1 Methodology

Our methodology integrates data curation, knowledge graph construction, advanced language model fine-tuning, retrieval augmentation, inference optimization, and engineering validation to create specialized and efficient Small Language Models (SLMs) for chemical process engineering tasks, specifically PFD/PID interpretation, analysis, and generation.

The pipeline begins with the curation of the ChemAtlas database (1,020+ common chemicals) from manufacturer catalogs and using an agentic web navigation framework that autonomously retrieves, interprets, and synthesizes multimodal data from diverse web sources to generate process flow and instrumentation descriptions of industrial production processes.

This structured information is then used to construct a Knowledge Graph (KG), where text chunks are processed by GPT-4o to extract semantic triples (subject-predicate-object), entities are canonicalized based on high semantic (embedding) and string (Levenshtein) similarity, and the graph is partitioned into hierarchical communities using the Leiden algorithm to optimize modularity for efficient retrieval.

Leveraging ChemAtlas, our curated database of 1,020 industrial chemicals, we employ advanced teacher LLMs (GPT-4o and Anthropic Claude Haiku) to generate and cross-validate over 20,000 synthetic QA pairs through self-instruct bootstrapping, initiated from a small seed set of human-written examples. These QA pairs are rigorously scored, validated, and filtered using NVIDIA's Nemotron-4-340B to ensure quality before training smaller language models (SLMs).

The resulting datasets comprise six specialized subsets:

1. Factual QA for core process engineering fundamentals;
2. SynDIP datasets comprising comprehensive process contexts (reactions, operating conditions, control strategies), process flow and instrumentation descriptions (equivalent to agentic web retrieved knowledge on industrial chemical production but generated from LLMs pretrained knowledge);
3. LogiCore with multi-step reasoning chains for design justification, control logic validation, etc.;
4. DPO containing preference-ranked pairs for alignment tuning across process engineering fundamentals;
5. Local RAIT (document-grounded QA using individual SynDIP technical documents), and Global RAIT (cross-document synthesis QA requiring integration of multiple SynDIP sources) to enhance retrieval-augmented capabilities for complex engineering tasks.

Additionally, we developed a 1.5K QA out-of-distribution benchmark using ChemAtlas to evaluate generalization performance on QA tasks, along with the ChemEval dataset (100 novel chemicals) to test zero-shot generation of complete

process descriptions including process flow and instrumentation descriptions for unseen chemicals.

Base SLMs, specifically Llama-3.2-1B and SmolLM-135M, are subsequently customized using QLoRA (Quantized Low-Rank Adaptation) using 4-bit NF4 precision with frozen base weights. Two primary fine-tuning strategies are employed on the synthetic datasets:

1. A sequential pipeline involving Supervised Fine-Tuning (SFT) on Factual QA, SynDIP, and LogiCore datasets, followed by Direct Preference Optimization (DPO) on custom DPO datasets, and concluding with Retrieval-Augmented Instruction Tuning (RAIT) on Local/Global RAIT datasets;

2. A reinforcement learning approach adapting Group Relative Policy Optimization (GRPO), applied sequentially first on SFT datasets and then further refined on RAIT datasets, utilizing a composite reward function (combining ROUGE-L, length penalty, and LLM quality score) and KL divergence regularization for stability.

The fine-tuned SLMs are integrated with the structured KG via Graph Retrieval-Augmented Generation (Graph RAG). During inference, relevant graph communities are retrieved based on query similarity to pre-computed summaries; a dynamic subgraph containing entities, relationships, and source chunks is constructed; and this context is passed to the SLM for grounded, multi-hop reasoning.

To enhance performance, a suite of inference optimization and reliability techniques is implemented: structural pruning (width and depth) guided by importance heuristics reduces model size; PagedAttention combined with KV cache quantization mitigates memory fragmentation and reduces cache footprint; Lookahead Decoding accelerates generation latency through parallel token speculation; FlashAttention optimizes the core attention computation to reduce memory bandwidth bottlenecks; and Test-Time Inference Scaling improves output reliability using self-consistency sampling, confidence-weighted entropy scoring, iterative self-reflection/revision, and consensus aggregation.

Finally, the practical engineering feasibility of generated PFD/PID descriptions is validated using the DWSIM open-source chemical process simulator, where PFDs are translated into flowsheets to verify material/energy balances and thermodynamic consistency, while PIDs are functionally assessed by implementing control loops in DWSIM's dynamic environment to evaluate stability and control performance (e.g., setpoint tracking, disturbance rejection).

This comprehensive methodology ensures the developed SLMs are knowledgeable, aligned with engineering principles, efficient, and reliable for practical deployment.

Figure visually outlines the overall architecture. Part (a) depicts the SLM fine-tuning pipeline, showing the progression from a general pre-trained model through initial preference alignment (DPO), followed by task-specific fine-tuning via either instruction tuning or reinforcement learning (GRPO), and concluding

with optional model compression (pruning). Part (b) illustrates the operational Retrieval-Augmented Generation (RAG) framework.

It shows how a user query, after passing guardrails, is processed by a Meta-Agent. This agent utilizes the specialized SLM developed in part (a) as its core reasoning engine. The SLM, guided by the Meta-Agent, retrieves necessary context by accessing both a Memory database (e.g., for conversational history) and a Graph database (containing structured process knowledge). This retrieved information informs the SLM's response generation.

The inference process of the SLM is further enhanced by integrated optimizations (FlashAttention, Paged Attention KV Caching, Lookahead Decoding, Test-Time Scaling). An initial response generated by the SLM undergoes evaluation by a Critique-Agent, utilizing feedback mechanisms (SLM-as-a-judge, Reward Model-as-a-judge, or Human-as-a-judge) to potentially trigger refinement before producing the final Output.

In summary, our integrated framework leverages knowledge graph-based retrieval augmentation, domain-specific SLM fine-tuning pipelines, comprehensive inference optimizations, and feedback-driven refinement. This approach yields robust performance on complex reasoning tasks and demonstrates effective generalization via the generation of plausible, simulator-validated process descriptions for previously unseen chemicals.