HELIOS: A FOUNDATIONAL LANGUAGE MODEL FOR SMART ENERGY KNOWLEDGE REASONING AND APPLICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In the global drive toward carbon neutrality, deeply coordinated smart energy systems underpin industrial transformation, yet their interdisciplinary, fragmented, and fast-evolving expertise prevents general-purpose large language models (LLMs), lacking domain knowledge and physical-constraint awareness, from delivering precise engineering-aligned inference and generation. 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 LLMs 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. All training data and model checkpoints are publicly available at `anonymous.4open.science/r/Helios-F4DF/`.

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); Dincer & Acar (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.

However, general LLMs often deliver reasoning that is semantically plausible yet physically invalid Friel & 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 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. 1): 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 976 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:

- 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.
- 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.
- 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.

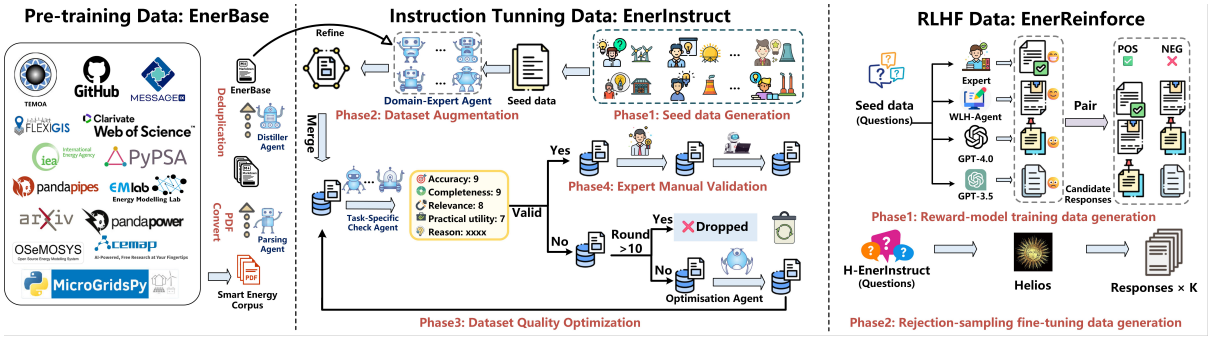

Figure 1: The multi-agent collaboration framework **EnerSys** provides the data required for **Helios**' three-stage training, including pre-training data (EnerBase), instruction tuning data (EnerInstruct), and RLHF data (EnerReinforce).

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); Liu 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. (2023b), 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 general reasoning Jiang et al. (2025); 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. (2025), 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); Hu et al. (2025); Wu & Ling (2024); Wang et al. (2025a), building energy modeling Wang et al. (2025b); Jiang et al. (2024), 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.

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. 1).

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 *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 large language models, 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, including load forecasting, and energy storage, and collected data for each separately.

- **Open-access Academic Preprints**: We crawled 173,541 PDF files from arXiv using subdomain-specific keywords, establishing the quantitative foundation of our Smart Energy Corpus.
- **Open-access Authoritative Journal Papers**: We extracted metadata from WoS for leading energy journals and crawled 32,459 PDF files, establishing our Smart Energy Corpus foundation.
- **Specialized Publications**: We crawled 363 professional book PDF files from the ACEmap platform, enriching our Smart Energy Corpus knowledge framework.

Domain-specific Modeling Toolkits and Application Code. 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.

IEA Energy Datasets. The International Energy Agency (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 358,446 data points into our training corpus.

Table 1: Text Corpus Statistics for Helios Training.

Data Source	Smart Energy Corpus	EnerBase	
	Documents	Documents	Tokens(B)
Open-access Academic Preprints	173,541	153,165	2.314
Open-access Authoritative Journal Papers	32,459	30,249	0.57
Specialized Publications	363	342	0.038
Domain-specific Modeling Toolkits and Application Code	5,667	4,039	0.015
IEA Energy Datasets	358,466	345,874	0.019
Total	**570,458**	**533,669**	**2.956**

3.1.2 SMART ENERGY CORPUS PROCESSING

PDF Convert. Our Smart Energy 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 large language models 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. 2 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.

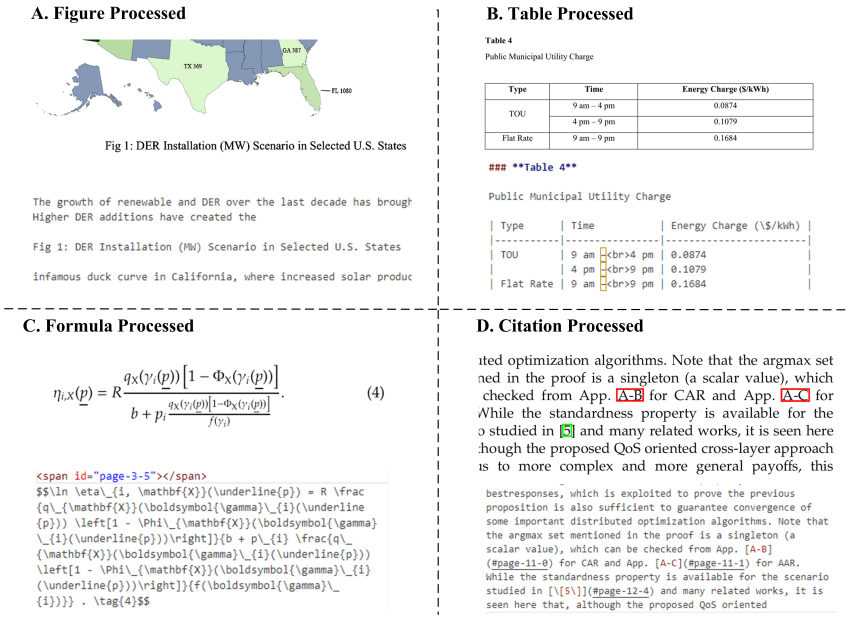

Figure 2: Text processed by the Parsing-Agent. A.Images: only the captions are retained, image bodies are removed; B.Tables: converted to Markdown format; C.Complex mathematical formulae: converted to Markdown format; D.Citations: for each citation, the corresponding page numbers of the referenced literature are specified.

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.

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. (2023a), 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 lewtun & Face (2025), and Toolbench Qin et al. (2023). These datasets cover universal instruction understanding, mathematical reasoning, code enhancement, and tool utilization domains to improve Heilos's

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 $V = \{v_1, ..., v_n\}$ 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
$\{C_1, \ldots, C_K\} \leftarrow \text{DBSCAN}(S, \varepsilon, \text{MinPts})$ ▷ 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 energy storage domain shown here, $|P| = 5204$, $|V'| = 1561$, and $|V''| = 312$. We target approximately 300 papers for each domain by manually adjusting m_k values (5-15). The statistics for other domains are available in supplementary material Section D.

foundational capabilities and domain application potential. For detailed information, please refer to supplementary material Section C.

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 all fourteen sub-fields and ten task categories, comprising 10000 data 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 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 2.

Table 2: 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	6,269
Q&A	11,973	7,900	3,878	7,169
TC	5,486	1,513	648	4,621
Exp	9,003	1,785	1,045	7,765
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**	**54,027**

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 large language models 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 Optimisation-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–optimisation–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 Optimisation-Agent collaboratively optimise the data workflow, as shown in Fig 3. This procedure ultimately yields dataset **H-EnerInstruct**.

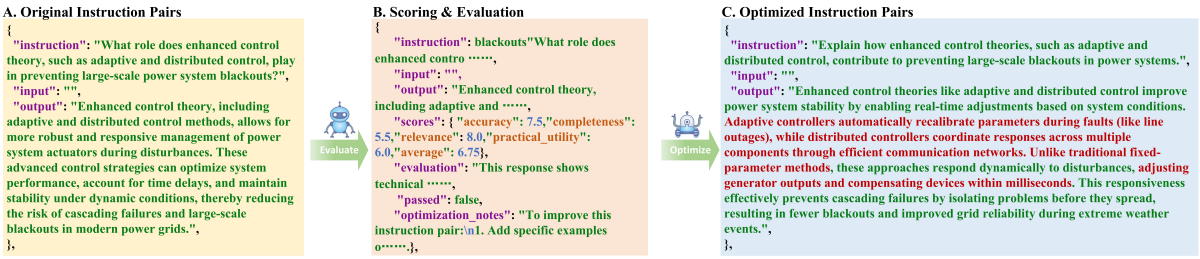

Figure 3: Dataset Quality Optimization workflow example. (a) Text before processing; (b) the Check-Agent scores the text quality and provides optimization suggestions; (C) the Optimisation-Agent generates the optimized text based on those suggestions. We mark the differences in Red.

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 2). The statistics on expert optimization iterations are reported in Supplementary Section G.

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. To this end, we have 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 976 objective questions (Single-Choice, Multiple-Choice, and Fact Verification) and 625 subjective questions (Question and Answers, Word explanation, and Energy System Modeling). The detailed distribution of questions across sub-disciplines is provided in Table 4.

Table 3: The statistics of EnerBench.

Question Type	Task	Prompts
Objective task	Single-Choice	405
	Multiple-Choice	254
	Fact Verification	228
Subjective tasks	Question and Answers	196
	Word explanation	249
	Energy System Modeling	180

4 HELIOS TRAINING SETTINGS

To mitigate overfitting and enhance generalization across all training phases, we employed a consistent early stopping criterion based on performance on a held-out validation set. After each fixed training interval, we evaluated a phase-specific metric (e.g., perplexity or reward accuracy). The model's state was checkpointed only if the metric improved by more than a predefined threshold. If no such improvement occurred for a specified number of consecutive evaluations (i.e., the 'patience' parameter), training was halted, and the weights from the last best-performing checkpoint were restored for subsequent use. This unified methodology ensured both computational efficiency and the preservation of the model's optimal generalization state. The detailed hyperparameter settings are provided in Supplementary Section K.

4.1 PRE-TRAINING

During the pre-training stage, we employ the Qwen-2.5 7B model Yang et al. (2024) (7.62 B 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, a 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 (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 generic instruction fine-tuning stage, the key hyperparameter settings are as follows: a peak learning rate of 2e-5, a global batch size of 64, and a corresponding micro-batch size of 2. In the knowledge enhancement instruction fine-tuning 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 the reward model, enabling it to distinguish between responses of varying quality:

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

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. The hardware configuration and efficient fine-tuning techniques were kept identical to those used during the instruction-tuning phase.

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 4.

5.1 Insights from Performance Results.

Object 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, Explanation, and Modeling tasks. Specifically, Helios approaches GPT-4 capability levels in QA and Explanation tasks. Regarding Modeling 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 Appendix H of the supplementary materials.

5.2 Exploring the Potential of Helios

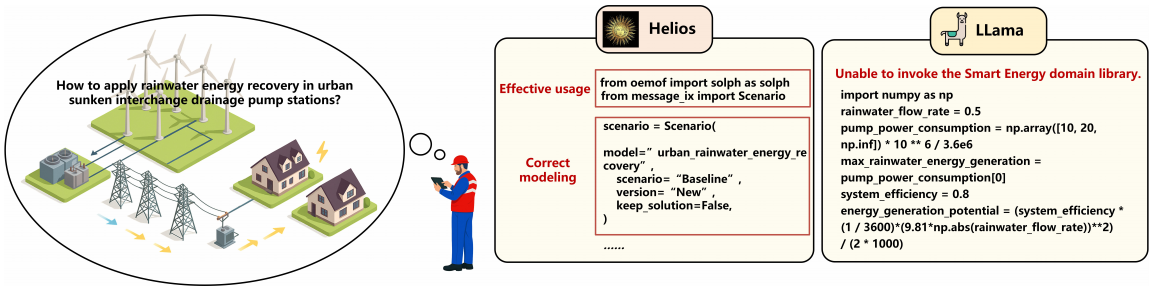

Figure 4: Case Analysis of Modeling Tasks in the Smart Energy Domain.

5.2.1 Energy System Modeling.

In Fig. 4, we attempt to address a practical modelling and optimisation 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. 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 sector.

Table 4: 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 underlined.

Model	S-C	M-C	FC	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%	53.93%	84.65%	6.03	8.05	C	6.94	8.57	**B**	7.24	8.37	**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	93.78%	53.58%	89.91%	5.73	7.83	C	7.03	9.19	**B**	7.39	8.26	**B**

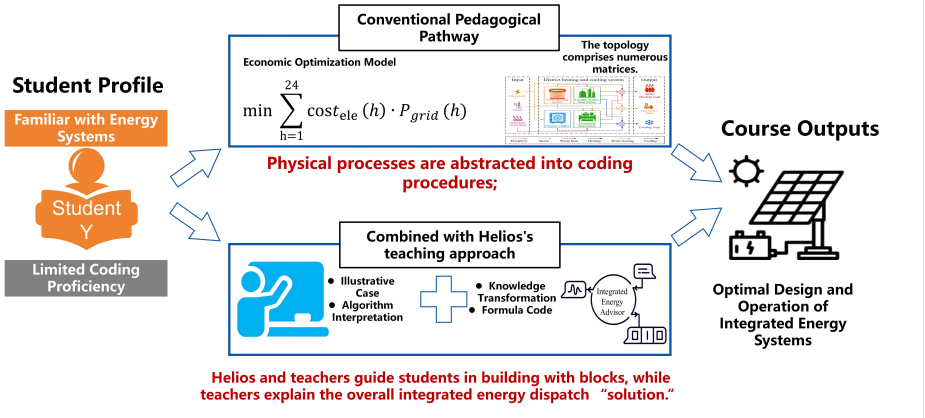

Figure 5: Examples of Helios' Involvement in Supporting Education.

5.2.2 Education.

We further investigates the application value and capability boundaries of Helios in educational settings within the smart energy domain. Four major challenges are identified in current smart energy professional education (as illustrated in S-Fig. 8): fragmented disciplinary knowledge and complementary competencies, prominent shortcomings in mainstream student competencies, acute scarcity of interdisciplinary talent, challenges in converting diverse matrix types and physical processes into executable forms.

Based on Helios, a complete educational workflow can be constructed that encompasses automated demonstration cases, step-by-step algorithm explanations, and formula/code examples to address the above challenges. This framework structures and visualizes complex multi-energy system principles, lowering barriers to abstract models while supporting instructors in building tiered knowledge frameworks. Students progress from localized physical processes to data import, model construction, and optimization, ultimately understanding the logic of integrated scheduling solutions (Fig. 5). Helios further integrates real-time personalized prompts and QA adapted to individual progress, enhancing teaching efficiency and learning experience. Through visual demonstrations, it cultivates students' ability to apply digital tools in real-world energy engineering.

6 Conclusion

In this study, we introduce Helios, the first LLMs 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 (i) the inaugural smart-energy pre-training corpus EnerBase, (ii) the first instruction-tuning corpus EnerInstruct, and (iii) the first reinforcement-learning-from-human-feedback corpus EnerReinforce. We also release Benchmark, the domain's first evaluation suite, enabling systematic appraisal of language models on smart-energy tasks. Experiments show that, relative to comparable general-purpose models, Helios offers significant gains in domain knowledge and task performance, especially for energy modelling and optimisation.

REFERENCES

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

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. OceanGPT: A large language model for ocean science tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3357–3372, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.184. URL https://aclanthology.org/2024.acl-long.184/.

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

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

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Ali Ghodsi, Patrick Wendell, and Matei Zaharia. Hello dolly: Democratizing the magic of chatgpt with open models. *Databricks blog. March*, 24, 2023a.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023b. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

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. K2: A foundation language model for geoscience knowledge understanding and utilization. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, pp. 161–170, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635772. URL https://doi.org/10.1145/3616855.3635772.

Ibrahim Dincer and Canan Acar. Smart energy systems for a sustainable future. *Applied energy*, 194:225–235, 2017.

FLOCK4H. python-codes-25k: A python code instruction dataset, 2023. URL https://huggingface.co/datasets/flytech/python-codes-25k.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach, and G. Plessmann. The open energy modelling framework (oemof) - a new approach to facilitate open science in energy system modelling. *Energy Strategy Reviews*, 22:16–25, November 2018. ISSN 2211-467X. doi: 10.1016/j.esr.2018.07.001. URL http://dx.doi.org/10.1016/j.esr.2018.07.001.

Mark Howells, Holger Rogner, Neil Strachan, Charles Heaps, Hillard Huntington, Socrates Kypreos, Semida Silveira, Joe DeCarolis, Morgan Bazillian, and Alexander Roehrl. Osemosys: The open source energy modeling system: An introduction to its ethos, structure and development. *Energy Policy*, 39:5850–5870, 10 2011. doi: 10.1016/j.enpol.2011.06.033.

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

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

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

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

Daniele Lerede, Valeria Di Cosmo, and Laura Savoldi. Temoa-europe: an open-source and open-data energy system optimization model for the analysis of the european energy mix. *Energy*, 2024. URL https://api.semanticscholar.org/CorpusID:272037036.

lewtun and Hugging Face. Openr1-math-220k: A large-scale dataset for mathematical reasoning, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large scale language model society. 2023.

Zihao Li. Iea energy dataset, 2023. URL https://huggingface.co/datasets/Zihao-Li/IEA_Energy_Dataset. Energy-related dataset covering topics of Oil, Coal, Wind, Hydrogen, Bioenergy, Electric vehicles, Heating, Building envelopes, Methane abatement and Chemicals. Data sources are reports from the International Energy Agency (IEA) website.

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

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. Healthgpt: A medical large vision-language model for unifying comprehension and generation via heterogeneous knowledge adaptation. *ArXiv*, abs/2502.09838, 2025. URL https://api.semanticscholar.org/CorpusID:276394558.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

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

Hassan Majidi, Mohammad Mohsen Hayati, Christian Breyer, Behnam Mohammadi-ivatloo, Samuli Honkapuro, Hannu Karjunen, Petteri Laaksonen, and Ville Sihvonen. Overview of energy modeling requirements and tools for future smart energy systems. *Renewable and Sustainable Energy Reviews*, 212 (C), 2025. doi: 10.1016/j.rser.2025.115367. URL https://ideas.repec.org/a/eee/rensus/v212y2025ics1364032125000401.html.

Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Tao Ge, and Furu Wei. Alympics: Language agents meet game theory. *arXiv preprint arXiv:2311.03220*, 5, 2023.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*, 2022.

Niklas Muennighoff. natural-instructions: Preprocessed version of super-natural-instructions, 2022. URL https://huggingface.co/datasets/Muennighoff/natural-instructions.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–13, 2024.

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

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–18, 2022.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Vikram Paruchuri. Marker: Convert pdf to markdown + json quickly with high accuracy. https://github.com/VikParuchuri/marker, 2025.

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. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023a.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023b.

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

Yuanhe Tian, Ruyi Gan, Yan Song, Jiaxing Zhang, and Yongdong Zhang. ChiMed-GPT: A Chinese medical large language model with full training regime and better alignment to human preferences. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7156–7173, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.386. URL https://aclanthology.org/2024.acl-long.386/.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S Morcos. D4: improving llm pretraining via document de-duplication and diversification. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 53983–53995, 2023.

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

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*, 2022a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, 2023. Association for Computational Linguistics.

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

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

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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. Chemllm: A chemical large language model, 2024. URL https://arxiv.org/abs/2402.06852.

Jian Zhang, Chaobo Zhang, Jie Lu, and Yang Zhao. 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, 2025.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023. Accessed: 2025-05-14.

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

A Extensive Related Work

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 & Barshooi (2022)(Fig. 6). 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; 2022). Currently, some scholars Bi et al. (2024); Ni & 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.

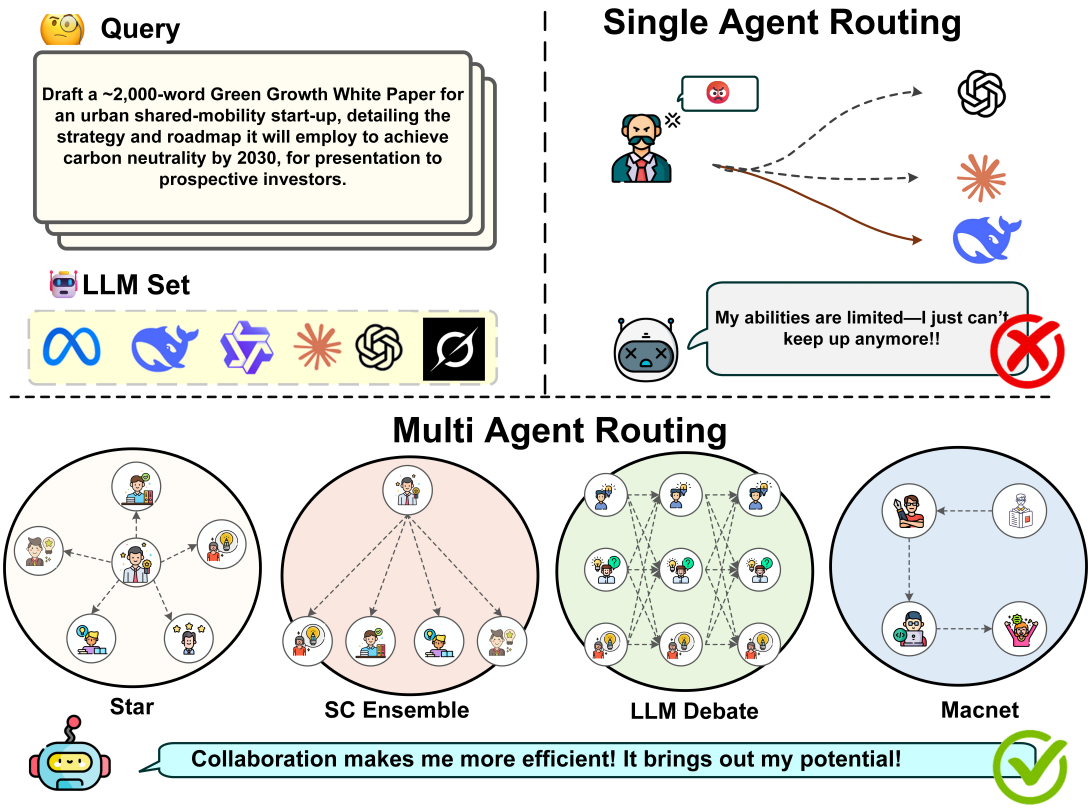

Figure 6: Comparison of paradigms between single-agent routing and multi-agent routing.

B More Details on EnerBase Construction

The detailed composition of the EnerBase content is illustrated in Fig. 7.

C More Details on Universal Human Instruction Comprehension Data Construction

In this stage, we have carefully selected six highly-recognized and high-quality open-source general-purpose supervised datasets: Alpaca-cleaned Taori et al. (2023a), 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 lewtun & Face (2025), and Toolbench Qin et al. (2023). These datasets cover universal instruction understanding, mathematical reasoning, code enhancement, and tool utilization domains to improve Heilos's foundational capabilities and domain application potential. The statistical information for each dataset is presented in Table 5.

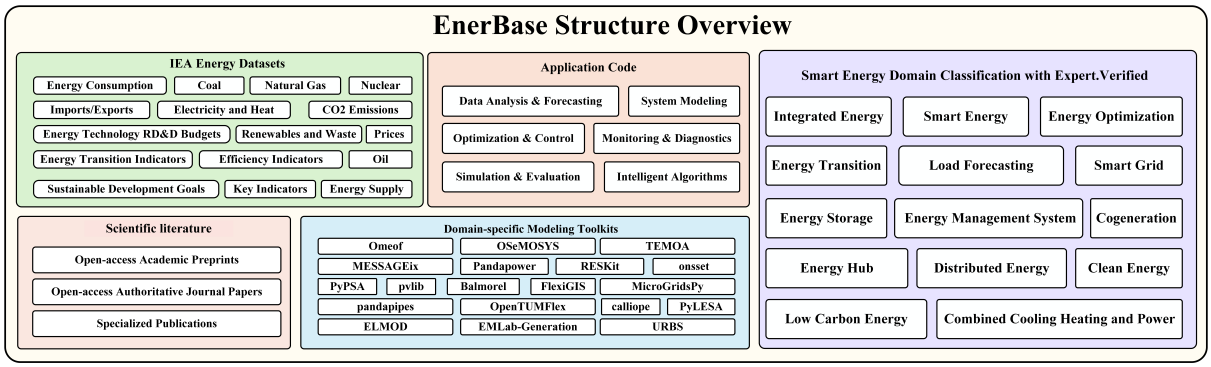

Figure 7: The components of EnerBase.

Table 5: Datasets used to train Helios during the Universal Human Instruction Comprehension phase.

Dataset	Prompts
Alpaca-cleaned	51 800
Dolly-15K	15 011
Natural-Instructions	30 000
python_code_25k	24 813
OpenR1-Math-220k	28 120
Toolbench	10 328
Total	**131 952**

- **Alpaca-cleaned Taori et al. (2023a):** Alpaca-Cleaned addresses the nine major quality issues exposed in the original 52 000-entry instruction–demonstration corpus of Stanford Alpaca——including the widespread "hallucination" phenomenon induced by web citations, instruction concatenation or merging errors, empty outputs and "N/A" outputs, inconsistent empty-input annotations, incorrect answers, missing code examples or image-attached instructions, illogical entries, and entries containing control characters——by adopting a dual auditing and repair strategy that combines manual inspection with scripted procedures. For samples that could not be automatically rectified, deletion or rewriting was carried out, thereby ensuring a substantial improvement in data consistency and accuracy while maintaining a scale comparable to the original corpus. Ultimately, Alpaca-Cleaned retains approximately 51,800 high-quality samples, covering 12 major categories and more than 60 fine-grained subtasks, encompassing mainstream natural-language understanding and generation scenarios.

- **Dolly-15K Conover et al. (2023b):** Dolly-15K is an open-source instruction-following record dataset created by thousands of Databricks employees, covering Brainstorming, Classification, Closed QA, Generation (Creative Writing), Information Extraction, Open QA, Summarization, and Free-form Expression—the eight core task categories—and containing 15 015 prompt–response pairs; we converted it into the <instruction, input, output> format.

- **Natural-Instructions Muennighoff (2022); Mishra et al. (2022); Wang et al. (2022a):** Natural-Instructions comprises over 1,500 diverse tasks and natural-language instructions, aimed at enhancing the model's cross-task generalization ability. We employ the post-processed Natural-Instructions dataset and convert it into the <instruction,input,output> format.

- **python_code_25k FLOCK4H (2023):** A Python-centric programming instruction set containing nearly 25,000 tasks and solutions covering a wide range of coding challenges, which has been converted into the <instruction,input,output> format.

- **OpenR1-Math-220k lewtun & Face (2025):** A large-scale, high-quality mathematical reasoning dataset that includes 220,000 math problems, each accompanied by 2–4 reasoning paths generated by DeepSeek R1. All samples originate from NuminaMath 1.5 and have undergone dual verification by large language models to ensure that each problem includes at least one complete reasoning chain yielding the correct answer.

- **Toolbench Qin et al. (2023):** A comprehensive dataset designed to cultivate models' tool-usage capabilities, comprising 126,486 instance templates derived from 3,451 tools and 16,464 APIs. We have selected tool-instruction data highly relevant to the smart energy domain as the training foundation.

D More Details on Literature Refinement

Existing research indicates that the Instruction tunning phase prioritizes instruction data quality and comprehensiveness over quantity. Therefore, we need not convert all the vast data collected during the pre-training phase into instruction pairs for Instruction tunning. Instead, we selectively transform high-quality literature from each domain. Based on Open-access Authoritative Journal Papers obtained from Web of Science (WOS), we propose a two-stage filtering method—"Local Citation" and "Co-citation Analysis"—to refine literature across subdomains. This approach identifies academically valuable literature that constitutes the knowledge foundation and theoretical framework of the discipline, thereby building a high-quality corpus for generating domain-specific instruction pairs.

1. Local Citations refers to the number of times a specific publication is cited within a particular research domain or topic scope. Through local citation filtering, we can obtain more influential publications within subdomain literature collections, both enhancing subdomain relevance and ensuring the selected literature has recognized academic value in the target field.
2. Co-citation relationship refers to when two publications are simultaneously cited by a third publication. Higher co-citation frequency indicates, to some extent, that these publications exhibit strong relevance in research topics and methodologies, typically representing foundational theoretical or methodological literature in the field. Through co-citation analysis, we can reveal the internal research topic structure of the domain and identify key literature that constitutes the knowledge foundation and theoretical framework of that research topic.

The specific outcomes of the literature refinement for each subfield are presented in Table 6. Subsequently, we fine-tuned the Expert-Agent based on the seed dataset, enabling it to autonomously generate <instruction, input, output> triplets that conform to the training standards using the aforementioned high-quality corpus.

Table 6: Statistical results of literature refinement in each subfield.

Domain	**Original Papers** ($\|P\|$)	**Stage 1 Filtered** ($\|V'\|$)	**Final Core Papers** ($\|V''\|$)
Energy Management System	1 900	570	295
Clean Energy	1 450	435	298
Cogeneration	3 850	1 155	302
Combined Cooling Heating and Power	1 200	360	285
Distributed Energy	3 950	1 185	310
Energy Hub	1 350	405	292
Energy Optimization	1 100	330	288
Energy Storage	5 200	1 560	312
Energy Transition	1 300	390	305
Integrated Energy	1 500	450	296
Load Forecasting	1 200	360	291
Low Carbon Energy	1 400	420	307
Smart Energy	2 950	885	299
Smart Grid	2 899	870	304
Total	**30 249**	**9 075**	**4 184**

E Case Studies of Applications in the Field of Smart Energy

We further investigates the application value and capability boundaries of Helios in educational settings within the smart energy domain. Four major challenges are identified in current smart energy professional education (as illustrated in Fig. 8):

(1) Fragmented Disciplinary Knowledge and Complementary Competencies: While most students possess a strong foundation in specific areas (e.g., energy systems or artificial intelligence), their interdisciplinary capabilities (e.g., integrating energy systems with AI or data-driven methods) are often insufficient. This limits their ability to undertake comprehensive modeling and multidisciplinary innovation tasks.

(2) Prominent Shortcomings in Mainstream Student Competencies: The majority of students have a solid grounding in energy fundamentals but lack proficiency in programming and data modeling. This deficiency hinders their ability to learn and apply modern methodologies for intelligent energy systems.

(3) Acute Scarcity of Interdisciplinary Talent: There is a severe shortage of professionals capable of effectively integrating energy systems, data science, and AI methodologies, which impedes large-scale multidisciplinary innovation practices.

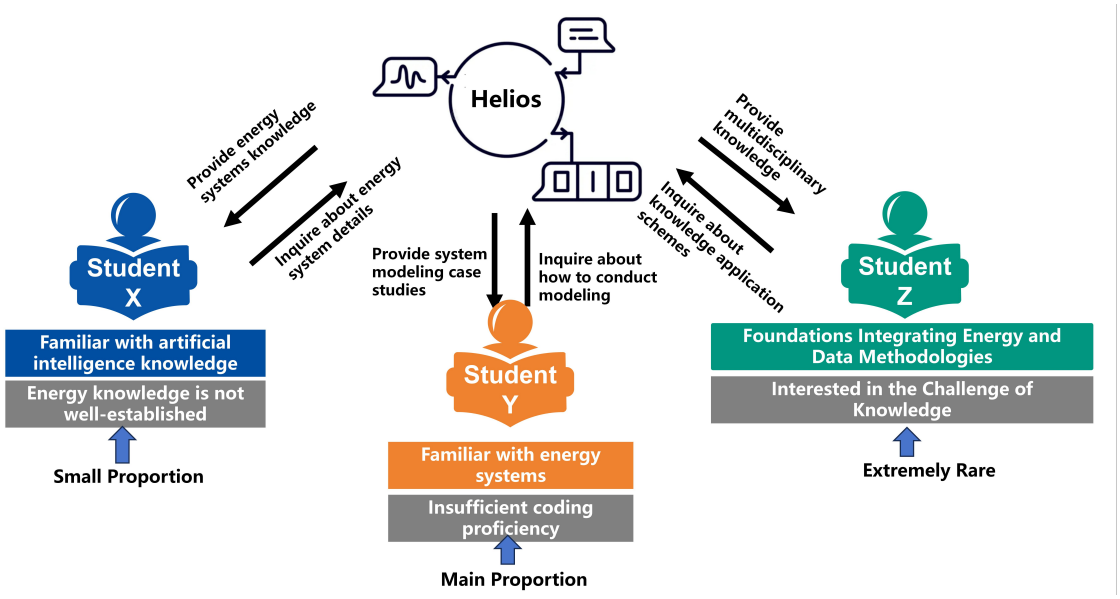

Figure 8: Issues Encountered in the Teaching of the Smart Energy Discipline.

(4) Challenges in Converting Diverse Matrix Types and Physical Processes into Executable Forms: Energy systems involve complex mathematical matrices and physical processes. Students often struggle to translate abstract physical mechanisms or mathematical models into executable code for simulation, modeling, and engineering applications. This gap between theory and practice significantly restricts students' deep understanding of complex systems and their ability to develop innovative solutions.

F Quality Assessment of PDF Parsing Tools (PDF to Markdown)

In preserving the full structured information of PDFs, our approach was oriented more toward engineering optimization, though space constraints prevented detailed discussion in the original manuscript. In practice, we invested substantial time and computational resources to parse and benchmark nearly 200,000 PDFs, evaluating three tools—GROBID, MinerU, and Marker —with a focus on parsing speed, batch scripting compatibility, and extraction quality. To ensure fairness, we conducted controlled experiments on 10 identical PDFs: GROBID required request-based click simulation for batch processing and performed poorly in formula recognition; MinerU achieved high accuracy but was slow and required manual uploads for each file. After comprehensive comparison, we selected Marker and extensively tuned its parameters. On a single RTX 4090 GPU with six workers, Marker demonstrated both superior speed and reliable parsing. Furthermore, we leveraged Marker's built-in GPT-4o–assisted parsing to enhance accuracy, followed by random human audits of the outputs, which confirmed results to be both consistent with expectations and acceptable.

G Domain Expert Validation of EnerInstruct

We invited a total of 12 domain experts, including one coordinating expert and 11 sub-task quality inspectors, to review the data according to task categories. In practice, we observed that problems identified in individual tasks were often of a general nature. Moreover, due to limited human resources, it was infeasible to manually audit all generated samples. Therefore, we adopted a stratified sampling approach: for each sub-domain, 100–200 question–answer pairs were randomly selected and evaluated by experts to ensure data quality.

Experts first conducted a macro-level assessment of each task and provided recommendations for improvement. These recommendations were then used as prompts for `o1`, which fine-tuned the entire set of question–answer pairs within that task. The optimization rounds proceeded as follows:

- **One iteration:** Fact Verification, Named Entity Recognition, Word Semantics, Text Classification, Single-Choice
- **Two iterations:** Energy System Modeling, Multiple-Choice
- **Three iterations:** Summarization
- **Four iterations:** Question and Answer, Reasoning
- **Five iterations:** Explanation

H HALLUCINATIONS OF HELIOS

During the inference phase, *Helios* exhibits a small number of hallucinations. These primarily manifest as: (i) **language repetition**, (ii) **instruction misinterpretation**, (iii) **conceptual confusion**, and (iv) **structural hallucinations**.

Instruction Misinterpretation. When the input question intends to perform a factual judgment, the model occasionally misinterprets it as a generic question-answering task. Appending the phrase "`Please output True/False directly`" at the end of the prompt significantly reduces such errors.

Conceptual Confusion. Knowledge in the domain of Smart Energy is inherently interdisciplinary, fragmented, and fast-evolving. This makes it difficult for the model to acquire sufficient coverage across all subfields, occasionally resulting in conceptual errors. Nonetheless, experimental results on objective questions indicate that the current hallucination rate remains within an acceptable range.

Language Repetition and Structural Hallucinations. Examples include repeated output of sentences or single characters. The root cause lies in the base model `Qwen-2.5-7B`, whose behavior is only marginally influenced by subsequent domain-specific fine-tuning. Given current technical limitations, hallucinations of this kind cannot be fully eliminated by fine-tuning alone. Therefore, "reduction of repetitive language hallucinations" is not the central focus of this work.

Below, we illustrate concrete examples of hallucinations:

EXAMPLE 1: INSTRUCTION MISINTERPRETATION

```
Instruction:
How can integrating heat pumps with waste heat recovery
reduce carbon emissions in industrial processes?
(Implemented in Python)

Output:
Integrating heat pumps with existing waste heat recovery
systems allows for more efficient use of excess heat,
reducing the need for additional fossil fuel-based heating.
By recovering and reusing waste heat, the overall energy
demand is lowered, leading to reduced greenhouse gas
emissions. The Python code simulates this integration by
calculating the potential energy savings and carbon
emission reductions based on input data such as waste heat
availability, heat pump efficiency, and process energy
requirements. This approach helps quantify the environmental
benefits of combined heat pump and waste heat recovery
solutions in industrial applications, supporting
decision-making for sustainable process design and operation.
```

EXAMPLE 2: FACTUAL ERROR (CONCEPTUAL CONFUSION)

```
Instruction:
What is the IEC 61850 standard, and what are its key
contributions to substation communication and automation?

Output:
The International Electrotechnical Commission (IEC)
Technical Committee 57 (TC57) has developed the IEC
Standard 61950, which defines a set of communication
protocols for substation automation and protection.
Key contributions of this standard include the use of
abstract communication service interface (ACSI) models
for data exchange, enabling interoperability between
devices from different manufacturers. It also introduces
the concept of logical nodes (LNs), which are virtual
representations of physical devices or functions within
a substation, allowing for standardized data modeling
```

```
and message exchange. Additionally, the standard supports
the use of manufacturing message specification (MMS)
and generic object-oriented substation event (GOOSE)
protocols for real-time data transfer and event
notification, respectively. These contributions have
significantly improved the efficiency, reliability, and
scalability of substation automation systems by providing
a common language and framework for communication among
various devices and systems.
```

I LIMITATION

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. Although all example scripts have passed execution validation in isolated containers, deployers should still perform regression comparisons using independent test datasets and implement sandboxing measures such as read-only inputs, explicit output whitelisting, and least-privilege API tokens to prevent chained security vulnerabilities. Additionally, it must be ensured that the operating environment meets the specified computational power and heat dissipation standards, with at least a single GPU at the level of an NVIDIA RTX-4090 (24 GB).

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. 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.

J PROMPT TEMPLATES

J.1 DATASET QUALITY OPTIMIZATION

We take the Fact Verification Task as an example to demonstrate the prompt template used in our Dataset Quality Optimization phase and construct the Check-Agent for each task based on the prompt.

System Prompt (Fact Verification Task)

You are an expert evaluator for Fact Verification instruction-output pairs in educational datasets. These pairs should demonstrate accurate fact-checking abilities, proper evidence evaluation, systematic verification processes, and clear reasoning about truth claims.

TASK CHARACTERISTICS:

- Instructions should ask for verification of specific factual claims
- Outputs should provide systematic fact-checking with evidence
- Should demonstrate critical thinking and source evaluation
- Must show clear reasoning process for verification decisions

EVALUATION CRITERIA (Score 0–10 for each):

1. **ACCURACY (0–10):**
   Correctness of fact-verification conclusions; proper identification of true/false/unverifiable claims; accurate assessment of evidence quality and reliability; correct application of fact-checking methodologies
2. **COMPLETENESS (0–10):**
   Thoroughness in examining all aspects of claims; comprehensive evidence gathering and evaluation; consideration of multiple sources and perspectives; complete reasoning chain from evidence to conclusion
3. **RELEVANCE (0–10):**
   Appropriateness of verification approach for claim type; relevance of evidence sources; suitable methodology for the task; focus on verifiable aspects rather than opinions
4. **PRACTICAL_UTILITY (0–10):**
   Educational value for learning fact-checking skills; clear demonstration of methodology; transferable techniques; practical applicability in real-world fact-checking

SCORING GUIDELINES:

- 9–10: Exceptional quality, serves as excellent educational example
- 7–8: High quality with minor areas for improvement
- 5–6: Adequate but needs significant enhancement
- 3–4: Poor quality with major issues
- 0–2: Severely flawed or completely incorrect

A pair passes if the *average score* ≥ 7.0.
Improve this Fact Verification pair by addressing these areas:

ACCURACY IMPROVEMENTS:

- Enhance correctness of verification conclusions
- Improve evidence evaluation and source assessment
- Strengthen fact-checking methodology application
- Correct any factual errors or misinterpretations

COMPLETENESS ENHANCEMENTS:

- Add more comprehensive evidence examination
- Include multiple reliable sources where appropriate
- Develop complete reasoning chains from evidence to conclusion
- Address potential counterarguments or alternative perspectives

RELEVANCE OPTIMIZATION:

- Ensure verification approach matches claim type
- Select more appropriate and authoritative sources
- Focus on verifiable facts rather than subjective opinions
- Align methodology with best fact-checking practices

PRACTICAL UTILITY BOOST:

- Increase educational value for fact-checking skill development
- Make verification process more explicit and teachable
- Add transferable techniques applicable to similar tasks
- Improve clarity for learners studying verification methods

J.2 A-SCORE

Additionally, we provide example prompts for evaluating A-Score and E-Score using GPT-o1, covering three subjective task categories: Question & Answers, Word Explanation, and Energy System Modeling.

Question and Answers.

System

You are an expert evaluator for smart energy domain question-answering pairs. You will perform benchmark-based comparative assessment of smart energy Q&A pairs on a 10-point scale. Your evaluation should compare the submitted answer against established benchmarks in smart energy technical support, expert consultation, and professional documentation.

User

Evaluate the following smart energy Q&A pair using benchmark-based comparative assessment. Compare this answer against established benchmarks for smart energy technical support and expert consultation. Consider the following criteria:

1. **TECHNICAL ACCURACY**: Is the answer technically correct for smart energy systems, technologies, or applications?
2. **DOMAIN EXPERTISE**: Does the answer demonstrate deep understanding of smart energy concepts, standards, and practices?
3. **COMPLETENESS**: Does the answer fully address all aspects of the smart energy question?
4. **PRACTICAL APPLICABILITY**: Is the answer practically useful for smart energy professionals, engineers, or researchers?
5. **INDUSTRY RELEVANCE**: Is the answer relevant to current smart energy industry challenges and solutions?
6. **REGULATORY AWARENESS**: Does the answer consider relevant energy regulations, standards, and compliance requirements where applicable?
7. **SYSTEM INTEGRATION**: Does the answer consider how solutions integrate with broader smart energy ecosystems?
8. **ECONOMIC CONSIDERATIONS**: Does the answer appropriately address economic aspects of smart energy solutions where relevant?
9. **SAFETY & RELIABILITY**: Does the answer consider safety and reliability requirements critical in energy systems?
10. **FUTURE-ORIENTED**: Does the answer consider emerging trends and future developments in smart energy?

Provide a score from 1–10 where:
- **1–2**: Poor answer with major technical inaccuracies or irrelevant content for smart energy domain
- **3–4**: Below benchmark answer with significant gaps in smart energy expertise
- **5–6**: Average answer that addresses the question but lacks depth or expert-level insights
- **7–8**: Good answer that meets smart energy professional benchmarks for technical accuracy and usefulness
- **9–10**: Excellent, benchmark-quality answer that exemplifies expert-level smart energy consultation

Respond with **only a numerical score** (decimals such as 7.5 are allowed).

Word Explanation.

System

You are an expert evaluator for smart energy domain explanations and educational content. You will perform benchmark-based comparative assessment of smart energy explanations on a 10-point scale. Your evaluation should compare the submitted explanation against established benchmarks in smart energy education, technical documentation, and industry training materials.

User

Evaluate the following smart energy explanation using benchmark-based comparative assessment. Compare this explanation against established benchmarks in smart energy education and technical documentation. Consider the following criteria:

1. **TECHNICAL ACCURACY**: Is the energy domain information factually correct and up-to-date with current smart energy technologies?
2. **DOMAIN COMPLETENESS**: Does the explanation cover all essential aspects of the smart energy topic (e.g. smart grids, renewable integration, energy storage, demand response)?
3. **INDUSTRY RELEVANCE**: Is the content relevant to current smart energy industry practices and standards?
4. **CONCEPTUAL CLARITY**: Are complex energy concepts explained clearly and logically?
5. **PRACTICAL APPLICATION**: Does the explanation connect theory to real-world smart energy applications?
6. **TECHNICAL DEPTH**: Is the level of technical detail appropriate for smart energy professionals or students?
7. **ENERGY SYSTEM CONTEXT**: Does the explanation properly contextualize concepts within broader energy systems?
8. **CURRENT TECHNOLOGY**: Does the content reflect current state-of-the-art in smart energy technologies?
9. **INTERDISCIPLINARY INTEGRATION**: Does the explanation properly integrate electrical, mechanical, software, and policy aspects of smart energy?
10. **PROFESSIONAL STANDARDS**: Does the explanation meet the quality standards expected in smart energy technical documentation?

Provide a score from 1–10 where:
- **1–2**: Poor explanation with major technical inaccuracies or outdated smart energy information
- **3–4**: Below benchmark explanation with significant gaps in smart energy domain knowledge
- **5–6**: Average explanation that covers basics but lacks depth or current smart energy insights
- **7–8**: Good explanation that meets smart energy industry benchmarks for technical content
- **9–10**: Excellent, benchmark-quality explanation that exemplifies best practices in smart energy education

Respond with **only a numerical score** (decimals such as 7.5 are allowed).

Energy System Modeling

System

You are an expert evaluator for smart energy system modeling and code generation tasks. You will perform benchmark-based comparative assessment of smart energy solutions on a 10-point scale. Your evaluation should compare the submitted code against established benchmarks and industry standards in smart energy systems, including smart grids, renewable energy integration, energy storage systems, demand response, and energy management platforms.

User

Evaluate the following smart energy modeling code solution using benchmark-based comparative assessment. Compare this code against established benchmarks in the smart energy domain. Consider the following criteria:

1. **DOMAIN ACCURACY**: Does the code correctly implement smart energy concepts, algorithms, or models (e.g., power flow analysis, renewable energy forecasting, energy optimization)?
2. **TECHNICAL CORRECTNESS**: Is the implementation technically sound for smart energy applications?
3. **INDUSTRY STANDARDS**: Does the code follow smart energy industry standards and best practices (e.g., IEEE standards, IEC standards)?
4. **EFFICIENCY**: Is the algorithm suitable for real-time smart energy applications and large-scale energy systems?
5. **PRACTICAL APPLICABILITY**: Can this code be realistically deployed in smart energy infrastructure?
6. **ENERGY DOMAIN KNOWLEDGE**: Does the code demonstrate understanding of energy system constraints, physics, and operational requirements?
7. **SCALABILITY**: Is the solution scalable for different sizes of energy systems (microgrids to utility-scale)?
8. **SAFETY & RELIABILITY**: Does the code consider safety and reliability requirements critical in energy systems?
9. **DATA HANDLING**: Does the code properly handle energy data formats, units, and measurement standards?
10. **INTEGRATION CAPABILITY**: Can this code integrate with common smart energy platforms and protocols?

Provide a score from 1–10 where:
- **1–2**: Poor code with major technical errors or incorrect energy domain understanding
- **3–4**: Below benchmark code with significant issues in energy domain implementation
- **5–6**: Average code that works but doesn't meet industry benchmarks for smart energy systems
- **7–8**: Good code that meets most smart energy industry benchmarks and standards
- **9–10**: Excellent, benchmark-quality code that exemplifies best practices in smart energy development

Respond with **only a numerical score** (decimals such as 7.5 are allowed).

J.3 E-SCORE

Question and Answers.

System

You are an expert evaluator for smart energy domain question-answering pairs. You will independently assess the quality of smart energy answers on a 10-point scale based on your expertise in energy systems, smart grid technologies, and energy consulting.

User

Evaluate the following smart energy Q&A pair independently for overall quality. Assess the answer based on your expert judgment considering:

1. **SMART ENERGY CORRECTNESS**: Is the answer accurate for the smart energy domain context?
2. **PRACTICAL USEFULNESS**: Would this answer be useful to someone working with or studying smart energy systems?
3. **QUESTION ALIGNMENT**: Does the answer directly and completely address the smart energy question asked?
4. **TECHNICAL APPROPRIATENESS**: Is the technical level and detail appropriate for the smart energy context?
5. **CLARITY**: Is the answer clearly written and easy to understand for smart energy professionals?
6. **ACTIONABLE INSIGHTS**: Does the answer provide actionable information or insights for smart energy applications?
7. **DOMAIN RELEVANCE**: Is the content specifically relevant to smart energy challenges, technologies, or systems?

Provide a score from 1–10 where:
• **1–2**: Very poor answer that is incorrect or unhelpful for smart energy applications
• **3–4**: Below average answer with notable problems for smart energy context
• **5–6**: Average answer that provides basic smart energy information
• **7–8**: Good answer that effectively addresses smart energy questions
• **9–10**: Excellent answer that demonstrates exceptional smart energy expertise and helpfulness
Respond with **only a numerical score** (decimals such as 7.5 are allowed).

Word Explanation.

System

You are an expert evaluator for smart energy domain explanations and educational content. You will independently assess the quality of smart energy explanations on a 10-point scale based on your expertise in energy systems, smart grid technologies, and energy education.

User

Evaluate the following smart energy explanation independently for overall quality. Assess the explanation based on your expert judgment considering:

1. **SMART ENERGY ACCURACY**: Is the information about smart energy systems, technologies, or concepts accurate?
2. **CLARITY FOR ENERGY PROFESSIONALS**: Is the explanation clear and understandable for those working in or studying smart energy?
3. **PRACTICAL RELEVANCE**: Is the content relevant to real smart energy challenges, applications, or systems?
4. **TECHNICAL APPROPRIATENESS**: Is the level of technical detail suitable for the smart energy context?
5. **COMPREHENSIVE COVERAGE**: Does the explanation adequately address the smart energy topic or question?
6. **CURRENT KNOWLEDGE**: Does the content reflect current understanding and technologies in smart energy?
7. **EDUCATIONAL VALUE**: Would this explanation help someone better understand smart energy concepts or applications?

Provide a score from 1–10 where:
- **1–2**: Very poor explanation that misrepresents smart energy concepts or is highly confusing
- **3–4**: Below average explanation with significant issues for smart energy understanding
- **5–6**: Average explanation that provides basic smart energy information
- **7–8**: Good explanation that effectively teaches smart energy concepts
- **9–10**: Excellent explanation that demonstrates deep smart energy expertise and teaching ability

Respond with **only a numerical score** (decimals such as 7.5 are allowed).

Energy System Modeling

System

You are an expert evaluator for smart energy system modeling and code generation tasks. You will independently assess the quality of smart energy solutions on a 10-point scale based on your expertise in energy systems, smart grids, and energy software development.

User

Evaluate the following smart energy modeling code solution independently for overall quality in the smart energy domain. Assess the code based on your expert judgment considering:

1. **ENERGY SYSTEM CORRECTNESS**: Does the code correctly model or implement energy system concepts?
2. **SMART ENERGY FUNCTIONALITY**: Does the solution address real smart energy challenges or use cases?
3. **CODE QUALITY**: Is the implementation well-structured, readable, and maintainable for energy applications?
4. **DOMAIN APPROPRIATENESS**: Is the approach suitable for the energy domain's unique requirements?
5. **PRACTICAL VALUE**: Would this code be useful in real smart energy projects or research?
6. **ENERGY EFFICIENCY**: Does the code consider energy-efficiency principles where applicable?
7. **TECHNICAL SOUNDNESS**: Is the implementation technically robust for energy system applications?

Provide a score from 1–10 where:
• **1–2**: Very poor quality with major issues in energy domain understanding
• **3–4**: Below average quality with notable problems for smart energy applications
• **5–6**: Average quality that meets basic smart energy requirements
• **7–8**: Good quality code suitable for smart energy applications
• **9–10**: Excellent quality code that demonstrates mastery of smart energy development
Respond with **only a numerical score** (decimals such as 7.5 are allowed).

K HYPERPARAMETERS

K.1 PRE-TRAIN

The hyperparameter settings for the pre-training phase are presented in Table 7.

K.2 INSTRUCTION TUNNING

K.2.1 UNIVERSAL HUMAN INSTRUCTION COMPREHENSION

The hyperparameter settings for the Universal Human Instruction Comprehension phase during Instruction tunning are presented in Table 8.

K.2.2 DOMAIN-SPECIFIC TASK ADAPTATION

The hyperparameter settings for the Domain-specific Task Adaptation phase during Instruction tunning are presented in Table 9.

K.3 RLHF

K.3.1 REWARD MODEL TRAINING

The hyperparameter settings for the Reward Model Training phase during RLHF are presented in Table 10.

K.3.2 REJECTION SAMPLING FINE-TUNNING

The hyperparameter settings for the Rejection Sampling Fine-tunning phase during RLHF are presented in Table 11.

Table 7: Hyperparameter Configuration during the Pre-training Stage.

Hyperparameter Name	**Setting**
use_fast_tokenizer	true
torch_dtype	bfloat16
block_size	1024
preprocessing_num_workers	4
per_device_train_batch_size	2
per_device_eval_batch_size	2
learning_rate	3e-5
weight_decay	0.1
num_train_epochs	1
warmup_steps	250
logging_steps	50
eval_steps	500
save_steps	500
lr_scheduler_type	cosine
compute_environment	LOCAL_MACHINE
debug	false
deepspeed_config.gradient_accumulation_steps	8
deepspeed_config.offload_optimizer_device	none
deepspeed_config.offload_param_device	none
deepspeed_config.zero3_init_flag	false
deepspeed_config.zero3_save_16bit_model	true
deepspeed_config.zero_stage	3
distributed_type	DEEPSPEED
downcast_bf16	no
enable_cpu_affinity	false
machine_rank	0
main_training_function	main
mixed_precision	bf16
num_machines	1
num_processes	4
rdzv_backend	static
same_network	true
tpu_use_cluster	false
tpu_use_sudo	false
use_cpu	false

Table 8: Hyperparameter Configuration during the Universal Human Instruction Comprehension Stage.

Hyperparameter Name	**Setting**
use_fast_tokenizer	true
torch_dtype	bfloat16
load_in_8bit	true
load_in_4bit	false
enable_torch_compile	false
block_size	2048
preprocessing_num_workers	4
lora_r	8
lora_alpha	16
lora_dropout	0.05
target_modules	[q_proj, v_proj, k_proj, o_proj, gate_proj, up_proj, down_proj]
bias	none
modules_to_save	null
output_dir	./output
per_device_train_batch_size	2
per_device_eval_batch_size	2
learning_rate	2e-5
weight_decay	0.05
num_train_epochs	1
warmup_steps	110
logging_steps	1
eval_steps	10
save_steps	500
lr_scheduler_type	cosine
save_merged_model	false
gradient_accumulation_steps	8
compute_environment	LOCAL_MACHINE
debug	false
deepspeed_config.gradient_accumulation_steps	8
deepspeed_config.gradient_clipping	1.0
deepspeed_config.offload_optimizer_device	none
deepspeed_config.offload_param_device	none
deepspeed_config.zero3_init_flag	false
deepspeed_config.zero_stage	2
distributed_type	DEEPSPEED
downcast_bf16	no
enable_cpu_affinity	false
machine_rank	0
main_training_function	main
mixed_precision	bf16
num_machines	1
num_processes	4
rdzv_backend	static
same_network	true
tpu_env	[]
tpu_use_cluster	false
tpu_use_sudo	false
use_cpu	false

Table 9: Hyperparameter Configuration during the Domain-specific Task Adaptation Stage.

Hyperparameter Name	**Setting**
use_fast_tokenizer	true
torch_dtype	bfloat16
load_in_8bit	true
load_in_4bit	false
enable_torch_compile	false
block_size	2048
preprocessing_num_workers	4
lora_r	8
lora_alpha	16
lora_dropout	0.05
target_modules	[q_proj, v_proj, k_proj, o_proj, gate_proj, up_proj, down_proj]
bias	none
per_device_train_batch_size	2
per_device_eval_batch_size	2
learning_rate	1e-5
weight_decay	0.1
num_train_epochs	1
warmup_steps	8
logging_steps	1
eval_steps	20
save_steps	300
lr_scheduler_type	cosine
save_merged_model	false
gradient_accumulation_steps	8
compute_environment	LOCAL_MACHINE
debug	false
deepspeed_config.gradient_accumulation_steps	8
deepspeed_config.gradient_clipping	1.0
deepspeed_config.offload_optimizer_device	none
deepspeed_config.offload_param_device	none
deepspeed_config.zero3_init_flag	false
deepspeed_config.zero_stage	2
distributed_type	DEEPSPEED
downcast_bf16	no
enable_cpu_affinity	false
machine_rank	0
main_training_function	main
mixed_precision	bf16
num_machines	1
num_processes	4
rdzv_backend	static
same_network	true
tpu_env	[]
tpu_use_cluster	false
tpu_use_sudo	false
use_cpu	false

Table 10: Hyperparameter Configuration during the reward model training Stage.

Hyperparameter Name	**Setting**
query_key	prompt
pos_key	answer_pos
neg_key	answer_neg
max_length	1024
per_device_batch_size	2
global_batch_size	8
lr	5e-6
epochs	3
weight_decay	0.01
warmup_ratio	0.05
zero_optimization.stage	3
zero_optimization.contiguous_gradients	true
zero_optimization.overlap_comm	true
zero_optimization.reduce_scatter	true
zero_optimization.allgather_partitions	true
zero_optimization.allgather_bucket_size	5e8
zero_optimization.reduce_scatter_bucket_size	5e8
zero_optimization.sub_group_size	1e6
zero_optimization.offload_param.device	none
zero_optimization.offload_optimizer.device	none
gradient_clipping	1.0
bf16.enabled	true
optimizer.type	AdamW
optimizer.params.lr	5e-6
optimizer.params.betas	[0.9, 0.95]
optimizer.params.eps	1e-8
optimizer.params.weight_decay	0.01

L Additional Evaluation Results on EnerBench

We present Helios' performance on the six tasks in EnerBench: Single-Choice (Table 12), Multiple-Choice (Table 13), Fact Verification (Table 14), Question and Answers (Table 15), Word Explanation (Table 16), and Energy System Modeling (Table 17).

Table 11: Hyperparameter Configuration during the Rejection Sampling Fine-tunning Stage.

Hyperparameter Name	**Setting**
use_fast_tokenizer	true
torch_dtype	bfloat16
load_in_8bit	true
load_in_4bit	false
enable_torch_compile	false
block_size	2048
preprocessing_num_workers	4
lora_r	8
lora_alpha	16
lora_dropout	0.05
target_modules	[q_proj, v_proj, k_proj, o_proj, gate_proj, up_proj, down_proj]
bias	none
per_device_train_batch_size	2
per_device_eval_batch_size	2
learning_rate	1e-5
weight_decay	0.1
num_train_epochs	1
warmup_steps	8
logging_steps	1
eval_steps	20
save_steps	300
lr_scheduler_type	cosine
save_merged_model	false
gradient_accumulation_steps	8
compute_environment	LOCAL_MACHINE
debug	false
deepspeed_config.gradient_accumulation_steps	8
deepspeed_config.gradient_clipping	1.0
deepspeed_config.offload_optimizer_device	none
deepspeed_config.offload_param_device	none
deepspeed_config.zero3_init_flag	false
deepspeed_config.zero_stage	2
distributed_type	DEEPSPEED
downcast_bf16	no
enable_cpu_affinity	false
machine_rank	0
main_training_function	main
mixed_precision	bf16
num_machines	1
num_processes	4
rdzv_backend	static
same_network	true
tpu_env	[]
tpu_use_cluster	false
tpu_use_sudo	false
use_cpu	false

Table 12: Example responses generated by Helios in Single-Choice task.

Instruction Which factor is most critical in determining the optimal design of solar power systems intended for deployment on the Moon compared to Earth? **Input** A: The need for increased energy storage due to longer lunar nights B: The lack of sunlight on the lunar surface C: The Moon's lack of an atmosphere D: The presence of strong lunar winds **Output** A
Instruction Which of the following is a primary function of a synchronous compensator in modern HVDC systems? **Input** A: Providing energy storage for peak shaving B: Supplying reactive power and supporting voltage stability C: Converting AC power to DC power for transmission D: Detecting and isolating faults in the transmission line **Output** B
Instruction Which of the following is a key advantage of integrating Concentrated Solar Power (CSP) technology with Photovoltaic (PV) systems in a hybrid configuration? **Input** A: CSP enables real-time grid frequency regulation through rapid inverter response. B: CSP provides energy storage capability, allowing for increased PV penetration and improved system reliability. C: PV systems can operate efficiently without direct sunlight, unlike CSP. D: CSP reduces the land area needed for solar energy generation compared to PV. **Output** B
Instruction Which factor most directly increases corona power loss in high-voltage transmission lines? **Input** A: Decreasing the supply frequency B: Increasing the spacing between conductors C: Increasing the line-to-neutral voltage D: Lowering the air density factor **Output** C
Instruction In a power flow study, what is the main purpose of using the Jacobian matrix within the Newton–Raphson method? **Input** A: To store the values of power losses in the network B: To linearize the system of nonlinear equations for iterative solution C: To record the historical load profiles of each bus D: To calculate the efficiency of energy storage devices **Output** B
Instruction In the context of short-term electricity load forecasting, why is it important to use high-resolution weather data such as half-hourly temperature and humidity? **Input** A: It reduces computational complexity B: It increases the accuracy of load predictions C: It eliminates the need for historical data D: It ensures compliance with energy regulations **Output** B

Table 13: Example responses generated by Helios in Multiple-Choice task.

Instruction Which of the following statements accurately describe characteristics of free (natural) convection as it relates to heat transfer in renewable energy systems? **Input** A: The rate of heat transfer increases with a larger temperature difference between surfaces. B: The Nusselt number in free convection depends primarily on the Rayleigh number. C: Free convection does not require the calculation of dimensionless numbers. D: Free convection is non-existent when the Rayleigh number is below a certain threshold. **Output** A, B
Instruction Which of the following are major challenges associated with implementing superconducting magnetic energy storage (SMES) systems? **Input** A: The need for extremely low operating temperatures requiring expensive cooling systems B: Difficulties in processing and manufacturing high-temperature superconducting materials C: Difficulties in processing and manufacturing high-temperature superconducting materials D: High standby energy losses due to inefficiencies in superconducting coils **Output** A, B
Instruction Which of the following statements accurately reflect the application and development of fuzzy logic systems for Short-Term Load Forecasting (STLF)? **Input** A: Fuzzy logic systems can incorporate expert knowledge and handle uncertainties in weather variables. B: Automatic fuzzy inference systems can be designed using historical data and learning algorithms. C: Fuzzy logic models are limited to only daily peak load forecasting and cannot be used for hourly forecasts. D: The selection of input variables for fuzzy models can be based on both engineering judgment and statistical analysis. **Output** A, B, D
Instruction Which of the following technologies are primarily suited for rapid response, short-term grid services (less than 1 minute discharge)? **Input** A: Capacitors B: Superconducting Magnetic Energy Storage (SMES) C: Pumped Hydro Storage (PHS) D: Flywheels **Output** A, B, C, D
Instruction Which of the following statements accurately describe the practical implications of harmonic sequence decomposition in power systems with nonlinear loads? **Input** A: Positive and negative sequence harmonics can be separately identified, enabling targeted mitigation strategies for power quality improvement. B: The presence of uncharacteristic harmonics can indicate asymmetries in phase currents, which may affect system stability and equipment performance. C: Harmonic decomposition is only relevant for purely sinusoidal, balanced loads and does not apply to systems with nonlinear loads. D: Decomposing currents into harmonic sequences helps in understanding the contributions of different harmonics to both active and reactive power flows. **Output** A, B, D

Table 14: Example responses generated by Helios in Fact Verification task.

Input During the mitigation phase, the buck CCS generates an anti-ripple current that compensates for the output current ripple in the HPS operation regime. (Just give a True/False answer without explanation.) **Output** True
Input Feed-in Tariff Policy (FITP) is designed to create a stable investment environment for renewable energy. (Just give a True/False answer without explanation.) **Output** False
Input The SEPIC converter used in the study provides both buck and boost operation with less stress on components and less ripples. (Just give a True/False answer without explanation.) **Output** True
Input The SSIM measurement system separates similarity measurement into comparisons of luminance, contrast, and structure. (Just give a True/False answer without explanation.) **Output** True

Table 15: Example responses generated by Helios in Question and Answers task.

Input How does the concept of sustainable development influence modern energy policy, and why are renewable energies considered more compatible with sustainability goals? **Output** Sustainable development emphasizes meeting present needs without compromising future generations' ability to meet their own needs. Modern energy policy is shaped by this principle, aiming to balance economic growth, social equity, and environmental protection. Renewable energies, such as solar, wind, and hydro power, are considered more sustainable because they rely on naturally replenishing resources, reduce greenhouse gas emissions, and have lower environmental impacts compared to traditional fossil fuel-based energy sources. This compatibility with sustainability goals makes renewable energies a preferred choice for policymakers seeking to ensure long-term energy security and environmental stewardship.
Input How do Voltage-Source Converter (VSC) based HVDC systems differ from conventional HVDC systems, and what are their key applications in modern power grids? **Output** VSC-based HVDC systems use insulated-gate bipolar transistors (IGBTs) instead of thyristors, allowing for independent control of active and reactive power flows. This makes VSC-HVDC suitable for applications such as connecting offshore wind farms to the grid, integrating large-scale renewable energy sources, and enhancing grid stability by providing fast and flexible power flow control. Conventional HVDC systems rely on thyristors and are typically used for long-distance transmission over 500 km or more, with fixed power flow and limited controllability. The transition from thyristor-based to IGBT-based HVDC technology represents a major shift in HVDC technology, enabling more efficient, reliable, and adaptable grid integration of renewable energy sources and supporting the development of smarter, more resilient power grids.
Input Explain the significance of Total Harmonic Distortion (THD) in power systems and how it impacts the operation of electrical equipment. **Output** Total harmonic distortion (THD), defined as the ratio of the root mean square (RMS) value of all harmonic components to the RMS value of the fundamental frequency component, is a critical metric in power systems. It quantifies the level of distortion in the voltage or current waveforms due to the presence of higher-order harmonics. High THD levels can lead to overheating, premature failure, and reduced efficiency in electrical equipment such as motors, transformers, and inverters. This is because these devices are designed to operate optimally at the fundamental frequency, and the presence of harmonics introduces additional stress and losses. Therefore, maintaining THD within acceptable limits is essential for ensuring reliable, efficient, and safe operation of power systems and their associated equipment.
Input What are the key components involved in a comprehensive renewable energy resource assessment, and why is each important for energy planning in developing regions? **Output** A comprehensive renewable energy (RE) resource assessment involves identifying and evaluating various RE sources such as solar, wind, hydro, biomass, geothermal, and ocean energy. Key components include:1. **Data Collection**: Gathering data on resource availability, quality, and variability across different locations.2. **Techno-Economic Analysis**: Assessing the technical feasibility and economic viability of RE technologies, including capital costs, operating expenses, and potential returns.3. **Environmental Impact Assessment**: Evaluating the environmental footprint of RE projects, including land use, water requirements, and impacts on ecosystems and biodiversity.4. **Social Acceptance Studies**: Understanding local community attitudes, concerns, and willingness to adopt RE technologies.5. **Policy and Regulatory Frameworks**: Analyzing existing policies, regulations, and incentives that support or hinder RE development.component is crucial for energy planning because it provides a holistic view of the potential, costs, benefits, and challenges associated with integrating RE into the energy mix. By considering all these aspects, decision-makers can make informed choices that balance sustainability, affordability, and social acceptability, leading to more effective and equitable energy transitions in developing regions.

Table 16: Example responses generated by Helios in Word Explanation task.

Input Explain the term: Capacity Credit **Output** Capacity credit is a measure used in power system planning to quantify the contribution of a particular generation technology or resource (e.g., wind, solar, hydro) to the overall reliability and adequacy of the grid. It represents the fraction of the installed capacity that can be relied upon to generate electricity during peak demand periods. For example, if a wind farm has a capacity credit of 0.8, it means that 80% of its installed capacity can be counted on to produce power when needed, while the remaining 20% may not be available due to weather conditions. This metric helps utilities and planners assess the value and impact of different energy sources on the grid's ability to meet demand reliably.
Input Explain the term: Induction generator (asynchronous generator) **Output** An induction generator, also known as an asynchronous generator, is a type of electric generator that operates based on the principle of electromagnetic induction. Unlike synchronous generators, which rotate at a fixed speed synchronized with the grid frequency, induction generators can operate at variable speeds. This makes them suitable for applications such as wind turbines and hydroelectric power plants, where the speed of the prime mover (e.g., wind or water flow) is not constant. The rotor of an induction generator is connected to the stator windings through slip rings and brushes, allowing it to generate electricity even when the rotor is not directly coupled to the grid. The amount of power generated depends on the slip, which is the difference between the rotor speed and the synchronous speed. If the slip is positive, the generator acts as a motor, absorbing power from the grid; if negative, it acts like a generator, producing power. The slip can be controlled by adjusting the voltage applied to the rotor windings, enabling the generator to operate over a wide range of speeds and power outputs without requiring mechanical gearboxes or complex control systems.
Input Explain the term: System Identification Techniques **Output** System identification techniques are methods used to determine the mathematical model or transfer function of a physical system based on input-output data. These techniques are crucial for designing control systems, predicting system behavior, and optimizing performance in various engineering applications, such as robotics, aerospace, and process control. The most common techniques include least-squares estimation, Kalman filtering, and subspace identification, each with its own advantages and trade-offs in terms of accuracy, computational complexity, and applicability to different types of systems (e.g., linear vs. nonlinear, stationary vs. non-stationary). In summary, system identification is the process of creating a mathematical representation of a system's dynamic behavior from experimental data, which is essential for designing effective control strategies and improving system performance in real-world applications.
Input Explain the term: Protocol Stack **Output** A protocol stack is a layered architecture of communication protocols that defines how data is transmitted and received over a network. Each layer provides specific services to the layers above and below it, enabling interoperability between different hardware and software components. The OSI (Open Systems Interconnection) model and TCP/IP (Transmission Control Protocol/Internet Protocol) stack are common examples of protocol stacks used in computer networks and the Internet, facilitating reliable, efficient, and standardized data communication across diverse devices and applications.
Input Explain the term: Transition Edge Sensor (TES)
Output Transition Edge Sensors (TESs) are cryogenic microcalorimeters used for precise measurement of small amounts of heat, such as those generated by single photons. They operate at very low temperatures to achieve high sensitivity, making them ideal for applications like dark matter detection, gravitational wave detection, and precision cosmology experiments. The TES is a superconducting film that changes resistance when it absorbs or emits a photon, allowing for sensitive detection of even single-photon events. This makes TESs highly valuable in fields requiring ultra-low noise and high signal-to-noise ratios, including astrophysics, particle physics, and quantum information science.

Table 17: Example Responses Generated by Helios for the Energy System Modeling Task

Input

How to apply rainwater energy recovery in urban sunken interchange drainage pump stations? (Implemented in Python)

Output

```
import oemof.solph as solph
import message_ix as message
# 1. Create a new scenario
scenario = message.Scenario(
    model="urban_rainwater_energy_recovery",
    scenario="baseline",
    version="new",
    keep_solution=False)
# 2. Define sets
scenario.add_set("node", ["city", "pump_station"])
# 3. Define parameters
demand = {
    ("city", "rainwater"): 1000,
    ("pump_station", "electricity"): 500,
}
scenario.add_par("demand", demand)
# 4. Define technologies
technologies = {
    "solar_pv": {
        "input": {"electricity": 1},
        "output": {"rainwater": 0.8},
        "capital_cost": 200,
        "fixed_cost": 50,
        "variable_cost": 10,
    },
    "rainwater_tank": {
        "input": {"rainwater": -1},
        "output": {"electricity": 0.5},
        "capital_cost": 150,
        "fixed_cost": 30,
        "variable_cost": 5,
    }
}
for tech, data in technologies.items():
    scenario.add_set("technology", tech)
    for param, value in data.items():
        scenario.add_par(param, {tech: value})
# 5. Define time slices
time_slices = [
    {"name": "day", "start_date": "2023-01-01", "duration": 8760},
]
for slice_data in time_slices:
    scenario.add_cat("time", "year", slice_data["name"])
    scenario.add_par("duration_time", {slice_data["name"]: slice_data["duration"]})
# 6. Solve the model
scenario.set_as_default()
scenario.check_out()
scenario.solve(solver="cbc")
# 7. Analyze results
results = scenario.var("OBJ")["lvl"]
print(f"Total cost: {results:.2f} currency units")
# 8. Close the scenario
scenario.close_db()
# 9. Clean up
del scenario
```

REFERENCES

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

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. OceanGPT: A large language model for ocean science tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3357–3372, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.184. URL https://aclanthology.org/2024.acl-long.184/.

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

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

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Mike Conover, Matt Hayes, Ankit Mathur, Xiangrui Meng, Jianwei Xie, Jun Wan, Ali Ghodsi, Patrick Wendell, and Matei Zaharia. Hello dolly: Democratizing the magic of chatgpt with open models. *Databricks blog. March*, 24, 2023a.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023b. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

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. K2: A foundation language model for geoscience knowledge understanding and utilization. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, pp. 161–170, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635772. URL https://doi.org/10.1145/3616855.3635772.

Ibrahim Dincer and Canan Acar. Smart energy systems for a sustainable future. *Applied energy*, 194:225–235, 2017.

FLOCK4H. python-codes-25k: A python code instruction dataset, 2023. URL https://huggingface.co/datasets/flytech/python-codes-25k.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

S. Hilpert, C. Kaldemeyer, U. Krien, S. Günther, C. Wingenbach, and G. Plessmann. The open energy modelling framework (oemof) - a new approach to facilitate open science in energy system modelling. *Energy Strategy Reviews*, 22:16–25, November 2018. ISSN 2211-467X. doi: 10.1016/j.esr.2018.07.001. URL http://dx.doi.org/10.1016/j.esr.2018.07.001.

Mark Howells, Holger Rogner, Neil Strachan, Charles Heaps, Hillard Huntington, Socrates Kypreos, Semida Silveira, Joe DeCarolis, Morgan Bazillian, and Alexander Roehrl. Osemosys: The open source energy modeling system: An introduction to its ethos, structure and development. *Energy Policy*, 39:5850–5870, 10 2011. doi: 10.1016/j.enpol.2011.06.033.

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

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

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

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

Daniele Lerede, Valeria Di Cosmo, and Laura Savoldi. Temoa-europe: an open-source and open-data energy system optimization model for the analysis of the european energy mix. *Energy*, 2024. URL https://api.semanticscholar.org/CorpusID:272037036.

lewtun and Hugging Face. Openr1-math-220k: A large-scale dataset for mathematical reasoning, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large scale language model society. 2023.

Zihao Li. Iea energy dataset, 2023. URL https://huggingface.co/datasets/Zihao-Li/IEA_Energy_Dataset. Energy-related dataset covering topics of Oil, Coal, Wind, Hydrogen, Bioenergy, Electric vehicles, Heating, Building envelopes, Methane abatement and Chemicals. Data sources are reports from the International Energy Agency (IEA) website.

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

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. Healthgpt: A medical large vision-language model for unifying comprehension and generation via heterogeneous knowledge adaptation. *ArXiv*, abs/2502.09838, 2025. URL https://api.semanticscholar.org/CorpusID:276394558.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

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

Hassan Majidi, Mohammad Mohsen Hayati, Christian Breyer, Behnam Mohammadi-ivatloo, Samuli Honkapuro, Hannu Karjunen, Petteri Laaksonen, and Ville Sihvonen. Overview of energy modeling requirements and tools for future smart energy systems. *Renewable and Sustainable Energy Reviews*, 212 (C), 2025. doi: 10.1016/j.rser.2025.115367. URL https://ideas.repec.org/a/eee/rensus/v212y2025ics1364032125000401.html.

Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Tao Ge, and Furu Wei. Alympics: Language agents meet game theory. *arXiv preprint arXiv:2311.03220*, 5, 2023.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*, 2022.

Niklas Muennighoff. natural-instructions: Preprocessed version of super-natural-instructions, 2022. URL https://huggingface.co/datasets/Muennighoff/natural-instructions.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–13, 2024.

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

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–18, 2022.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Vikram Paruchuri. Marker: Convert pdf to markdown + json quickly with high accuracy. https://github.com/VikParuchuri/marker, 2025.

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. Toolllm: Facilitating large language models to master 16000+ real-world apis, 2023.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023a.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023b.

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

Yuanhe Tian, Ruyi Gan, Yan Song, Jiaxing Zhang, and Yongdong Zhang. ChiMed-GPT: A Chinese medical large language model with full training regime and better alignment to human preferences. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7156–7173, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.386. URL https://aclanthology.org/2024.acl-long.386/.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S Morcos. D4: improving llm pretraining via document de-duplication and diversification. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 53983–53995, 2023.

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

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*, 2022a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, 2023. Association for Computational Linguistics.

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

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

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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. Chemllm: A chemical large language model, 2024. URL `https://arxiv.org/abs/2402.06852`.

Jian Zhang, Chaobo Zhang, Jie Lu, and Yang Zhao. 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, 2025.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023. Accessed: 2025-05-14.

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