Helios: A Foundational Language Model for Smart Energy Knowledge Reasoning and Application

Haoyu Jiang[1,†] **Fanjie Zeng**[2,†] **Boan Qu**[2,†] **Xiaojie Lin**[1,*] **Wei Zhong**[1,3]

[1] College of Energy Engineering, Zhejiang University
[2] Polytechnic Institute, Zhejiang University
[3]Shanghai Institute for Advanced Study, Zhejiang University

{haoyu.jiang, zengfanjie, boan.qu, xiaojie.lin, wzhong}@zju.edu.cn https://helios-llm.github.io/

Abstract

In the pursuit of carbon neutrality, smart energy systems integrating renewable energy, storage, and demand response have become central to energy system transformation. However, fragmented and rapidly evolving interdisciplinary knowledge increases the cognitive and information integration burden for operational decision-making. Although large language models (LLMs) have shown promise in smart energy tasks through fine-tuning or prompt engineering, their lack of domain knowledge and physical constraints often results in semantically plausible but physically inconsistent outputs, limiting their engineering reliability. To address these challenges, we introduce **Helios**, the first large language model tailored to the smart energy domain, together with a comprehensive suite of resources to advance LLM research in this field. Specifically, we develop **EnerSys**, a multi-agent collaborative framework for end-to-end dataset construction, through which we produce: (1) the first smart energy knowledge base, **EnerBase**, to enrich the model's foundational expertise; (2) the first instruction tuning dataset, **EnerInstruct**, to strengthen performance on domain-specific downstream tasks; and (3) the first Reinforcement Learning from Human Feedback (RLHF) dataset, **EnerReinforce**, to align the model with human preferences and industry standards. Leveraging these resources, Helios undergoes large-scale pretraining, instruction tuning, and RLHF. We also release **EnerBench**, the first benchmark for evaluating LLMs in smart energy scenarios, and demonstrate that our approach significantly enhances domain knowledge mastery, task execution accuracy, and alignment with human preferences.

1 Introduction

Driven by the global pursuit of carbon neutrality, smart energy systems must enhance overall efficiency through the intelligent coordination of renewable energy integration, energy storage dispatch, and demand-side response (Lund et al., 2017). Smart energy is highly interdisciplinary, encompassing power engineering, information science, economics, and other fields, and its knowledge base is fragmented and rapidly evolving (Ceglia et al., 2020; Thellufsen et al., 2020). Building on recent advances in general large language models (LLMs) in semantic understanding, logical reasoning, and multitask generalization, a growing body of research has used fine-tuning and prompt engineering to adapt LLMs to task-specific applications in smart energy, such as load forecasting (Jin et al., 2023; Liao et al., 2025; Hu et al., 2025), building energy consumption modeling (Wang et al., 2025b; Jiang et al., 2024), and HVAC fault diagnosis (Zhang et al., 2025), thereby supporting case modeling and intelligent decision making.

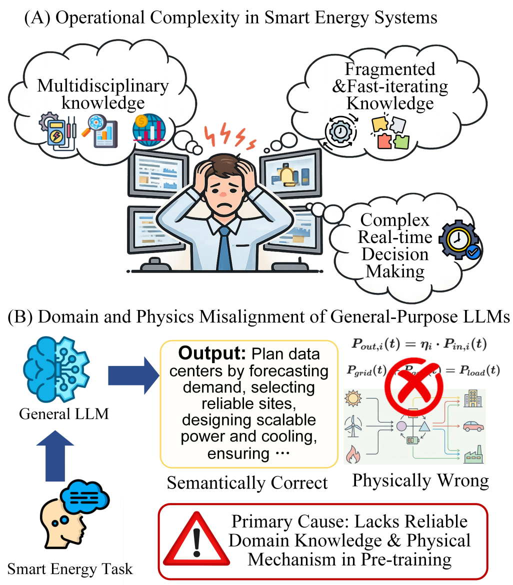

Figure 1: Limitations of Scheduling Decision-Making in the Smart Energy Domain.

However, general LLMs often deliver reasoning that is semantically plausible yet physically

[†] These authors contributed equally to this work.
[*] Corresponding author.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics Volume 1: Long Papers, pages 3208–3220
March 24-29, 2026

invalid (Friel and Sanyal, 2023). This limitation arises chiefly because their pre-training corpora lack reliable knowledge from the smart energy domain, leaving the models without essential domain context and physical constraints (Deng et al., 2024). Current approaches (Hu et al., 2025; Zhang et al., 2025) mainly invoke the prior knowledge already embedded in LLMs and do not explicitly enrich them with smart energy expertise. To alleviate these challenges, we introduce the first-ever open-sourced foundational LLM for the smart-energy domain, referred to as **Helios** (Originating from the ancient Greek sun-god, signifies the illumination of the pathway toward sustainable development through the radiance of smart energy, thereby advancing the harmonious co-existence of humanity and the natural environment). Helios is capable of effectively tackling a broad spectrum of smart-energy tasks. Furthermore, we present **EnerSys**, an end-to-end multiagent collaborative framework for dataset construction that integrates automated data generation, screening, and refinement, thereby furnishing Helios with an extensive and high-quality data foundation.

EnerSys covers three dataset-construction phases (as shown in Fig. 2): In the construction of the pre-training dataset, the Parsing-Agent and Deduplication Agent extract structured knowledge from the Smart Energy Corpus (scientific papers, domain-modeling code, IEA datasets, etc.) and eliminate redundancy, building a comprehensive, balanced smart energy domain knowledge base, **EnerBase**; In instruction-tuning dataset construction, on expert-crafted seed data, we deploy Expert-Agents for each of 14 smart-energy sub-domains, letting them generate instruction–response pairs from the seeds and a high-quality corpus; the Check-Agent then scores samples on accuracy, completeness, relevance, and usability, and the Refine-Agent automatically fixes those below par. This pipeline yielded the **EnerInstruct**; In the RLHF dataset construction, agents like Write-like-Human craft multi-level candidate answers to given questions, thereby creating the **EnerReinforce** to supply the reward model with differentiated contrastive samples. Using these datasets, we complete Helios pre-training (adding domain basics), supervised fine-tuning (boosting downstream skills), and RLHF reinforcement (aligning with human preferences). Concurrently, adhering to a dual-track paradigm of "public item-bank retrieval + expert-targeted design," we build **EnerBench**, containing 625 subjective and 887 objective questions, to systematically assess LLMs performance in smart-energy scenarios. Experiments show Helios surpasses general-purpose LLMs on both tasks, with output style tightly matching professional context. Our contributions can be summarized as follows:

1) We design Helios, the first foundation large language model in the smart-energy domain; it effectively tackles a wide range of smart-energy tasks and produces outputs that are deeply consistent with professional discourse.

2) We propose EnerSys, an end-to-end, multi-agent collaborative framework for dataset construction, through which we develop a domain knowledge base, an instruction-tuning dataset, and an RLHF database for smart energy. In addition, we release Smart Energy Bench, a benchmark that systematically evaluates LLMs' comprehensive performance in smart-energy scenarios.

3) Relative to general LLMs, Helios delivers superior results on subjective (multiple-choice, cloze, and judgment) and objective (essay writing, term explanation, and modelling-and-optimization) tasks in the smart-energy field.

2 Related Work

Foundation Language Models. LLMs trained on vast amounts of diverse and heterogeneous data, have accumulated extensive domain knowledge and contextual modeling capabilities. They have demonstrated human-level performance in many tasks. LLMs can be categorized into two types: 1) Closed-source models (such as OpenAI o1 (Jaech et al., 2024) and Claude): These models provide inference interfaces via APIs, making them suitable for industrial-grade deployment without the need for building custom computational resources. However, they cannot be customized or extended according to specific needs; 2) Open-source models (such as DeepSeek (Guo et al., 2025; DeepSeek-AI et al., 2024), Llama and Qwen (Bai et al., 2023, 2025)): These models offer complete training weights, allowing for customized applications based on downstream task requirements. This has led to the development of instruction-tuned models like Alpaca (Taori et al., 2023a), Vicuna (Chiang et al., 2023), and Dolly (Conover et al., 2023a). In this process, the quality of datasets becomes a critical factor affecting training outcomes.

Domain Language Models. LLMs excel in

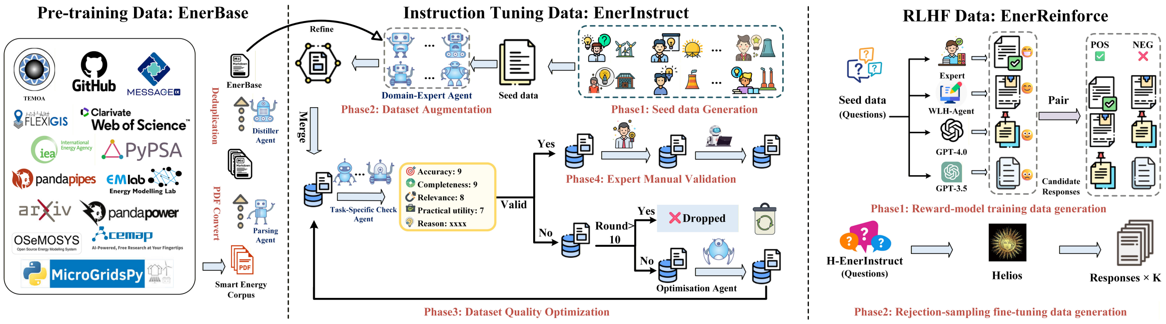

Figure 2: The multi-agent collaboration framework **EnerSys** provides the data required for **Helios**' three-stage training, including pre-training data (EnerBase), instruction tuning data (EnerInstruct), and RLHF data (EnerReinforce).

general reasoning (Jiang et al., 2025a; Nam et al., 2024), their performance in specialized applications is hampered by a lack of domain expertise. This limitation has led researchers to adapt foundation models for vertical domains such as medicine (Tian et al., 2024; Lin et al., 2025a), chemistry (Zhang et al., 2024; Zheng et al., 2025), ocean science (Bi et al., 2024), and geography (Deng et al., 2024). However, most domains are still in the early stages of exploration. In the energy sector, existing research primarily leverages general models' prior knowledge through prompt engineering or fine-tuning for load forecasting (Jin et al., 2023; Jiang et al., 2025b; Wu and Ling, 2024; Wang et al., 2025a), building energy modeling (Wang et al., 2025b; Jiang et al., 2024), teaching assistant (Lin et al., 2025b), and HVAC fault diagnosis (Zhang et al., 2025). These approaches focus on application (applying LLMs' prior knowledge to downstream tasks) rather than accumulation (enriching models with energy domain knowledge through pretraining). Furthermore, the energy domain faces a scarcity of high-quality training data due to literature repository access restrictions and computational resource costs (Wang et al., 2022b; Chen et al., 2024). The current instruction fine-tuning data construction method (Wang et al., 2023; Zhang et al., 2023), which relies heavily on large model generation, amplifies discrepancies between response styles and human preferences, posing a significant challenge. This paper introduces the first energy domain-specific large language model, a novel development that completes full-process training, constructs the domain's first training and evaluation datasets, and enhances model response alignment with human preferences through RLHF.

Multi-Agent Systems. Due to the extensive domain knowledge and robust semantic understanding capabilities of LLMs, they are employed as core components of agents to support intelligent decision-making and natural language interaction (Guo et al., 2024). In single-agent systems, a single agent carries out decision-making and task execution, which is suitable for structured scenarios with fewer variables (Chen et al., 2023). However, as the complexity of problems increases, single-agent systems face issues such as low decision-making efficiency, slow response times, and poor fault tolerance (Amirkhani and Barshooi, 2022). In contrast, multi-agent systems can effectively address these challenges through the collaboration of specialized agents and have been widely applied in complex scenarios such as interactive games (Mao et al., 2023; Xu et al., 2023), financial markets (Li et al., 2023), and social simulations (Park et al., 2023). Currently, some scholars (Bi et al., 2024; Ni and Buehler, 2024) are exploring ways to improve the efficiency and representativeness of domain dataset construction through multi-agent collaboration and distributed decision-making mechanisms. However, constructing domain datasets typically involves multiple steps, including data generation, deduplication, filtering, and optimization. Existing research often focuses only on optimizing specific steps and has not fully leveraged the potential of multi-agent systems in dataset construction.

3 Data Collection and Curation

To meet the stringent high-quality data requirements of Helios during the pre-training, instruction tuning, and RLHF stages, we have designed an efficient multi-agent collaborative dataset construction framework, EnerSys (see Fig. 2).

3.1 Pre-training Data: EnerBase

In this work, we conducted specialized text data pre-training based on the Qwen2.5-7B foundation model. The constructed Smart Energy Corpus includes open-access academic preprints, authoritative journal papers, specialized publications, domain-specific modeling toolkits, and application code, and IEA energy datasets from the smart energy domain. Data was collected from arXiv, Web of Science (WoS), Acemap, Github, and HuggingFace platforms. After data preprocessing, we constructed **EnerBase**, a high-quality pre-training corpus of *approximately 3 billion tokens* to enhance the model's accumulation of professional knowledge and technical application capabilities in the smart energy domain. In brief, the statistical characteristics of Smart Energy Corpus are shown in Table 1.

3.1.1 Smart Energy Corpus Collection

Scientific Literature. The smart energy domain's extensive scientific literature provides a high-quality training corpus for LLMs, enhancing their domain-specific knowledge understanding and application capabilities. To ensure the comprehensiveness of the corpus, we systematically decomposed the smart energy domain into 14 specialized sub-domains, and collected data for each separately. **1) Open-access Academic Preprints (OAP):** We crawled 173,541 PDF files from arXiv using subdomain-specific keywords, establishing the quantitative foundation of our Smart Energy Corpus. **2) Open-access Authoritative Journal Papers (OAJP):** We extracted metadata from WoS for leading energy journals and crawled 32,459 PDF files, establishing the *qualitative foundation* of our Smart Energy Corpus. **3) Specialized Publications (SP):** We crawled 363 book PDF files from the Acemap, enriching the Smart Energy Corpus knowledge framework.

Domain-specific Modeling Toolkits and Application Code (DMT&AC). Modern smart energy systems face exponential complexity due to multi-dimensional coupling of renewable integration, demand-side response, and power-carbon market mechanisms. Researchers employ high-precision algorithms and parallel computing for large-scale system optimization. Python dominates energy system modeling with its scientific computing ecosystem and machine learning capabilities, with 89% of modeling tools now open-source through community development (Majidi et al., 2025). To enhance language models' capabilities in parsing and generating specialized code for smart energy applications, we selected 19 representative frameworks (including Oemof (Hilpert et al., 2018), OSeMOSYS (Howells et al., 2011), TEMOA (Lerede et al., 2024)) and application libraries, extracting 5,389 Python files and 278 Jupyter notebooks.

International Energy Agency Datasets (IEAD). The IEA, covering 75% of global energy demand, has evolved from an oil crisis response mechanism to a platform governing energy security, economic growth, and environmental protection. Its statistics system provides authoritative data on supply-demand balance, emissions, renewables, and efficiency indicators across 170+ countries. To enhance LLMs' analytical capabilities for energy transition assessment, we incorporated the IEA_Energy_Dataset (Li, 2023) with 358,466 data points into our training corpus.

Table 1: Text Corpus Statistics for Helios Training.

Data Source	Smart Energy Corpus	EnerBase	
	Documents	Documents	Tokens (B)
OAP	173,541	153,165	2.314
OAJP	32,459	30,249	0.57
SP	363	342	0.038
DMT&AC	5,667	4,039	0.015
IEAD	358,466	345,874	0.019
Total	**570,496**	**533,669**	**2.956**

3.1.2 Smart Energy Corpus Processing

PDF Convert. Collected corpus primarily exists in PDF format, necessitating conversion to a unified format suitable for model training. Scientific literature contains abundant structured information, including tables, equations, and formulas; direct conversion to TXT format would result in critical information loss, causing LLMs to learn incomplete or incorrect content. Therefore, we selected Markdown as our unified conversion format to preserve these essential structural elements.

To balance computational throughput with structural integrity, we developed Python scripts based on Marker (Paruchuri, 2025). For processing efficiency, we deployed 10 servers equipped with NVIDIA RTX 4090 GPUs in a distributed architecture, each server configured with six parallel conversion workers. To enhance quality, we integrated OpenAI's GPT-4o as an intelligent agent (Parsing-Agent) to perform table reconstruction, mathematical formula standardization, form parsing, figure description generation, and reference normalization, ensuring structural completeness. Detailed hyperparameter configurations are provided in the supplementary table. Our system achieved an average processing speed of 2.21 seconds per page, completing the entire conversion process within 5 days. Fig. 3 demonstrates sample conversion results. The computationally efficient and structurally complete PDF-to-Markdown conversion framework, based on intelligent agents, presented in this paper, has been open-sourced on GitHub along with the dataset.

Content Filtering. Building upon this foundation, we additionally implement filtering mechanisms to remove private information, harmful content, and unintelligible or corrupted text.

Deduplication. Nevertheless, the Smart Energy Corpus inevitably contains a proportion of semantically similar fragments, causing the model during pre-training to update along nearly identical gradient directions and thus to "memorise" specific passages rather than acquire generalisable logical patterns (Tirumala et al., 2023). To address this problem, following the methodology outlined in (Abbas et al., 2023), we developed an efficient large-scale deduplication agent, Corpus Distiller, built on BERT-base. Corpus Distiller first performs K-Means clustering in the embedding space and subsequently removes samples located within the same epsilon-ball in each cluster.

Table 2: Datasets used to train Helios during the Universal Human Instruction Comprehension phase.

Dataset	Prompts
Alpaca-cleaned (Taori et al., 2023b)	51 800
Dolly-15K (Conover et al., 2023b)	15 011
Natural-Instructions (Muennighoff, 2022)	30 000
python_code_25k (FLOCK4H, 2023)	24 813
OpenR1-Math-220k (R1, 2025)	28 120
Toolbench (Qin et al., 2023)	10 328
Total	**160 072**

3.2 Instruction Tuning Data

Instruction Tuning is the key to bridging large-scale unsupervised pre-trained models with downstream applications. We have constructed a two-phase instruction fine-tuning framework of "Universal Human Instruction Comprehension (UHIC) to Domain-specific Task Adaptation (DS-TA)": First, high-quality general instruction samples are employed to conduct preliminary fine-tuning, enabling the model to learn to accomplish tasks according to natural-language instructions; subsequently, knowledge-intensive, specialized data are introduced for further fine-tuning, thereby enhancing the model's adaptability to domain-specific tasks. For these two phases, we curate a complementary general instruction dataset and knowledge-intensive dataset **EnerInstruct**, each uniformly organized in an <instruction,input,output> triplet format.

3.2.1 Universal Human Instruction Comprehension Data

In this stage, we have carefully selected six highly-recognized and high-quality open-source general-purpose supervised datasets: Alpaca-cleaned (Taori et al., 2023b), Dolly-15K (Conover et al., 2023b), Natural-Instructions (Muennighoff, 2022; Mishra et al., 2022; Wang et al., 2022a), python_code_25k (FLOCK4H, 2023), OpenR1-Math-220k (R1, 2025), and Toolbench (Qin et al., 2023). These datasets cover universal instruction understanding, mathematical reasoning, code enhancement, and tool utilization domains to improve Helios's foundational capabilities and domain application potential. The data volume of each dataset in the UHIC phase is shown in Table 2.

3.2.2 Domain-specific Task Adaptation data: EnerInstruct

Seed Data Collection. In this study, we engaged 10 senior experts in the smart energy domain to manually construct sample pairs for eleven downstream tasks: Fact Verification (FV), Reasoning (Res), Named Entity Recognition (NER), Summarization (Sum), Word Semantics (WS), Question and Answers (Q&A), Text Classification (TC), Explanation (Exp), Energy System Modeling (ESM), Single-Choice (S-C) and Multiple-Choice (M-C). Which across fourteen sub-fields: clean energy, cogeneration, combined cooling–heating–and–power, distributed energy, energy hub, energy management system, energy optimization, energy storage, energy transition, integrated energy, load forecasting, smart energy, smart grid, and virtual power plant. The resulting seed dataset, covers 14 sub-fields and 10 task categories, comprising 10 000 samples.

Dataset Augmentation. Smart energy encompasses multiple subfields, each exhibiting unique statistical characteristics and potential patterns. To ensure the professionalism and accuracy of the generated results, we design domain-specific expert agents for each subfield, enabling them to independently generate high-quality sample pairs for their respective areas and achieve parallelization and high-throughput data output. Specifically, we first refine the literature from each subfield within the Open-access Authoritative Journal Papers using a two-stage selection method based on "local

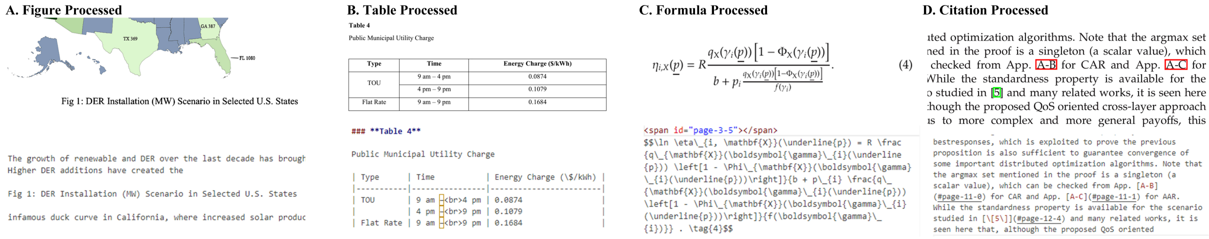

Figure 3: Text processed by the Parsing-Agent. A.Images: only the captions are retained, image bodies are removed; B.Tables: converted to Markdown format; C.Complex mathematical formulae: converted to Markdown format; D.Citations: for each citation, the corresponding page numbers of the referenced literature are specified.

Table 3: Statistics of EnerInstruct categorized by tasks.

Tasks	Records	Dataset Quality Optimization		Total (Cleaned)
		Filtered	optimized	
FV	20,839	17,370	706	4,175
Res	6,057	351	100	5,806
NER	423	327	283	379
Sum	449	392	323	380
WS	6,830	6,166	5,714	6,378
Q&A	11,973	7,900	3,878	7,951
TC	5,486	1,513	648	4,621
Exp	9,003	1,785	1,045	8,263
ESM	721	674	672	719
S-C	8,234	2,638	1,523	7,119
M-C	10,780	3,368	2,213	9,625
Entire Data	**80,795**	**42,484**	**17,105**	**55,416**

Algorithm 1 Two-Stage Literature Refinement for Energy Storage Domain

Require: P: Publication set; θ_{LC}: Citation threshold (70-th percentile); ε: DBSCAN distance (0.7); MinPts: DBSCAN density (5); m_k: Top papers per cluster
Ensure: V'': Refined core paper set

Stage 1: Local Citation Filtering
Construct paper network $V = \{v_1, \dots, v_n\}$ from P where each v_i represents a paper
Define citation indicator: $I(v_i \to v_j) = 1$ if paper v_i cites paper v_j, 0 otherwise
for $v_i \in V$ **do**
  $LC(v_i) \leftarrow \sum_{v_j \in V} I(v_j \to v_i)$ ▷ Local citation count
end for
$V' \leftarrow \{v_i \mid LC(v_i) \geq \theta_{LC}\}$ ▷ Filter high-cited papers
Stage 2: Co-citation Analysis
Build co-citation matrix $c_{ij} = \sum_{v_k \in V} I(v_k \to v_i) I(v_k \to v_j)$
$s_{ij} \leftarrow c_{ij} / \sqrt{c_{ii} c_{jj}}$ ▷ Normalized co-citation similarity
$\{C_1, \dots, C_K\} \leftarrow \text{DBSCAN}(S, \varepsilon, \text{MinPts})$ ▷ Cluster by similarity matrix S
for each cluster C_k **do**
  for $v_i \in C_k$ **do**
    $CD(v_i) \leftarrow \sum_{v_j \in C_k} s_{ij}$ ▷ Centrality degree within cluster
  end for
  $T_k \leftarrow$ top-m_k papers in C_k ranked by $CD(v_i)$
end for
$V'' \leftarrow \bigcup_{k=1}^{K} T_k$ ▷ Union of top papers from all clusters

Note: For the energy storage domain, $|P| = 5204$, $|V'| = 1561$, and $|V''| = 312$. We target approximately 300 papers per domain by adjusting m_k (5–15). Statistics for other domains are given in Appendix D.

citation count" and "co-citation analysis" to identify high academic value papers that constitute the foundational knowledge and theoretical framework of the discipline. These papers serve as a high-quality corpus (using the energy storage subfield as an example, refer to Algorithm 1). Subsequently, we fine-tune the corresponding expert agents using seed datasets from each subfield, enabling them to autonomously generate <instruction, input, output> triplets that conform to training standards based on the high-quality corpus. The original DS-TA phase data are constructed, with task assignment details provided in Table 3.

Dataset Quality Optimization. During dataset construction, domain-expert agents generated a large number of highly specialised samples for the various tasks. Nevertheless, the stochastic nature of sampling, the structural constraints imposed by the context-window length, and the potential hallucinations produced by LLMs can all cause fluctuations in sample quality and completeness. Extensive empirical work demonstrates that dataset size governs coverage and diversity, whereas sample quality determines the attainable upper bound on performance; the two must be carefully balanced. Suppose low-quality samples containing redundancy, noise, or inconsistent annotations are used for training. They will dilute the informative signal, amplify systemic bias, and ultimately erode the model's ability to follow instructions. To this end, we constructed a Check-Agent based on OpenAI o1, categorized by task, scoring each sample across the four dimensions of accuracy, completeness, relevance, and usefulness (out of 10), and providing reasons.

Samples that reach or exceed the threshold are retained, whereas those that do not are forwarded to an independent Optimization-Agent. Guided by the evaluation feedback, this agent performs automatic remediation—correcting errors, supplementing and enriching content, or conducting deeper analysis as necessary. The revised sample is then returned to the Check-Agent for re-evaluation. This "scoring–Optimization–re-scoring" loop may iterate up to ten times: if a sample passes within the allotted rounds, it is admitted to the training corpus; if it fails all ten rounds, it is deemed irreparable and permanently discarded. Check-Agent and the Optimization-Agent collaboratively optimise the data workflow, as shown in Fig 4. This procedure ultimately yields dataset **H-EnerInstruct**.

Expert Manual Validation. Finally, a panel of 12 domain experts rigorously examined each task sample in **H-EnerInstruct** (sampling 100–200 entries per task, proportional to that task's size). Tasks that did not meet the required standard were flagged, and the experts issued uniform revision guidelines that were then refined by OpenAI o1 to ensure the dataset's reliability. The optimized data were merged with the seed dataset to produce the final DS-TA phase dataset, **EnerInstruct** (Table 3). The statistics on expert optimization iterations are reported in Supplementary Section C.

3.3 RLHF Data: EnerReinforce

After large-scale pre-training and supervised instruction fine-tuning, Helios can already address a wide range of tasks in the energy domain. Nevertheless, these stages seldom make human values or preferences explicit, so the resulting models may acquire generation patterns that diverge from human expectations. To align Helios more effectively with human preferences, we adopt a targeted, two-stage approach consisting of reward model training followed by rejection sampling fine-tuning, and constructed **EnerReinforce**, which includes:

1) Reward Model Training Data: We sampled 5,000 subjective questions $Q_{RM} = \{q_i\}_{i=1}^{5000}$, and their expert answers $E_{Exp} = \{e_i\}_{i=1}^{5000}$ from seed dataset. For each q_i, additional answers E_{WLH}, $E_{GPT-4.0}$, $E_{GPT-3.5}$ were generated with (i) a Write-like-Human agent, (ii) GPT-4.0, and (iii) GPT-3.5. The four answers were then ranked by quality in the order E_{Exp} > E_{WLH} > $E_{GPT-4.0}$ > $E_{GPT-3.5}$, and pair adjacent response to obtain 3 sets of positive and negative sample pairs $\mathcal{P}_i = \{(E_{Exp}, E_{WLH}), (E_{WLH}, E_{GPT-4.0}), (E_{GPT-4.0}, E_{GPT-3.5})\}$, and formed the Reward-model training dataset $\mathcal{X}_{RM} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{5000}\}$.

2) Rejection Sampling Fine-tuning Data: From the subjective part of **EnerInstruct**, we selected 10,000 questions $Q_{RS} = \{q_i\}_{i=1}^{10000}$, that do not overlap with the seed set. Helios generated five candidate answers $A_i = \{a_i^1, a_i^2, \ldots, a_i^5\}$ for each question q_i, and these candidates serve as the basis for the rejection-sampling fine-tuning stage $\mathcal{X}_{RS} = \{(q_1, A_1), (q_2, A_2), \ldots, (q_{10000}, A_{10000})\}$.

3.4 Evaluation on Expertise in Smart Energy: EnerBench

To systematically assess the problem-solving capabilities of LLMs on scientific questions in smart energy research, we developed EnerBench, whose item-generation workflow adheres to a dual-track paradigm of Public-bank retrieval and Expert-directed authoring:

1) **Public-bank Retrieval:** Using each sub-discipline as a search keyword, representative questions were automatically harvested from multiple open-source evaluation platforms, ensuring extensive topical coverage and diversity.

2) **Expert-directed Authoring:** Five senior scholars in the smart energy domain were commissioned to craft additional, high-quality items for every task within each sub-discipline, thereby augmenting the benchmark's novelty and difficulty.

In its final form, EnerBench comprises 887 objective questions (S-C, M-C, and FV) and 625 subjective questions (Q&A, Exp, and ESM). The detailed distribution of questions across sub-disciplines is provided in Table 4.

Table 4: The statistics of EnerBench.

Question Type	Task	Prompts
Objective task	S-C	405
	M-C	254
	FV	228
Subjective tasks	Q&A	196
	Exp	249
	ESM	180

4 Helios training settings

4.1 Pre-training

During the pre-training stage, we employ the Qwen-2.5 7B model (Yang et al., 2024) (7.62B trainable parameters) as the initialization weights for Helios. A single-epoch training is subsequently conducted on a domain-specific corpus of approximately 3 billion tokens in the smart energy domain (22532 gradient update steps); the training hardware configuration consists of four NVIDIA A100-SXM 80 GB GPUs, with a total training time of 87 hours. The principal hyper-parameter settings are as follows: a peak learning rate of 3e-5, global batch size of 64, and a corresponding micro-batch size of 2.

4.2 Instruction tuning

In both stages of instruction learning (UHIC and DS-TA), we employ the Low-Rank Adaptation

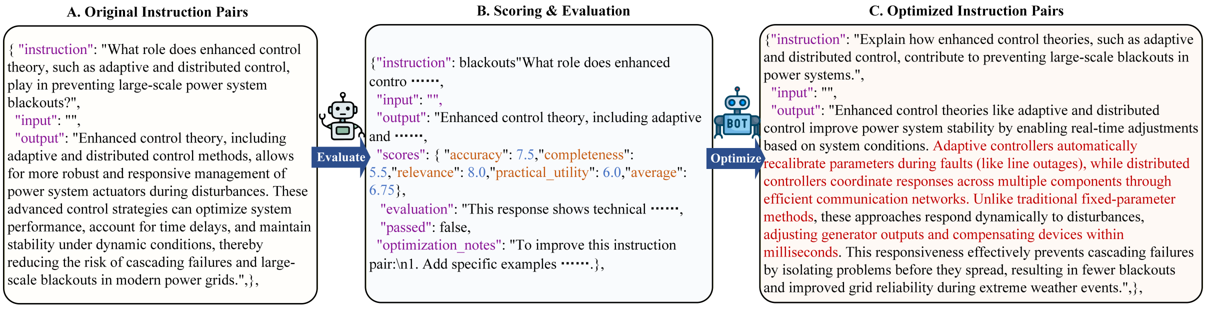

Figure 4: Dataset Quality Optimization workflow example. (A) Text before processing; (B) the Check-Agent scores the text quality and provides optimization suggestions; (C) the Optimization-Agent generates the optimized text based on those suggestions. We mark the differences in Red.

Table 5: Comparison of the performance of different LLMs across all tasks in EnerBench. The best results are indicated in **bold**, and the second-best results are <u>underlined</u>.

Model	**S-C**	**M-C**	**FV**	**ESM**			**Exp**			**Q&A**		
				A	**E**	**H**	**A**	**E**	**H**	**A**	**E**	**H**
Qwen3-8B-Instruct	50.24%	27.56%	47.81%	1.74	5.88	D	1.63	5.04	D	4.74	6.50	C
Llama3-8B-Instruct	68.42%	37.60%	58.77%	3.29	6.03	D	3.53	6.29	D	5.13	6.47	C
Qwen3-14B-Instruct	64.59%	35.24%	54.61%	2.26	6.54	D	4.36	6.90	C	6.22	7.06	C
Qwen3-32B-Instruct	80.14%	44.09%	62.72%	3.82	6.93	D	5.51	7.32	C	6.83	7.51	**B**
GPT-3.5-Turbo	91.63%	<u>53.93%</u>	84.65%	<u>6.03</u>	<u>8.05</u>	<u>C</u>	6.94	8.57	**B**	7.24	<u>8.37</u>	**B**
GPT-4	**95.69%**	**61.18%**	**93.86%**	**7.61**	**8.97**	**B**	**8.63**	**9.58**	**B**	**7.64**	**9.21**	**B**
Helios	<u>93.78%</u>	53.58%	<u>89.91%</u>	5.73	7.83	<u>C</u>	<u>7.03</u>	<u>9.19</u>	**B**	<u>7.39</u>	8.26	**B**

(LoRA) technique: while keeping the pre-trained weights $W_0 \in \mathbb{R}^{n\times d}$ completely frozen, we inject two trainable low-rank matrices $A \in \mathbb{R}^{n\times r}$ and $B \in \mathbb{R}^{r\times d}$ in parallel ($r \ll \min(n, d)$). Here, n and d represent the input and output dimensions of the weight matrix W_0, and r denotes the rank of the low-rank matrices. This approach preserves the general representations learned from large-scale corpora during pre-training, while significantly reducing the number of trainable parameters and lowering computational costs. The corresponding forward propagation is given by:

$$h = W_0 x + BAx, \tag{1}$$

where h denotes the adapted output. The training hardware configuration consists of four NVIDIA RTX 4090 GPUs, with a total training time of 17 hours. During the instruction tuning stage, a two-stage fine-tuning of the model was performed. The model was first fine-tuned with generic instructions and then fine-tuned with knowledge enhancement. In the UHIC stage, the key hyperparameter settings are as follows: a peak learning rate of 2e-5, global batch size of 64, and a corresponding micro-batch size of 2. In the DS-TA stage, the key hyperparameter settings are as follows: a peak learning rate of 1e-5, a global batch size of 64, and a corresponding micro-batch size of 2.

4.3 RLHF

Reward Model Training. We employ a pairwise ranking loss to train reward model, enabling it to distinguish between responses of varying quality:

$$\mathcal{L}_{\text{RM}} = -\frac{1}{|\mathcal{D}_{\text{RM}}|}\sum_{i=1}^{\mathcal{D}_q}\sum_{j=1}^{\mathcal{D}_{\text{pair}}} \log\sigma\big(r_\phi(q_i, a_i^{j+}) - r_\phi(q_i, a_i^{j-})\big), \tag{2}$$

where $\mathcal{D}_{RM}$ denotes the set of training examples ($\mathcal{D}_{RM} = \mathcal{D}_q * \mathcal{D}_{\text{pair}}$), $\mathcal{D}_q$ denotes the cardinality of Q_{RM}, $\mathcal{D}_{\text{pair}}$ denotes the number of positive–negative sample pairs associated with q_i. a_i^{j+} and a_i^{j-} represent the j-th positive and negative samples of q_i, respectively. $r_\phi(q_i, a_i^j)$ is the quality score assigned by the reward model to response a_i^j; and $\sigma(\cdot)$ is the sigmoid function, which maps the difference in scores to the probability that the positive sample is preferred over the negative one. By minimizing $\mathcal{L}_{\text{RM}}$, the model is driven to enlarge the gap between $r_\phi(x, y^+)$ and $r_\phi(x, y^-)$, thereby learning to distinguish responses of differing quality. For the hyperparameters, we train for three epochs with a batch size of 8, and the warm-up stage accounts for 5% of the total steps.

Rejection Sampling Fine-tuning. During the Rejection Sampling fine-tuning phase, the reward model is used to evaluate and rank $\mathcal{X}_{RS}$:

$$s_i = \{r_\phi(q_i, a_i^j)\}_{j=1}^{\mathcal{D}_c}, A_i^* = sort(A_i, desc\ by\ s_i), \tag{3}$$

$\mathcal{D}_c$ denotes the number of candidate responses in A_i, s_i denotes the score assigned to each response by the reward model. A_i^* is obtained by sorting A_i in descending order of s_i. Then, select the Top-k samples as the "gold standard" for further fine-tuning Helios:

$$\mathcal{X}_{\text{RS}}^{Gold} = \{(q_i, a_i^*) | q_i \in Q_{RS}, a_i^* \in \text{TopK}(A_i^*, k)\}_{i=1}^{\mathcal{D}_r}, \tag{4}$$

where $\mathcal{D}_r$ denotes the number of questions in Q_{RS}, a_i^* is the set of the top k values sampled from A_i^*. We trained the model for 5 epochs with a learning rate of 3e-5 and a batch size of 64.

5 Evaluation and Results

We evaluated the performance of Helios, Qwen3-8B-Instruct, Llama3-8B-Instruct, Qwen3-14B-Instruct, Qwen3-32B-Instruct, GPT3.5-Turbo and GPT-4 on EnerBench and compared their results. The results are presented in Table 5.

Objective Tasks in EnerBench. For objective tasks, performance is evaluated using accuracy. Specifically, for multiple-choice items, the scoring rubric is: full credit is awarded only when all correct options are selected; partial credit is granted when some correct options are omitted; and no credit is given if any incorrect option is chosen. Helios attains an average accuracy of 79.09% in answering object questions, markedly outperforming models of comparable size such as Qwen3-8B-Instruct (41.87%) and Llama3-8B-Instruct (54.93%), and reaching a level comparable to GPT-4 with approximately 220 billion parameters. This indicates that the model successfully acquired intelligent-energy domain knowledge during further pre-training.

Subjective Tasks in EnerBench. For subjective tasks, we implemented a tri-dimensional evaluation framework: A-Score (GPT-o1 benchmark-based comparative assessment on a 10-point scale), E-Score (GPT-o1 independent quality assessment on a 10-point scale), and H-Grade (expert evaluation using an A/B/C/D grading system). The assessment results demonstrate that Helios outperforms parameter-equivalent models like Qwen3-8B-Instruct and Llama3-8B-Instruct across domain-specific QA, Exp, and ESM tasks. Specifically, Helios approaches GPT-4 capability levels in QA and Exp tasks. Regarding ESM capabilities, Helios can leverage energy domain-specific libraries for complex problem modeling. However, it still exhibits a performance gap compared to GPT-4 due to parameter size constraints, yet achieves performance comparable to GPT-3.5-Turbo. We provide a detailed discussion of model hallucinations in section H of the supplementary materials.

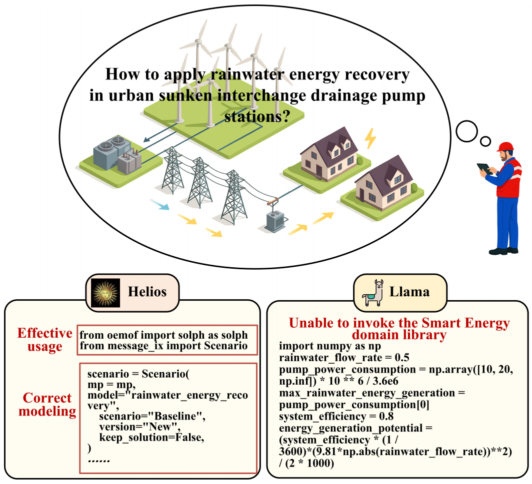

Figure 5: Case analysis of modeling tasks in the smart energy domain.

Exploring the Potential of Helios. We attempt to address a practical modelling and Optimization task in the smart energy domain using Helios. In this example, our requirement is: *How to apply rainwater energy recovery in urban sunken interchange drainage pump stations?* and to provide the implementation code (Fig. 5). It can be observed that Helios is able to effectively invoke domain-specific packages for intelligent energy (oemof and message_ix) to accomplish the modelling task, whereas Llama3-8B can only call numpy to perform purely numerical computations, which deviates substantially from the task requirements and lacks practical relevance to the energy domain.

6 Conclusion

In this study, we introduce Helios, the first LLM explicitly developed for the smart energy domain, capable of addressing diverse tasks. We introduce EnerSys, a comprehensive multi-agent pipeline that furnishes Helios with high-quality data, producing EnerBase, EnerInstruct, and EnerReinforce. We also release Benchmark, the domain's first evaluation suite, enabling systematic appraisal of LLMs on smart energy tasks. Experiments show that, Helios offers significant gains in domain knowledge and task performance.

Limitations

Although Helios has demonstrated excellent capabilities in knowledge integration and automatic code generation within the smart energy domain, its role is consistently positioned as an "intelligent reference assistant" rather than an autonomous decision-making engine. In high-risk tasks such as power system modeling, dispatch, and safety assessment, Helios only outputs code drafts and inferential suggestions for review by engineers. Direct deployment without professional review could lead to significant economic losses or even physical risks due to potential model assumption biases, numerical instability, or omission of boundary conditions. Consequently, the model's outputs do not constitute an engineering guarantee. The final decision-making responsibility must be borne by the user and their affiliated institution; when results are uncertain or contradict engineering experience, it is essential to revert to traditional manual calculation and simulation for verification.

Concerning hallucinations in Helios, they mainly manifest as linguistic repetition, instruction misunderstanding, conceptual confusion, and structural errors. For instance, in factual judgment tasks, the model occasionally misinterprets the task as question-answering, a problem that is significantly mitigated by explicitly appending "Please output True/False directly" to the prompt. Conceptual confusion stems from the interdisciplinary, fragmented, and rapidly evolving nature of smart energy knowledge; experiments show its occurrence rate remains within an acceptable range. Linguistic repetition and structural hallucinations are largely associated with the base model, Qwen-2.5-7B, and are difficult to eliminate completely through domain-specific fine-tuning alone; thus, they are not a primary focus of this paper.In summary, Helios has the aforementioned limitations regarding ethics, risks, and deployment, and should be applied cautiously within a strict framework of human-computer collaboration and safety governance.

Ethical considerations

Regarding ethics and bias, Helios is primarily trained on high-quality corpora such as academic papers and monographs, and its instruction data has undergone rigorous cleaning, resulting in minimal potential for ethical or bias issues.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No.52576234) and the National Key R & D Program of China (Grant No.2023YFE0108600). This work is also supported by the "Pioneer" and "Leading Goose" R & D Program of Zhejiang (Grant No. 2025C02237)

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*.

Abdollah Amirkhani and Amir Hossein Barshooi. 2022. Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55(5):3897–3935.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. 2024. OceanGPT: A large language model for ocean science tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3357–3372, Bangkok, Thailand. Association for Computational Linguistics.

F Ceglia, P Esposito, E Marrasso, and M Sasso. 2020. From smart energy community to smart energy municipalities: Literature review, agendas and pathways. *Journal of Cleaner Production*, 254:120118.

Fuhao Chen, Jie Yan, Yongqian Liu, Yamin Yan, and Lina Bertling Tjernberg. 2024. A novel meta-learning approach for few-shot short-term wind power forecasting. *Applied Energy*, 362:122838.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Ali Ghodsi, Patrick Wendell, and Matei Zaharia. 2023a. Hello dolly: Democratizing the magic of chatgpt with open models.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023b. Free dolly: Introducing the world's first truly open instruction-tuned llm.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Cheng Deng, Tianhang Zhang, Zhongmou He, Qiyuan Chen, Yuanyuan Shi, Yi Xu, Luoyi Fu, Weinan Zhang, Xinbing Wang, Chenghu Zhou, Zhouhan Lin, and Junxian He. 2024. K2: A foundation language model for geoscience knowledge understanding and utilization. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, page 161–170.

FLOCK4H. 2023. python-codes-25k: A python code instruction dataset.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach, and G. Plessmann. 2018. The open energy modelling framework (oemof) - a new approach to facilitate open science in energy system modelling. *Energy Strategy Reviews*, 22:16–25.

Mark Howells, Holger Rogner, Neil Strachan, Charles Heaps, Hillard Huntington, Socrates Kypreos, Semida Silveira, Joe DeCarolis, Morgan Bazillian, and Alexander Roehrl. 2011. Osemosys: The open source energy modeling system: An introduction to its ethos, structure and development. *Energy Policy*, 39:5850–5870.

Yi Hu, Hyeonjin Kim, Kai Ye, and Ning Lu. 2025. Applying fine-tuned llms for reducing data needs in load profile analysis. *Applied Energy*, 377:124666.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 242 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Gang Jiang, Zhihao Ma, Liang Zhang, and Jianli Chen. 2024. Eplus-llm: A large language model-based computing platform for automated building energy modeling. *Applied Energy*, 367:123431.

Haoyu Jiang, Zhi-Qi Cheng, Gabriel Moreira, Jiawen Zhu, Jingdong Sun, Bukun Ren, Jun-Yan He, Qi Dai, and Xian-Sheng Hua. 2025a. Ucdr-adapter: Exploring adaptation of pre-trained vision-language models for universal cross-domain retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5429–5438.

Haoyu Jiang, Boan Qu, Junjie Zhu, Fanjie Zeng, Xiaojie Lin, and Wei Zhong. 2025b. Hyperload: A cross-modality enhanced large language model-based framework for green data center cooling load prediction. *Preprint*, arXiv:2512.19114.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2023. Time-llm: Time series forecasting by reprogramming large language models.

Daniele Lerede, Valeria Di Cosmo, and Laura Savoldi. 2024. Temoa-europe: an open-source and open-data energy system optimization model for the analysis of the european energy mix. *Energy*, 308:132850.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 36, pages 51991–52008.

Zihao Li. 2023. Iea energy dataset.

Wenlong Liao, Shouxiang Wang, Dechang Yang, Zhe Yang, Jiannong Fang, Christian Rehtanz, and Fernando Porté-Agel. 2025. Timegpt in load forecasting: A large time series model perspective. *Applied Energy*, 379:124973.

Tianwei Lin, Wenqiao Zhang, Sijing Li, Yuqian Yuan, Binhe Yu, Haoyuan Li, Wanggui He, Hao Jiang, Mengze Li, Xiaohui Song, Siliang Tang, Jun Xiao, Hui Lin, Yueting Zhuang, and Bengchin Ooi. 2025a.

Healthgpt: A medical large vision-language model for unifying comprehension and generation via heterogeneous knowledge adaptation. In *Proceedings of the International Conference on Machine Learning*.

Xiaojie Lin, Zheng Luo, Liuliu Du-Ikonen, Xueru Lin, Yihui Mao, Haoyu Jiang, Shuai Wang, Chongshuo Yuan, Wei Zhong, and Zitao Yu. 2025b. Generative artificial intelligence: Pioneering a new paradigm for research and education in smart energy systems. *Energy and AI*, 22:100610.

Henrik Lund, Poul Alberg Østergaard, David Connolly, and Brian Vad Mathiesen. 2017. Smart energy and smart energy systems. *Energy*, 137:556–565.

Hassan Majidi, Mohammad Mohsen Hayati, Christian Breyer, Behnam Mohammadi-ivatloo, Samuli Honkapuro, Hannu Karjunen, Petteri Laaksonen, and Ville Sihvonen. 2025. Overview of energy modeling requirements and tools for future smart energy systems. *Renewable and Sustainable Energy Reviews*, 212:115367.

Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Tao Ge, and Furu Wei. 2023. Alympics: Llm agents meet game theory. In *Proceedings of the International Conference on Computational Linguistics*, pages 2845–2866.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the International Conference on Computational Linguistics*.

Niklas Muennighoff. 2022. natural-instructions: Preprocessed version of super-natural-instructions.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM International Conference on Software Engineering*, pages 1–13.

Bo Ni and Markus J Buehler. 2024. Mechagents: Large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge. *Extreme Mechanics Letters*, 67:102131.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 1–22.

Vikram Paruchuri. 2025. Marker: Convert pdf to markdown + json quickly with high accuracy.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *Proceedings of the International Conference on Learning Representations*.

Open R1. 2025. Openr1-math-220k: A large-scale dataset for mathematical reasoning.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023a. Alpaca: A strong, replicable instruction-following model.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Stanford alpaca: An instruction-following llama model.

Jakob Zinck Thellufsen, Henrik Lund, P Sorknæs, PA Østergaard, M Chang, D Drysdale, Steen Nielsen, SR Djørup, and K Sperling. 2020. Smart energy cities in a 100% renewable energy context. *Renewable and Sustainable Energy Reviews*, 129:109922.

Yuanhe Tian, Ruyi Gan, Yan Song, Jiaxing Zhang, and Yongdong Zhang. 2024. ChiMed-GPT: A Chinese medical large language model with full training regime and better alignment to human preferences. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7156–7173.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S Morcos. 2023. D4: improving llm pretraining via document de-duplication and diversification. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 53983–53995.

Chengsen Wang, Qi Qi, Jingyu Wang, Haifeng Sun, Zirui Zhuang, Jinming Wu, Lei Zhang, and Jianxin Liao. 2025a. Chattime: A unified multimodal time series foundation model bridging numerical and textual data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12694–12702.

Meng Wang, Jingfeng Zhou, Yujing Liang, Hang Yu, and Rui Jing. 2025b. Climate change impacts on city-scale building energy performance based on gis-informed urban building energy modelling. *Sustainable Cities and Society*, 125:106331.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby

Kuznia, Krima Doshi, Maitreya Patel, and 21 others. 2022a. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ze Wang, Yipin Zhou, Rui Wang, Tsung-Yu Lin, Ashish Shah, and Ser Nam Lim. 2022b. Few-shot fast-adaptive anomaly detection. *Proceedings of the Advances in Neural Information Processing Systems*, 35:4957–4970.

Tangjie Wu and Qiang Ling. 2024. Stellm: Spatio-temporal enhanced pre-trained large language model for wind speed forecasting. *Applied Energy*, 375:124034.

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. 2023. Language agents with reinforcement learning for strategic play in the werewolf game. In *Proceedings of the International Conference on Machine Learning*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li. 2024. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*.

Jian Zhang, Chaobo Zhang, Jie Lu, and Yang Zhao. 2025. Domain-specific large language models for fault diagnosis of heating, ventilation, and air conditioning systems by labeled-data-supervised fine-tuning. *Applied Energy*, 377:124378.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, and Fei Wu. 2023. Instruction tuning for large language models: A survey. *ACM Computing Surveys*.

Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh TN Nguyen, Lauren T May, Geoffrey I Webb, and Shirui Pan. 2025. Large language models for scientific discovery in molecular property prediction. *Nature Machine Intelligence*, pages 1–11.