0.2 to 2.0)
- Typical: N = 2,000 for physician reader study

Step 2: Compute Step-Level PRM Scores

- Single forward pass: dialogue + note through PRM
- PRM score at step $i = p_\theta(\text{“+”} | \text{context}_{<i})$
- Extract softmax probability of "+" token at each step position

Step 3: Aggregate to Note-Level Score

- **PRM**: Product of all step scores
  - $\text{Score}_{\text{note}} = \prod_{i=1}^{N} \text{PRM}_i$
  - In practice: $\log \text{Score} = \sum_{i=1}^{N} \log \text{PRM}_i$ (numerical stability)
- **ORM**: Use only end-of-note step score

Step 4: Best-of-N Selection

- Select note with highest note-level score

Section 5: Evaluation Tasks

Task	Metric Type	Cases	ID/OOD	Note Source
A-Verify	Verification	80	OOD	LLaMA-Clinic
A-Prefer	Preference	80	OOD	LLaMA-Clinic
Dialogue-G	Verification	80	ID	Gemini Pro
A-Validate	Verification	20	ID	Gemini Pro

ID vs OOD Setting:

- **ID (In-Distribution)**: Notes from Gemini Pro (same model family as training data)
- **OOD (Out-of-Distribution)**: Notes from LLaMA-Clinic (LLaMA-2 13B with domain adaptation)
- OOD tests generalization to different model architectures

Task Details:

- **Verification (A-Verify, Dialogue-G, A-Validate)**: Select gold-reference from error-containing samples
- **Preference (A-Prefer)**: Select physician-preferred note from 3 candidates
  - Most challenging task
  - Uses physician preference labels from RLHF study
  - Captures subjective quality judgments

Section 5: Why PRMs Help - Comparison

PRM vs ORM Aggregation

Aggregation Method	A-Prefer	A-Verify
Product (PRM)	**56.2%**	**98.8%**
Last (ORM)	51.2%	98.8%
Mean	55.0%	98.8%
Min	45.0%	97.5%
Max	42.5%	96.2%

Why Product Works Best:

1. **Fine-grained error detection**: Any single error significantly lowers score
2. **Encourages consistency**: All steps must be high quality
3. **Better preference alignment**: Captures physician's holistic judgment

Additional Benefits of PRMs:

- **Explainability**: Can show physicians exactly which steps scored low
- **Debugging**: Developers can identify systematic error patterns
- **Future work**: Enable MCTS, step-level RL

Section 6: Main Results - Overall Performance

Model	A-Prefer (OOD)	A-Verify (OOD)	Dialogue-G (ID)	A-Validate (ID)
Gemini Pro 1.0	43.8%	46.2%	45.0%	45.0%
Gemini Pro 1.5	50.0%	93.8%	95.0%	90.0%
GPT-4o	41.2%	86.2%	81.2%	90.0%
o1	52.5%	73.8%	66.2%	75.0%
o3-mini	53.8%	86.2%	86.2%	90.0%
Vanilla ORM	37.5%	70.0%	76.2%	85.0%
ORM (Ours)	51.2%	**98.8%**	97.5%	100%
PRM (Ours)	**56.2%**	**98.8%**	**98.8%**	**100%**

Key Numbers to Emphasize:

- **98.8% on A-Verify**: Near-perfect gold-reference detection (OOD)
- **56.2% on A-Prefer**: Best physician-preference alignment (OOD)
- **100% on A-Validate**: Perfect performance on ID verification

Comparison Highlights:

- **vs Gemini Pro 1.5**: +6.2% on A-Prefer, +5% on A-Verify
- **vs o3-mini**: +2.4% on A-Prefer (with 8B vs proprietary model)
- **vs Vanilla ORM**: +18.7% on A-Prefer (massive improvement)

Section 6: Main Results - PRM vs ORM Insights

Same Model Checkpoint, Different Usage

Verification Tasks (A-Verify, Dialogue-G, A-Validate):

- PRM and ORM perform **comparably**
- Both achieve near-perfect accuracy
- Detecting errors is straightforward for both

Preference Task (A-Prefer):

- **PRM: 56.2%**
- **ORM (Ours): 51.2%**
- **Vanilla ORM: 37.5%**
- **PRM advantage: 5%** over same-checkpoint ORM

Critical Insights:

1. **Step-level training helps even for ORM**:
   - ORM (Ours) vs Vanilla ORM: 51.2% vs 37.5% (+13.7%)
   - Learning step-level correctness improves overall judgment
2. **Product aggregation better captures preferences**:
   - Multiplicative scoring ensures consistency
   - Better aligns with physician holistic evaluation

Section 7: Ablation Studies - Loss Functions

Question: Which tokens should be included in loss computation?

Loss Function	PRM (A-Prefer)	ORM (A-Prefer)
All tokens	46.2%	38.8%
Score-Token-Only	37.5%	27.5%
Special-Token	48.8%	42.5%
Notes-Only	**55.0%**	**48.8%**
+ Data Selection	**56.2%**	**51.2%**

Findings:

- **Score-Token-Only**: Severe degradation (37.5%)
  - Model can't learn context without seeing note text
- **Notes-Only**: Best performance (55.0%)
  - Masks dialogue tokens, focuses on clinical documentation
  - Prevents overfitting to repetitive dialogue patterns
- **All tokens**: Baseline (46.2%)
- **Special-Token**: Slight improvement (48.8%)

Interpretation: Model needs to learn to **generate** notes (conditioned on dialogue) while simultaneously predicting correctness

Section 7: Ablation Studies - Data Selection

Question: Does quality filtering help? What about paraphrases?

Data Selection	Samples	PRM (A-Prefer)
High Quality Only	5,854	35.0%
High + Paraphrases	5,854	45.0%
High + Medium	10,094	48.8%
High + Med + Para	10,094	46.2%
All Data + Para	**10,952**	**56.2%**

Findings:

1. **Paraphrases help**: Especially for smaller datasets (+10%)
2. **More data > perfect quality**: Full dataset beats filtered subsets
3. **Best combination**: All data + paraphrases (56.2%)

Hypothesis: When training data is limited, model benefits from diverse examples (including lower-quality ones) to learn robust patterns

Practical Implication: Don't over-filter training data; diversity and quantity can outweigh perfect quality

Section 8: Physician Reader Study - Setup

Research Question: Which metric better predicts downstream Best-of-N performance?

- A-Prefer (physician preference alignment)
- A-Verify (gold-reference detection)

Methodology:

1. **Select PRM checkpoints** with different A-Prefer/A-Verify scores
   - Dual High (both metrics high)
   - Dual Low (both metrics low)
   - High A-Prefer vs Low A-Prefer (matched on A-Verify)
   - High A-Verify vs Low A-Verify (matched on A-Prefer)
2. **Generate candidates**: 2,000 notes per case using LLaMA-Clinic
   - Vary temperature (0.2 to 2.0)
   - Total: 40 cases from ACI-BENCH test1
3. **Select with PRM**: Each checkpoint selects top note via Best-of-2000
4. **Physician review**: 9 physicians evaluate in blinded, randomized manner
   - Pairwise comparisons between checkpoint selections
   - Majority vote determines winner

Section 8: Physician Reader Study - Results

Win Rates (percentage preferring first model):

Comparison	Win Rate
Dual High vs Dual Low	55.6%
High A-Prefer vs Low A-Prefer	50.0%
High A-Verify vs Low A-Verify	47.5%

Key Findings:

1. **Both metrics predict real-world performance**
   - Models strong on either metric outperform weak ones
   - Dual High significantly beats Dual Low (55.6% win rate)
2. **A-Prefer slightly more predictive**
   - 50.0% vs 47.5% win rates
   - Makes intuitive sense - directly measures preference
3. **Optimize for both metrics**
   - Verification accuracy matters for error detection
   - Preference alignment matters for subjective quality
   - Best models excel at both

Practical Implication: For real-world deployment, monitor both verification and preference metrics

Section 9: Limitations

Study Limitations

1. **Limited Physician Pool**
   - Only 9 physicians in reader study
   - All from internal medicine specialty
   - Preferences may not generalize to other specialties
2. **Single Specialty Focus**
   - "Best Practice" format designed for internal medicine
   - Other specialties may have different documentation standards
   - Step definitions may need adjustment for other domains
3. **Proprietary LLM Dependency**
   - Uses Gemini Pro 1.5 for error generation
   - Could potentially use open-source alternatives (e.g., LLaMA-3.1 405B)
   - Methodology is model-agnostic
4. **Modest Preference Performance**
   - 56.2% on A-Prefer is SOTA but leaves room for improvement
   - Many factors influencing physician preferences still unexplored

Section 9: Ethical Considerations & Future Work

Ethical Considerations

- **Child safety**: Cautious about content involving minors
- **Bias amplification**: Synthetic data must not introduce/amplify biases
  - Certain error types disproportionately associated with specific populations
  - Could reinforce inequities in care delivery
- **Validation at scale**: Essential before real-world deployment
- **Reward hacking**: Need safeguards against unintended optimization

Future Directions

1. **Other medical specialties**
   - Adapt step definitions and error taxonomy
   - Surgery, pediatrics, emergency medicine
2. **Expand error taxonomy**
   - Additional error types beyond current four
   - Specialty-specific errors
3. **Inference-time scaling**
   - Monte Carlo Tree Search (MCTS)
   - Beam search with PRM guidance
4. **Step-level reinforcement learning**
   - Use PRM scores as rewards for policy gradient training

Section 10: Conclusions

Main Contributions Summary

1. **Novel Framework**
   - First PRM for clinical text generation
   - Generalizable methodology for domains without ground-truth
2. **Domain-Informed Design**
   - Hierarchical step decomposition based on clinical expertise
   - Error taxonomy from real LLM failure modes
3. **Scalable Supervision**
   - LLM-generated synthetic errors at scale
   - Avoids expensive human annotation
4. **State-of-the-Art Results**
   - 98.8% verification accuracy (OOD)
   - 56.2% preference alignment (OOD)
   - Outperforms proprietary reasoning models with 8B parameters

"PRMs are not about math — they're about structure."