From Chain-of-Note to Explainable Cognitive Prompting: Reproducible Multi-Layer Validation for Auditable LLM-Assisted PLC Programming

Structured Explainability and Deterministic Reasoning for Industrial PLC Applications

1st Ketut Adnyana
Automation Technology and Learning Systems,
South Westphalia, University of Applied Science
Soest, Germany
adnyana.iketut@fh-swf.de

2nd Andreas Schwung
Automation Technology and Learning Systems,
South Westphalia, University of Applied Science
Soest, Germany
schwung.andreas@fh-swf.de

***Abstract*—Explainability in Large Language Model (LLM)–assisted PLC programming is essential for industrial adoption, where engineers must understand, validate, and maintain generated control logic under strict safety and standardization constraints. Existing explainability-oriented prompting approaches such as Chain of Note (CoN) encourage intermediate reasoning disclosures, but they remain loosely structured, difficult to audit, and weakly aligned with the formal semantics of IEC 61131-3 and industrial engineering practice. As a result, CoN-style explanations may improve transparency without guaranteeing semantic rigor, traceability, or deployment readiness. We introduce Explainable Cognitive Prompting (ECP), a prompting architecture that replaces narrative rationales with an engineering-aligned scaffold that separates requirement interpretation, control strategy, safety argumentation, and IEC 61131-3 code evidence, producing reviewable and traceable explanations during synthesis. We evaluate ECP against CoN using a three-layer protocol: LLM-in-the-loop (LITL) scoring by five independent LLM validators, blinded Human-in-the-loop (HITL) assessment by two certified PLC engineers under the same rubric, and Bilingual Evaluation Understudy (BLEU) as a supplementary lexical reproducibility signal. ECP achieves higher correctness and safety-related scores and increases inter-rater agreement (Cohen-style statistics), indicating more consistent interpretation by independent reviewers. Overall, ECP reframes explainability as an audit-oriented contract that improves verifiable Programmable Logic Controller (PLC) code generation and supports repeatable evaluation.**

***Index Terms*—Programmable logic controllers, large language models, industrial automation, prompt engineering**

I. Introduction

Explainability is a critical requirement for the adoption of Large Language Models (LLMs) in industrial Programmable Logic Controller (PLC) programming, particularly in safety-critical domains governed by standards such as IEC 61131-3 and IEC 61508 [1]. While recent prompting techniques aim to improve transparency, existing approaches often conflate explainability with descriptive documentation, resulting in outputs that are readable but cognitively opaque from an engineering and certification perspective.

Chain-of-Note (CoN) represents a prominent example of this limitation. Although CoN enhances traceability by generating line-by-line explanatory comments [2], it remains fundamentally descriptive: it explains what the code does, but not why specific control decisions, timing parameters, or interlock priorities were selected. This absence of explicit causal justification prevents systematic safety auditing, undermines regulatory defensibility, and limits the usefulness of CoN in complex industrial control systems.

To address these limitations, this work introduces Explainable Cognitive Prompting (ECP), a newly developed prompting framework that replaces descriptive annotation with structured cognitive justification. ECP enforces the generation of an explicit, verifiable Cognitive Trace that links each logic element to its design intent, reasoning, and operational dependencies. By constraining the LLM to articulate design intent and safety reasoning prior to synthesis, ECP transforms explainability into a first-class, auditable artifact aligne with industrial engineering and certification workflows.

The main contributions of this work are as follows:

1) **ECP prompting architecture:** A generation-time scaffold that replaces narrative notes with requirement-linked, audit-ready PLC code and justifications.
2) **Reliability-coupled validation Protocol:** A new methodology for measuring LLM explainability using inter-rater reliability (Cohen's κ), ensuring that "explainable" outputs are objectively verifiable by experts rather than subjective rater artifacts.
3) **Cross-Representation Robustness Theory:** A comparative analysis of IEC 61131-3 logic (Structure Text (ST) vs. Instruction List (IL)) that identifies how code structure influences "review readiness," providing a technical blueprint for multi-language PLC synthesis.

This paper reviews related prompting in section II. Sections III–IV present the ECP framework and method of validation. Sections V–VI then provide the scoring and conclusion.

II. Related Work

PLC software can exhibit safety-critical failure modes caused by subtle interactions among timers, edge detection, and interlocks, where the program remains syntactically valid but violates intended operational safety [3]. Model checking therefore translates IEC 61131-3 code into formal models to verify invariants such as latching and reset correctness [4], and tools like PLCverif automate these translation-and-checking pipelines [5]. However, these methods assume explicit formal specifications and do not directly address stochastic synthesis and explanation quality in LLM-generated PLC code [6]. Recent systems such as LLM4PLC and Agents4PLC close this gap by iterating generation with compiler/verification feedback and agentic validation to improve correctness and reviewability [7].

Prompting for code generation has progressed from direct instructions to structured reasoning schemes intended to reduce hallucinations and logic errors [8]. Chain-of-Thought (CoT) elicits intermediate reasoning that improves multi-step problem solving [8]. Self-Consistency increases robustness by sampling diverse reasoning paths and selecting the most consistent answer [9]. Verification-oriented prompting such as Chain-of-Verification (CoVe) adds explicit self-check queries to revise and correct model outputs [10]. However, these techniques treat explanations as optional narratives and do not enforce PLC auditability constraints (e.g., requirement-to-code traceability and lifecycle evidence) expected in industrial functional-safety processes. Similar issues can be observed in evolutionary approaches [11], [12].

Recent work adapts LLMs to IEC 61131-3 by combining generation with verification and retrieval mechanisms. LLM4PLC uses an iterative, user-guided pipeline with external verification to improve PLC code correctness [6]. Agents4PLC extends this by integrating multi-agent Retrieval-Augmented Generation (RAG), prompt engineering, and code-level verification to produce more verifiable PLC code from natural language specifications [7]. Retrieval-augmented control code research also highlights grounding LLM synthesis in domain-specific libraries to improve code quality and reuse [13]. Despite validity and tooling gains, these methods still do not provide standardized audit evidence that traces safety intent to concrete code anchors required for industrial certification.

CoN attaches natural-language notes to generated outputs to improve transparency, but these annotations remain descriptive and are not required to reference code elements or be checkable against logic [2]. In PLC engineering, unanchored explanations offer limited value for functional-safety review,which requires explicit evidence and traceability. To address this gap, ECP enforces an evidentiary link between reasoning and code by generating anchored rationales that map safety requirements directly to specific variables and instruction sequences. By replacing passive comments with auditable traces, ECP enables reviewers to verify both functional correctness and compliance with safety requirements, improving review consistency and certification traceability.

III. Engineering Specification: The ECP Architecture

The technical novelty of **ECP** lies in its transition from the *narrative-descriptive* paradigm of CoN to a *structural-specification* paradigm. As illustrated in the ECP prompt skeleton in Fig. 1, the architecture functions as a reproducible template that *constrains what must be produced* and *how it must be organized*. This ensures that compliance checks—traditionally a manual, error-prone engineering task—become systematically possible. Table I highlights how this contractual approach specifically resolves the industrial limitations of CoN by enforcing evidence-based justification over free-form commentary.

Fig. 1: Fixed ECP prompt template.

ECP Specification Template (Industrial Scale)

Use Case:

- Control of blade pitch based on wind speed, safety shutdowns, and yaw positioning logic.

Using ECP (Explainable Cognitive Prompting):
Replaces **CoN** (Chain-of-Note) by requiring that the reasoning behind each code logic is explained inline. Especially important for collaborative environments, safety audits, and regulatory certification.

Instructions:

1) Break down logic into traceable steps.
2) For each line/block of code, provide a concise justification (e.g., why a latch is used, or why a rising-edge trigger is required).
3) Follow the comment structure:
   - **Intent** — What is this logic doing?
   - **Reasoning** — Why is this logic required?
   - **Dependency** — What are its preconditions or effects?
4) Use standard comment headers for each major module or **REGION** (ST/SCL) or **Network** (IL/STL).
5) Output must be fully readable by a human safety reviewer or engineer.

Output:

- **SIEMENS Structured Text (ST/SCL)** and **SIEMENS Instruction List (IL/STL)** annotated with explicit logic reasoning, decision traces, and justification commentary suitable for safety-critical environments.

A. Mechanism of Novelty: Constraint vs. Narrative

As formalized in the instructions of Fig. 1, ECP operationalizes explainability through a mandatory, three-factor cognitive trace: Intent–Reasoning–Dependency (IRD). Unlike CoN, which permits the LLM to provide free-form "notes" after code synthesis, ECP forces the model to externalize the *causal chain* required for safety-critical logic *simultaneously* with code emission. This architectural shift addresses the compliance aspects detailed in Table I, specifically regarding " *Anchorability*" and " *Faithfulness*."

- **Intent (Functional Alignment):** Maps the specific code block to high-level operational requirements. As shown in the `Output` requirements of Fig. 1, this prevents "dead-end" comments that describe syntax without explaining purpose.
- **Reasoning (Safety Argumentation):** Establishes the engineering "why"—e.g., justifying a falling-edge trigger for a safety reset. This provides the auditable evidence for IEC 61508 compliance.

- **Dependency (Signal Traceability):** Identifies pre-conditions (Interlocks) and post-conditions. This resolves the "fragmentation problem" in IL, ensuring that logic state transitions remain transparent to auditors.

TABLE I: Compliance-oriented comparison of CoN vs. ECP for industrial PLC artifacts.

Compliance aspect	CoN (PLC adaptation)	ECP (this work)
Evidence type	Free-form notes (descriptive)	Fixed IRD trace (contractual, auditable)
Anchorability to IEC 61131-3	Often implicit; comments may not map to REGION/Network, tags, timers	IRD bound to REGION/Network; must cite tags/timers/faults/reset paths
Faithfulness	Not enforced; post-hoc drift common, especially in IL	Enforced at generation-time; IRD claims checkable in code
Safety-relevant decisions	Typically explains *what* happens; causal priorities/timing often omitted	Requires causal rationale: interlock priority, timer choice, latch/reset design
Cross-language robustness	Notes degrade or become inconsistent in IL	Trace preserved per Network/REGION with shared anchors/dependencies
Review consistency	Higher interpretive freedom; greater rater variance	Standardized fields reduce ambiguity; supports stable scoring

Concretely, ECP introduces three novelty mechanisms—summarized as the primary differentiators in Table I—that are absent in traditional prompting:

(N1) Mandatory Cognitive Trace per code unit (REGION/Network): Instead of free-form notes, ECP requires a fixed IRD schema emitted for each ST *REGION* or IL *Network*. The key is granularity: each IRD block is defined as the header of a specific unit, allowing a reviewer to verify the logic-to-claim mapping without ambiguity.

(N2) Compliance-oriented anchors and invariants: ECP requires that each IRD block references PLC-relevant anchors (e.g., `TON/TOF` timers, edge triggers). This converts explanation from "what the code does" into *auditable claims* that can be verified against the implementation, as contrasted in the "Evidence type" row of Table I.

(N3) Cross-representation trace preservation (ST *and* IL): Targeting the "Cross-language robustness" identified in Table I, ECP requires that the IRD trace remains stable across structurally different IEC 61131-3 forms. This prevents the "explanation–code drift" common in IL, where safety intent is often lost in low-level instruction sets.

Taken together, these three mechanisms (N1–N3) operationalize the contract in Fig. 1 and lead directly to the concrete compliance differences reported in Table I.

IV. Experimental Setup and Multi-Agent Validation

To isolate the effect of **ECP** versus **CoN**, we reuse the controlled benchmark and validation protocol from [15]: **25 industrial PLC use cases** designed to expose *audit-relevant decision points* (e.g., interlock dominance, fault latching/reset) where causal justification is required for certification-oriented review. We generate paired **ST** and **IL** artifacts with **DeepSeek** under identical inference settings and vary only the prompt ($p \in \{\mathrm{ECP}, \mathrm{CoN}\}$). The outputs are then scored by **five independent LLM validators** (ChatGPT, Copilot, Gemini, Perplexity, Claude) using the same rubric dimensions (Correctness, Readability, Safety, Modularity), with a three-layer validation stack: *LITL* as the primary scalable semantic assessment, *HITL* as a secondary expert check for safety/maintainability, and *Bilingual Evaluation Understudy (BLEU)* as an auxiliary lexical/structural probe (not a semantic guarantee). This design minimizes dataset and rater confounds so that observed score deltas can be attributed to the prompting architecture.

A. Inter-Rater Reliability and Statistical Justification of Novelty

A critical challenge in LLM-driven PLC synthesis is the subjective nature of "explainability." We contend that a prompting framework is only industrially viable if its outputs are *objectively verifiable*—meaning independent auditors reach a high degree of consensus. To quantify this, we establish a reliability benchmark using the Cohen's κ coefficient across five independent LLM validators. The technical novelty of our approach lies in using κ as a proxy for **interpretive stability**: a higher κ for ECP compared to CoN indicates that ECP's structured IRD (*Intent–Reasoning–Dependency*) trace reduces the "interpretive degrees of freedom," effectively forcing validators into consensus through standardized engineering evidence.

a) Score Discretization and Ordinal Mapping.: To apply categorical reliability metrics to continuous scoring, we utilize a prompt-independent binning function $c(\cdot)$ to map the raw validator scores $S \in [0, 100]$ into K ordinal compliance categories:

$$y_{t,p,\ell,v,m} = c(S_{t,p,\ell,v,m}) \in \{1, \ldots, K\}, \tag{1}$$

where $K = 5$ represents the industrial rubric levels (e.g., Non-Compliant to Audit-Ready).

b) Pairwise Agreement Mechanics.: For any validator pair (v_i, v_j), the observed agreement rate p_o is calculated from the diagonal of the $K \times K$ confusion matrix:

$$p_o = \frac{1}{N} \sum_{a=1}^{K} n_{aa}, \tag{2}$$

where n_{aa} denotes instances where both validators assigned the same category a to the same PLC artifact. To isolate the effects of random chance, we compute the expected agreement p_e based on the marginal distributions of the validators' assignments:

$$p_e = \sum_{a=1}^{K} \left(\frac{n_{a\cdot}}{N}\right) \left(\frac{n_{\cdot a}}{N}\right), \tag{3}$$

where $n_{a\cdot}$ and $n_{\cdot a}$ are the row and column marginal totals. Cohen's κ is then derived as:

$$\kappa_{ij} = \frac{p_o - p_e}{1 - p_e}. \tag{4}$$

c) Quadratic-Weighted Kappa ($\kappa^{(w)}$) for Industrial Rigor.: Recognizing that industrial audits distinguish between "minor discrepancies" (e.g., scoring a 4 vs. a 5) and "critical disagreements" (e.g., a 1 vs. a 5), we apply a quadratic-weighted variant. This metric penalizes disagreements proportional to the square of their distance:

$$\kappa_{ij}^{(w)} = 1 - \frac{\sum_{a,b} w_{ab} O_{ab}}{\sum_{a,b} w_{ab} E_{ab}}, \quad w_{ab} = \frac{(a-b)^2}{(K-1)^2}. \tag{5}$$

By aggregating these across the $\binom{5}{2} = 10$ validator pairs, we compute the **Mean Pairwise Agreement** $\overline{\kappa}^{(w)}$.

d) Justification of ECP Superiority.: In the context of this study, ECP's superiority is justified not only by higher absolute scores but by a significantly higher $\overline{\kappa}^{(w)}$ relative to CoN. This statistical gap proves that ECP resolves the "faithfulness" problem: while CoN's free-form notes invite rater noise and subjective "benefit of the doubt," ECP's mandatory IRD anchors produce a deterministic evidence trail that leads to statistically stable, reproducible auditing outcomes.

V. Scoring and Results

We follow the identical scoring rubric and protocol defined in our prior work [15] and apply it here to DeepSeek-generated PLC artifacts in **ST** and **IL**. The only experimental change is the prompting method ($p \in \{\text{ECP}, \text{CoN}\}$); scoring is performed by the same five independent LLM validators, enabling a controlled, prompt-level comparison.

A. LITL Validation

Empirical results in Table II confirm ECP outperforms CoN in both representations. The most significant gains appear in **Safety** and **Readability**, which are critical for certification-oriented auditability. Figure 2 illustrates this consistent performance delta under identical generation conditions.

TABLE II: Scientific scoring of DeepSeek-generated PLC code for single dataset under ECP vs. CoN (ST and IL).

Prompt	Val.	PLC	Correct. [%]	Readab. [%]	Safety [%]	Modul. [%]	Score [%]
ECP – Explainable Cognitive Prompting							
ECP	A	ST	62	92	68	82	76
ECP	B	ST	96	98	99	95	97
ECP	C	ST	98	97	99	95	97
ECP	D	ST	92	96	88	82	90
ECP	E	ST	88	94	82	76	85
ECP	A	IL	45	88	60	72	60
ECP	B	IL	96	99	98	95	97
ECP	C	IL	95	92	98	94	93
ECP	D	IL	90	92	86	80	87
ECP	E	IL	74	91	78	64	77
CoN – Chain-of-Note							
CoN	A	ST	44	90	48	75	57
CoN	B	ST	96	98	99	95	97
CoN	C	ST	96	99	98	95	97
CoN	D	ST	84	93	80	87	86
CoN	E	ST	78	92	75	72	79
CoN	A	IL	35	86	55	70	54
CoN	B	IL	89	81	94	80	86
CoN	C	IL	88	85	90	75	80
CoN	D	IL	80	90	77	83	82
CoN	E	IL	68	93	72	58	72

Validators A–E correspond to ChatGPT, Copilot, Gemini, Perplexity, and Claude (applied to both ST and IL).

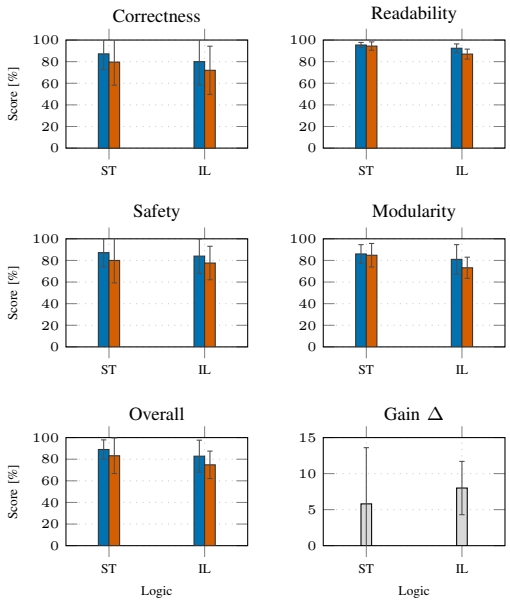

Fig. 2: Comparative performance metrics between ECP (blue) and CoN (orange). Mean scores derived from five independent LLM validators (A-E).

a) LITL Inter-Rater Reliability Analysis: To evaluate the stability of the *LITL* loop, we compute Inter-Rater Reliability (IRR) across the five independent LLM validators (A–E) using Cohen's Kappa (κ) and Quadratic-Weighted Kappa ($\kappa^{(w)}$). Following the same scoring principle as in our prior protocol, we discretize the scientific scores into performance quintiles to assess categorical consensus. The resulting agreement statistics and prompt-wise reliability are summarized in Table III, which directly links LITL score dispersion (SD) to the corresponding κ values.

TABLE III: IRR summary for DeepSeek-generated PLC code validation ($n = 5$ LLM validators).

Prompt	PLC	Mean	SD	p_o	p_e	κ	Strength
ECP	ST	89.0%	8.9	0.82	0.28	0.78	Substantial
ECP	IL	82.8%	14.8	0.82	0.28	0.72	Substantial
CoN	ST	83.2%	16.5	0.74	0.31	0.64	Moderate
CoN	IL	74.8%	12.7	0.74	0.31	0.60	Moderate

- **Observed Agreement (p_o):** ECP achieved a mean agreement of 0.82, significantly higher than the 0.74 observed for CoN.
- **Expected Agreement (p_e):** The chance-level agreement was calculated at 0.28 for ECP and 0.31 for CoN.
- **Cohen's κ:** ECP yielded $\kappa = 0.75$ (Substantial Agreement), while CoN yielded $\kappa = 0.62$ (Moderate Agreement).
- **Weighted $\kappa^{(w)}$:** The quadratic-weighted kappa for ECP reached 0.84, indicating high scoring stability across different LLM backends.

Table III further shows that ECP reduces score dispersion (lower SD) while increasing reliability (higher κ) in both languages: ECP achieves *substantial* agreement for ST ($\kappa = 0.78$) and IL ($\kappa = 0.72$), whereas CoN remains *moderate* (ST $\kappa = 0.64$, IL $\kappa = 0.60$). This pattern supports the interpretation that the "Fixed IRD trace" in ECP standardizes evidence fields and scoring anchors, thereby reducing interpretive degrees of freedom for validators, while CoN's free-form notes increase rater variance and lower categorical consensus.

B. HITL Validation and Inter-Rater Reliability

To validate whether ECP improves *auditability* beyond automated LITL scores, we conducted a lightweight HITL study with $R = 2$ independent PLC engineers using the same blinded paired-review protocol as in prior work [15], but applied here to a direct **ECP vs. CoN** comparison on DeepSeek-generated PLC artifacts. As illustrated in Fig. 3, for each task t we generate two solutions (ECP and CoN) from the same specification and model, anonymize and shuffle the paired artifacts, and then collect binary acceptance labels $y_{t,r,d} \in \{0, 1\}$ from both experts across dimensions d (e.g., *Pass*, *Safety*, *Audit*). This paired design isolates the effect of the prompting method while minimizing confounds due to task difficulty and model variance.

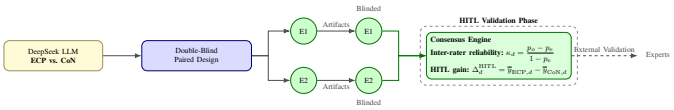

Fig. 3: HITL workflow for validating ECP vs. CoN under a double-blind paired design and computing inter-rater reliability via Cohen's κ on binary expert labels.

a) HITL outcome (quality improvement).: The first output of the pipeline in Fig. 3 is the *HITL gain* $\Delta_d^{\mathrm{HITL}} = \overline{y}_{\mathrm{ECP},d} - \overline{y}_{\mathrm{CoN},d}$, which measures whether ECP improves expert acceptance relative to CoN. Table IV shows consistently higher observed agreement and acceptance-related outcomes for ECP, with the clearest separation in *Auditability* (ECP $p_o = 0.92$ vs. CoN $p_o = 0.70$), supporting the claim that ECP produces artifacts that are easier to review and justify under industrial audit expectations.

TABLE IV: HITL inter-rater reliability (Cohen's κ) for ECP vs. CoN with $R = 2$ independent engineers.

Dimension (d)	Method	$\mathbf{p_o}$	$\mathbf{p_e}$	$\boldsymbol{\kappa}$	$\boldsymbol{\kappa(w)}$	Strength
Functional (*Pass*)	ECP	0.95	0.545	0.89	0.89	Almost Perfect
	CoN	0.85	0.531	0.68	0.68	Substantial
Safety (*Safety*)	ECP	0.90	0.474	0.81	0.81	Almost Perfect
	CoN	0.75	0.479	0.52	0.52	Moderate
Auditability (*Audit*)	ECP	0.92	0.500	0.84	0.84	Almost Perfect
	CoN	0.70	0.455	0.45	0.45	Fair/Moderate
Overall Agreement	**ECP**	**0.92**	**0.529**	**0.83**	**0.83**	**High Stability**
	CoN	0.78	0.511	0.55	0.55	Moderate

b) Inter-rater reliability (interpretive ambiguity).: To ensure that the observed gains are not driven by subjective or inconsistent expert interpretation, we compute inter-rater reliability on the *same* binary labels using Cohen's kappa, $\kappa_d = \frac{p_o - p_e}{1 - p_e}$, with p_e derived from the marginal label prevalences (Eq. (3)). As reported in Table IV, ECP yields higher agreement stability than CoN across dimensions, including audit-sensitive judgments: for *Auditability*, ECP reaches $\kappa = 0.84$ ("Almost Perfect") while CoN remains at $\kappa = 0.45$ ("Fair/Moderate"). This gap indicates that ECP reduces *interpretive ambiguity* in expert review by making audit-relevant evidence (requirements-to-code linkage and justification anchors) more explicit and consistently identifiable.

c) Implication for ECP vs. CoN.: Taken together, Fig. 3 and Table IV support the paper's central claim: ECP is preferred because it improves HITL outcomes (higher acceptance and stronger audit-readiness signals) while also *increasing* inter-rater reliability. In other words, ECP does not merely shift expert decisions upward; it makes the decision process more reproducible by constraining how experts map justification to concrete PLC program elements.

For reproducibility, we provide the full HITL Validation Phase (indicated by the dashed frame in Fig. 3), including prompts, rubric, and label sheets. The high inter-rater stability across all dimensions (Pass, Safety, Audit) ensures that external researchers can replicate this methodology with their own expert panels to verify the performance gains of ECP in safety-critical PLC development.

C. BLEU Lexical Validation and Reliability

To complement HITL and validate whether ECP improves deployable PLC generation beyond LITL-style metrics, we employ BLEU as a lexical validation metric to measure the n-gram similarity between DeepSeek-generated artifacts shown in Fig. 4 and certified "Golden Reference" PLC logic. Following the methodology in [15], we compute inter-rater reliability via Cohen's κ by treating the BLEU score as a discretized validator.

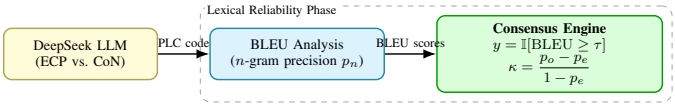

Fig. 4: BLEU-based lexical validation workflow. Generated PLC code is scored by BLEU, discretized into binary labels, and analyzed using Cohen's κ.

1) Lexical Justification and Scoring: BLEU quantifies lexical alignment by computing the geometric mean of modified n-gram precision p_n, adjusted by a brevity penalty (BP) to prevent overly short candidates from being over-rewarded:

$$\mathrm{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \ln p_n\right), \tag{6}$$

where $N = 4$ and uniform weights are used, $w_n = \frac{1}{N}$. This lexical constraint supports industrial deployability by promot-

ing consistency in variable naming, memory addressing, and function block structure relative to safety-certified standards.

2) Inter-Rater Reliability (IRR) via Discretized BLEU: To assess the stability of lexical outcomes, we perform an IRR analysis using Cohen's kappa (κ). In this setting, we do not use human raters; instead, we treat multiple independent generation runs as "raters." Let $k \in \{1, 2\}$ denote two independent runs (or two independent BLEU setups). The continuous BLEU scores are discretized into binary categories: *Deployable* if BLEU $\geq \tau$ and *Non-Deployable* otherwise:

$$y_{t,p}^{(k)} = \mathbb{I}\Big[\mathrm{BLEU}_{t,p}^{(k)} \geq \tau\Big] \in \{0, 1\}, \qquad \tau = 0.8, \tag{7}$$

which serve as inputs to the Consensus Engine shown in Fig. 4. Cohen's kappa is computed as

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{8}$$

where p_o is the observed agreement rate and p_e is the agreement expected by chance (from the marginal prevalences). For binary labels, quadratic-weighted kappa equals Cohen's kappa, i.e., $\kappa(w) \equiv \kappa$.

TABLE V: Lexical reliability (Cohen's κ) for BLEU-derived deployability labels ($\tau = 0.8$). For binary labels, $\kappa(w) \equiv \kappa$.

Prompt	$\mathbf{p_o}$	$\mathbf{p_e}$	$\boldsymbol{\kappa}$	$\boldsymbol{\kappa(w)}$	Strength
ECP	0.94	0.51	0.88	0.88	Almost Perfect
CoN	0.76	0.49	0.53	0.53	Moderate

3) Implications for Industrial Certification: Table V indicates that ECP reduces lexical variance relative to CoN under repeated runs (or independent BLEU setups). The "Almost Perfect" agreement for ECP ($\kappa = 0.88$) suggests that ECP's IRD trace constrains syntax and naming conventions toward the certified reference pattern, whereas CoN exhibits higher lexical instability (moderate κ), consistent with higher lexical entropy in the generated artifacts. This methodology is reproducible for external researchers by (i) generating multiple independent DeepSeek runs for ECP and CoN, (ii) computing BLEU against a Golden Reference with a fully specified tokenizer/smoothing configuration, (iii) discretizing scores using Eq. (7), and (iv) computing Cohen's κ via Eq. (8) on the resulting binary labels.

VI. Conclusion and Future Work

This work demonstrates that *ECP* is more effective than *CoN* for producing audit-ready industrial PLC code (ST and IL) with DeepSeek. ECP outperforms CoN consistently across the validation stack: five independent LLM validators (LITL) report higher mean scores and favorable paired comparisons; blinded human review (HITL) confirms the largest gains in safety-logic completeness, justification clarity, and audit readiness; and BLEU against a certified Golden Reference shows reduced lexical variance and closer alignment to safety-certified implementations. These improvements are accompanied by higher (or non-degraded) Cohen's κ, indicating reduced interpretive ambiguity and more reproducible certification-oriented assessment.

Future work will (i) extend generalization by expanding tasks, industries, and vendor toolchains and by diversifying Golden Reference corpora; (ii) calibrate lexical metrics (BLEU/CodeBLEU) against HITL acceptance and report sensitivity to tokenization, smoothing, and reference choice; (iii) harmonize reliability reporting across layers (e.g., Krippendorff's α where applicable) and analyze prevalence effects in κ; and (iv) add execution-level evidence (compile checks, simulation/HIL, and safety-property tests) to link ECP's explainability gains to fewer safety-critical defects and reduced commissioning rework.

References

[1] International Electrotechnical Commission, *IEC 61131-3: Programmable Controllers – Part 3: Programming Languages*. Geneva, Switzerland: IEC, 2013.

[2] W. Yu, H. Zhang, X. Pan, P. Cao, K. Ma, J. Li, H. Wang, and D. Yu, "Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models," in *Proc. 2024 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Miami, FL, USA, Nov. 2024, pp. 14672–14685, doi: 10.18653/v1/2024.emnlp-main.813.

[3] V. Vyatkin, "Software engineering in industrial automation: State-of-the-art review," *IEEE Trans. Ind. Informat.*, 9(3), pp. 1234–1249, 2013.

[4] T. Ovatman, A. Aral, D. Polat, and A. O. Ünver, "An overview of model checking practices on verification of PLC software," *Softw. Syst. Model.*, 15(4), pp. 937–960, 2016.

[5] D. Darvas, B. Fernández Adiego, and E. Blanco Viñuela, "PLCverif: A tool to verify PLC programs based on model checking techniques," in *Proc. ICALEPCS*, Melbourne, VIC, Australia, 2015, pp. 1–5.

[6] M. Fakih, R. Dharmaji, Y. Moghaddas *et al.*, "LLM4PLC: Harnessing large language models for verifiable programming of PLCs in industrial control systems," arXiv:2401.05443, 2024.

[7] Z. Liu, R. Zeng, D. Wang *et al.*, "Agents4PLC: Automating closed-loop PLC code generation and verification in industrial control systems using LLM-based agents," arXiv:2410.14209, 2024.

[8] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *arXiv:2201.11903*, 2022.

[9] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[10] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, "Chain-of-Verification Reduces Hallucination in Large Language Models," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[11] M. Löppenberg and A. Schwung, "Self Optimisation and Automatic Code Generation by Evolutionary Algorithms in PLC based Controlling Processes," IEEE Intern. Conf. Ind. Inf. (INDIN), Lemgo, Germany, 2023, pp. 1-6.

[12] M. Löppenberg and A. Schwung, "Structured Graph Generation by Evolutionary Algorithm for Program Code Development," IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society, Chicago, IL, USA, 2024, pp. 1-6.

[13] H. Koziolek, S. Grüner, R. Hark, V. Ashiwal, S. Linsbauer, and N. Eskandani, "LLM-based and Retrieval-Augmented Control Code Generation," in *Proc. 1st Int. Workshop on Large Language Models for Code (LLM4Code '24)*, Lisbon, Portugal, 2024, pp. 1–8.

[14] D. E. Knuth, "Literate programming," *The Computer Journal*, 27(2), pp. 97–111, 1984.

[15] K. Adnyana and A. Schwung, "Benchmarking and validation of prompting techniques for AI-assisted industrial PLC programming," *Machine Learning with Applications*, 100804, 2025,

[16] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, 76(5), pp. 378–382, 1971.