Internship Project
Report

Paper Idea Generation Pipeline

Arthur Nürnberg
Bachelor of Science Candidate in Computer Science
Technical University of Dresden

Under the guidance of

Dr. Pooyan Safari
Research Scientist, Digital Signal Processing

Photonic Network and Systems Department
Fraunhofer Institute for Telecommunications,
Heinrich-Hertz-Institut, HHI
Einsteinufer 17
10587 Berlin, Germany

Summer Internship 2024

Abstract

This report outlines the development of a **Paper Idea Generation Pipeline** undertaken during my internship at the Digital Signal Processing group within the Photonic Networks and Systems Department at Fraunhofer Heinrich Hertz Institute (HHI). The primary objective was to create an automated system leveraging Large Language Models (LLMs) to generate novel and feasible research ideas, specifically aimed at enhancing the brainstorming and idea development processes within the institute.

The pipeline integrates both open-source and closed-source LLMs, including Llama 3.1, Mistral, GPT-4o, the new Qwen 2.5, Gemma and Sonnet 3.5, and employs advanced prompt engineering techniques including Chain-of-Thought, Reflection, and Meta Prompting. Key components of the pipeline encompass idea generation, novelty checking via the Semantic Scholar API, detailed write-ups of selected ideas, and iterative reviews for refinement.

During the internship, the pipeline successfully generated over 100 unique and innovative research ideas in the field of optical networks and systems. The project provided valuable insights into the capabilities and limitations of current LLMs, highlighting the significant impact of prompt engineering on output quality. Challenges encountered included handling inconsistencies in model outputs, especially with smaller LLMs failing to conform to expected formats as well as the feasibility of the generated ideas.

The experience culminated in the development of a functional tool that not only streamlines the idea generation process but also enhances the potential for innovation within the institute. The report concludes with reflections on the skills acquired, such as advanced Python programming, API integration, and remote server management, and offers perspectives on future enhancements to the pipeline, including improved error handling, LaTeX formatting and broader applicability across different research domains.

Acknowledgements

I would like to extend my heartfelt gratitude to everyone who made my internship at the Fraunhofer Heinrich Hertz Institute both possible and enriching.

Firstly, I am deeply thankful for the opportunity to undertake this internship at such a credible and reputable institute. The environment here has been instrumental in fostering my professional growth and has provided me with invaluable experiences.

I would like to express my sincere appreciation to my colleague, Dr. Pooyan Safari, for his support, guidance, and insightful feedback throughout the project. Additionally, I am grateful to the entire Digital Signal Processing team for their collaboration.

I am particularly appreciative of the freedom granted to me in developing the research idea generation pipeline. This autonomy allowed me to explore innovative approaches and tailor the project to align with both my interests and the institute's goals.

This internship has been a remarkable learning experience, and I am deeply appreciative of the opportunities and support I have received.

Contents

Chapter 1

Introduction

The aim of this report is to provide an overview of my internship experience at the Digital Signal Processing group in the Photonic Networks and Systems Department at Fraunhofer Heinrich Hertz Institute (HHI), which took place from August 21, 2024 to September 30, 2024. The internship involved working on Machine Learning, with a focus on Large Language Models (LLMs).

The primary objective of the internship was the application of LLMs on HHI-owned Servers with dedicated GPUs. This work provided valuable insight into working on SSH servers, programming with Language Model APIs and Prompt Engineering. A pipeline was developed to generate ideas for academic papers, specifically tailored to brainstorming and idea development at the institute.

This work was based on the newly published paper, The AI Scientist [Lu et al., 2024], which aims to autonomously conduct scientific research by generating machine learning ideas, performing experiments, writing LaTeX documents, and improving through self-review with a Pipeline that leverages Large Language Models. [Minaee et al., 2024, Naveed et al., 2024].

This report underlines the tasks and responsibilities undertaken during the internship, followed by a detailed explanation of the Paper Idea Generation Pipeline. The report will conclude with a reflection on the outcomes, the developed skills, and an outlook on how the internship contributed to academic progress.

Motivation

In today's rapidly evolving technological landscape, generating innovative and relevant research ideas is an interesting research topic.

The upcoming of Large Language Models (LLMs), such as GPT-4, Llama 3 and other advanced AI systems, presents new opportunities to support and optimize this process. LLMs have the potential to significantly accelerate and enrich idea generation through their ability to process and generate natural language, while offering the possibility to automate the entire process of Idea Generation in such an efficient pipeline.

This is particularly relevant in research areas like optical networks and systems, where innovation and new methodologies are crucial for progress.
Furthermore, the project incorporates prompt engineering techniques such as Chain-of-Thought, Reflection. Through systematic application and evaluation of these techniques, best practices will be developed that are applicable across various research domains.

Collaboration

Weekly meetings with Dr. Safari were conducted to review the progress of the project. These meetings provided an opportunity for me to raise questions, receive valuable feedback, and obtain support when challenges arose.
At the end of my internship, I had the opportunity to present the results and discuss the challenges of the pipeline with the colleagues from the Digital Signal Processing Group.

Chapter 2

Tools and Technologies

SSH was utilized to connect to remote SSH servers equipped with GPUs dedicated for ML tasks.

tmux was employed to initiate and maintain terminal sessions, even during disconnection from SSH connections. This allowed scripts to continue running in the background without interruption. Upon reconnecting, it was possible to reattach to the active session.

Visual Studio Code was used for writing and debugging the pipeline, providing an integrated environment for coding, terminal sessions, SSH connections, and various IDE features. This setup significantly enhanced productivity by consolidating multiple functionalities in a single interface.

Python was the primary programming language for the entire project. Although Python presents challenges such as limited type safety, it offers simplicity and ease of use in writing code.

Large Language Models employed in the project included Llama 3.1 [Dubey et al., 2024, Meta AI, 2024], Mistral [Jiang et al., 2023, 2024, Team, 2023], Gemma 2 [Team et al., 2024], Yi [AI et al., 2024], Phi-3 [Abdin et al., 2024, Microsoft, 2024], Qwen 2.5 [Yang et al., 2024, Qwen Team, 2024], and Deepseek v2 [DeepSeek-AI, 2024], along with closed-source models such as GPT-4o [Achiam et al., 2023, OpenAI, 2024] and Claude Sonnet 3.5 [Anthropic].

Large Language Model APIs were integrated using several packages, including Ollama, LangChain, the Hugging Face Transformers Library [Wolf et al., 2020], the OpenAI API [OpenAI, 2020], and PyTorch.

Chapter 3

Methodology

The Pipeline consists of the following stages:
- Setup
- Idea Generation
- Novelty Check
- Writeup
- Review

The next sections will explain the entire detailed process of a single idea generation. This process will be repeated for the specified number of idea generations.

3.1 Setup

First, the arguments entered when running the script are parsed and saved for run configuration.

3.1.1 Script Arguments

The arguments can manipulate certain aspects of the pipeline. The following arguments can be specified:
`--skip_idea_generation`
This skips the Idea generation and loads up the ideas saved in the template. This is the idea generated from the last generation iteration or run.
`--skip_novelty_check`
If enabled, the generated results do not get checked with Semantic Scholar. More detail on novelty checking can be found in section 3.3. This means that every generated Idea will be processed and used for the writeup.

--model
Multiple LLMs are can be specified:

- Deepseek-Coder v2 - 16B
- Gemma 2 - 9B
- Gemma 2 - 27B
- Llama 3.1 - 8B
- Llama 3.1 - 70B - model by default
- Mistral - 7B
- Mistral-Large 2 - 123B
- Mistral-Nemo - 12 B
- Mixtral 8x7B
- Mixtral 8x22B
- Yi-Coder - 9B
- Phi 3.5 3.8B
- Qwen 2.5 32B
- Qwen 2.5 72B

The command `ollama list` indicates which models are downloaded locally.
Closed-Source API models such as Claude 3.5 Sonnet, GPT-4o, Deepseek-Coder v2, and the API version of Llama 3.1 405B are also available, but they do require an API key in order to work.
GPT-4o uses the `OPENAI_API_KEY` environment variable, Anthropic API `ANTHROPIC_API_KEY`, Deepseek `DEEPSEEK_API_KEY`, and Llama 3.1 `OPENROUTER_API_KEY` (OpenRouter API).
To set the environment environment variable, e. g. for the `OPENAI_API_KEY`, the following command is used:

```
export OPENAI_API_KEY="{YOUR API KEY}"
```

Where {`YOUR API KEY`} is replaced by the actual API key.
Additionally, if newer local models that are supported by Ollama can be downloaded through `ollama pull` {`model_name`}. They are incorporated into the script automatically. Models can be found in the Ollama model library.

--parallel
Specifies the number of parallel processes to run. To execute the ideas sequentially. By default this is set to 0, which means sequential execution.
--gpus
Expects a comma-separated list of GPUs to use, e. g. '0,1,2'. If not specified, all available GPus will be used.
--num-ideas
This specifies the number of ideas to generate. This does **not** necessarily result in the exact number of ideas since some might not pass the novelty check or fail in the process. By default, only one idea is generated.
--topic

A topic to explore and develop new ideas in can be specified. The current default is ”Optical Networks and Systems”.
`--skip_writeup`
If specified, only Ideas will be generated, without a Writeup of further details.
`--skip_review`
If true, this will skip generating the final meta review.
`--improvement`
If true, this will improve the final Idea Generation result based on the supplied review by the Large Language Model.

3.1.2 Script Execution

The following examples illustrate the execution of the script:

Qwen2.5 72B model, 3 idea generations and improvement based on the review

```
python launch_pipeline.py --model "qwen2.5:72b" --num-ideas 3 --improvement
```

Default Llama 3.1 70B model, 20 ideas, only generating ideas

```
python launch_pipeline.py --skip_writeup --num-ideas 20
```

Gemma 2 27B model, one idea, skipping the review

```
python launch_pipeline.py --skip_review --model gemma2:27b
```

Note: The order of the arguments does not matter. After parsing the arguments, the specified used model is set up.

3.2 Idea Generation

Unless `--skip_idea_generation` has been enabled, the script prompts the LLM to generate novel and impactful research ideas with the specified topic and number of ideas. This uses Persona, Reflection and Chain-Of-Thought Prompt Enginneering techniques to further refine the quality of the answers. The Prompts are shared in the Chapter 4.2. The language model is given multiple rounds of reflection to improve the answer. This answer should include a JSON-formatted string with the following fields:

- **Name** - A shortened descriptor of the idea.
- **Title** - A concise title for the idea.
- **Experiment** - An outline of the implementation.
- **Interestingness** - A rating from 1 to 10 based on its potential within the community.
- **Feasibility** - A rating from 1 to 10 based on the practicality of implementing the idea with the given resources.

- **Novelty** - A rating from 1 to 10 reflecting reflecting the uniqueness of the idea in the context of current search.

> → Note that LLM answers with JSON strings included often fail since smaller LLMs do not conform to the JSON format. This is a repeating issue throughout the Pipeline but does not fail the entire paper generation.

If the generated answer includes ”I am done”, the remaining reflection rounds are skipped and the idea is saved in the ideas file `ideas.json` of the directory of the corresponding run.
It is also worth mentioning that newer research has shown that while LLMs can generate novel ideas, they often lack feasibility and with larger amounts of ideas compared, more often produce very similar results, meaning the amount of new ideas ”conver ges”, the more ideas have been generated. [Si et al., 2024].

3.3 Novelty Check

If `--skip_novelty_check` *has been specified in the script parameters, this stage will be skipped and all generated Ideas will be used.*
In the Novelty Check the Semantic Scholar API is used to check for already existing similar ideas.

> → In order for Semantic Scholar to work without rate limits, a semantic scholar API access is needed - which can be applied for on the Semantic Scholar Website.

A query in JSON format is first generated to search for similar ideas in Semantic Scholar. After parsing the given JSON string the script starts querying Semantic Scholar and prompt the results to the LLM. This is repeated up to 20 rounds unless the answer contains ”Decision made: novel” or ”Decision made: not novel” based on the API results.
Once the novelty checking is done, the novelty will be saved in the idea JSON field ”Novelty” (True/False). Only the novel results will be used for the next stage.

3.4 Writeup

Once the ideas are created and proven for uniqueness, the writeup stage will start. Originally, this writeup section of The AI Scientist uses aider [Gauthier, 2024], a high-performing coding assistant scoring more than 20% in the SWE-Bench [Jimenez et al., 2024], a benchmark measuring the Software Engineering Completion based on Real-World GitHub Coding Problems.
Because aider does not work well with the smaller models, another solution has to be found. This led to the change to markdown due to the simplicity but also functionality

of the syntax.
This means there is no longer a template LaTeX file directly (like with aider) - but the LLM is prompted for each section of the draft and answers are saved in a dictionary. Additionally, the Pipeline changes to Markdown due to LLM compatibility issues with LaTeX.
The sections of the proposed paper Idea include Abstract, Introduction, Related Work, Background, Method and Experimental Setup. The code goes through every section with dedicated tips included in the prompt for each section to improve results. If errors occur, the script will retry twice with the corresponding error as context in a the prompt. Next, two rounds of refining every section (Reflection Technique) are done. After the Idea has been fully written, the LLM is prompted to improve any last mistakes and return the final improved Idea writeup.

3.5 Review Stage

The review uses Ensemble Prompting, which has been shown to improve the performance by aggregating multiple model outputs [Zhang et al., 2023]. This means that the LLM will be prompted independently multiple times to review the provided paper with the same query, which has been shown to substancially improve the answer [D'Arcy et al., 2024].
These prompts return reviews, each including a JSON string containing the following fields:

- **Summary**
- **Strengths** (list)
- **Weaknesses** (list)
- **Originality** (low, medium, high, very high)
- **Quality** (low, medium, high, very high)
- **Clarity** (low, medium, high, very high)
- **Significance** (low, medium, high, very high)
- **Questions** (questions to be answered)
- **Limitations** (limitations and potential negative societal impacts)
- **Ethical Concerns** (boolean)
- **Soundness** (poor, fair, good, excellent)
- **Presentation** (poor, fair, good, excellent)
- **Contribution** (poor, fair, good, excellent)
- **Overall** (1 to 10)
- **Confidence** (1 to 5)
- **Decision** (Accept or reject)

The review ensemble is then combined into one meta review with guidelines and persona prompting techniques which can be seen in section 4.2.
Is important to note that the reviews will unfortunately not always conform to the format. But still, this can help improve the original Results.

The meta review is then saved in a `review.txt`. If `--improvement` has been enabled, the Idea Result is further improved with prompting the LLM with the review and the entire Idea Result.

Because the Result is saved in markdown format, the entire file contents can be provided to the improvement stage, making it much more straightforward than the equivalent with LaTeX.

The LLM is then asked to **only** respond with the refined idea. This idea then gets reviewed with the same review process one more time and saved into a `review_improved.txt` file.

The final contents of a generated Idea include:

- Idea Generation in Markdown and PDF format
- Experiment Details
- All Ideas of the Same Run
- System Prompt
- Review
- Review of the Improved Version

Chapter 4

Prompting

4.1 Prompt Engineering Techniques

To improve quality of responses and moreover refine the answers further, different Prompt Engineering Techniques are used. Prompt Engineering can have a positive impact on the agent, no matter the Model size, Parameters or Architecture.

When correctly used, **Personas** can greatly improve the quality of the answer, making it more unique and specific [Lee et al., 2023]. Imagining and then assigning a possible persona most useful to a certain situation to the LLM is highly effective.

Introduced by Wei et al. [2022], **Chain-Of-Thought (CoT)** Prompting enables complex reasoning abilities. The LLM is asked to execute the task step by step in order to solve the problem.

Few-Shot-Examples can help when the LLM fails to achieve complex tasks [Brown et al., 2020]. Multiple examples (shots) of the specific task are given to steer the model to better performance. This serves as a condition for the generated examples.

Reflection converts Feedback from the environment into linguistic feedback [Shinn et al., 2023]. This feedback is then provided to the LLM for the next step. This helps the agent rapidly and effectively learn from prior mistakes. This can be achieved by giving the model multiple rounds of "Self-Reflection" in order for the model to correct its mistakes and refine its answer. This can be seen in the Idea Generation, Writeup and Review stages. By leveraging this technique, the performance of LLMs can improve substantially.

Meta Prompting [Zhang et al., 2024] is a novel approach and similar to Few-Shot-Prompting. But unlike the latter, it focuses on the structure rather than on the content. Furthermore, it enables abstract examples as frameworks, which makes it versatile and applicable in many types of tasks. For example, one might provide the model step-by-step outline as a baseline.

4.2 Prompts

4.2.1 Idea Generation Prompts

Idea Generation Prompt

```
Your task is to generate novel, feasible and impactful research
ideas in the field of {topic}.
Focus on key challenges.  Your goal is to develop experiments or
theoretical advancements that can significantly advance current
methodologies and have broader relevance within the field.
Here are the ideas that you have already generated:

’’’
{prev_ideas_string}
’’’

Your goal is to develop the next significant and creative idea for
topic research that can be feasibly investigated.  Focus on ideas
that address key challenges in {topic}.

Important Considerations:

Ensure that your idea has broader relevance within the field of
{topic}.  Consider how the idea could advance current {topic}
techniques or offer new insights.  Your idea should meaningfully
differ from the existing ones, offering a fresh perspective or
solving a different aspect of the problem.  Furthermore, your idea
should be executable by a team of 10 researchers within a maximum
time frame of 3 months, utilizing resources typically available
in an academic or research environment.  Avoid ideas that require
unproven or not yet available technologies, exorbitant budgets, or
resources that exceed what is usual in {topic}.
Respond in the following format:

THOUGHT:
<THOUGHT>

NEW IDEA JSON:
‘‘‘json
<JSON>
‘‘‘
```

```
In <THOUGHT>, briefly discuss your intuitions and motivations
for the idea.  Provide an overview of your high-level plan,
the necessary design choices, and the ideal outcomes of the
experiments.  Explain how the idea differs from existing ones and
why it is relevant for advancing {topic}.
In <JSON>, provide the new idea in JSON format with the following
fields:
- "Name":  A shortened descriptor of the idea.  Lowercase, no
spaces, underscores allowed.
- "Title":  A concise title for the idea, which will be used for
report writing.
- "Experiment":  An outline of the implementation.  Specify what
needs to be added or modified, how results will be obtained, and
any expected challenges.
- "Interestingness":  A rating from 1 to 10 (lowest to highest)
based on its potential impact within the community.
- "Feasibility":  A rating from 1 to 10 (lowest to highest)
based on the practicality of implementing the idea with the given
resources.
- "Novelty":  A rating from 1 to 10 (lowest to highest) reflecting
the uniqueness of the idea in the context of current research.
Be cautious and realistic in your ratings.  This JSON will be
automatically parsed, so ensure the format is precise.  You will
have {num_reflections} rounds to iterate on the idea, but you do
not need to use them all.
```

The topic is the specified topic when running the script with `--topic "some topic"`. Additionally, the previous ideas are provided to avoid similar ideas generated in the previous results.

After the request is fulfilled, the JSON string is parsed and the idea is saved in the `ideas.json` file where all of the ideas of the run are stored.

Idea Reflection Prompt

```
Round {current_round}/{num_reflections}.
In your thoughts, first carefully consider the quality, novelty,
and especially - the feasibility of the idea you just created.
Ensure that your prposed idea is practically implementable and can
realistically be executed within the given constraints.
Include any other factors that you think are important in
evaluating the idea.  Ensure the idea is clear and concise, and
the JSON is the correct format.
Do not make things overly complicated.
```

```
In the next attempt, try and refine and improve your idea.  Stick
to the spirit of the original idea unless there are glaring issues.
Respond in the same format as before:
THOUGHT: <THOUGHT>
NEW IDEA JSON:
‘‘‘json
<JSON>
‘‘‘

If there is nothing to improve, simply repeat the previous JSON
EXACTLY after the thought and include "I am done" at the end of the
thoughts but before the JSON.
ONLY INCLUDE "I am done" IF YOU ARE MAKING NO MORE CHANGES AND THE
IDEA IS FEASIBLE.
```

This reflection prompt is highly effective, because it uses Chain-Of-Thought and reflection to enhance the previous answer.
By default, the number of reflections is 5.

Novelty Prompt

```
Round {current_round}/{num_rounds}.
You have this idea:

’’’
{idea}
’’’

The results of the last query are (empty on first round):

’’’
{last_query_results}
’’’

Respond in the following format:

THOUGHT:
<THOUGHT>
RESPONSE:
‘‘‘json
<JSON>
‘‘‘
```

```
In <THOUGHT>, first briefly reason over the idea and identify any
query that could help you make your decision.
If you have made your decision, add "Decision made:  novel." or
"Decision made:  not novel." to your thoughts.
In <JSON>, respond in JSON format with ONLY the following field:

- "Query":  An optional search query to search the literature.  You
must make a query if you have not decided this round.

A query will work best if you are able to recall the exact name of
the paper you are looking for, or the authors.
This JSON will be automatically parsed, so ensure the format is
precise.
```

By default, this is repeated for 20 rounds. This not only searches for similar papers with specific queries, but also decides for the novelty.

This enables the fully autonomous process of generating ideas, checking their novelty and continuing with writeup and review.

4.2.2 Writeup Prompts

Writeup Prompt

```
We’ve provided the ’template.md’ file to the project.  We will be
filling it in section by section.
Please fill in the section "{section} of the Writeup.

Here is the proposed Idea:
{idea}

Some tips are provided below:

{per_section_tips[section]}

You have already written this draft of the idea (empty if nothing
written yet):

{written_draft}

Before every paragraph, please include a brief description of what
you plan to write in that paragraph.  Remember that the idea should
be feasible and executable, meaning that overly ambitious projects
```

```
that require breakthroughs in fundamental research or technologies
that are not yet available should be avoided.
Use the following format, writing the content as a replacement for
<TEXT>:
## {section}
<TEXT>

PLEASE DO NOT WRITE ANYTHING BELOW <TEXT> IN YOUR ANSWER!
```

The script will go through this with Abstract, Introduction, Background, Method and Experimental Setup. Tips are provided for each section to guide the LLM how the specific section should look like.

Related Work Prompt

```
For this section, very briefly sketch out the structure of the
section, and clearly indicate what related work you intend to
include.
DO NOT INCLUDE LINKS OR JUST NAMES AS LINKS IN THIS SECTION!
Just mention the general related work, but nothing specific.
You have already written the following draft:

{written_draft}

Use the following format, replacing <TEXT> with the actual text:
## Related Work
<TEXT>

DO NOT WRITE ANYTHING BELOW <TEXT>, as this will be parsed
automatically!
```

Because the Related Work Section is fundamentally different, more specific prompt is used.

Refinement Prompt

```
Great job! Now criticize and refine only the {section} that you
just wrote.
Refine it, making it more detailed and realistic while also
focussing on practical applications and methods that can
realistically be developed and tested within the project
constraints.
Make this complete in this pass, do not leave any placeholders.
```

```
You have already written the following:

{written_section}

Again, follow the following format:

## {section}
<TEXT>

Also, after the text DO NOT WRITE ANYTHING MORE!!!
Pay particular attention to fixing any errors such as:

{error_list}
```

This prompt runs for each section to refine the written paragraph further. To improve readability, the previously written draft is provided.
The error list contains common errors in writing markdown and helps to avoid simple errors, increasing the chance of correct formatting.

Second Refinement Prompt

```
Great Job!  Now that there is a complete draft of the entire
sketch, let's refine each section again.
Criticize and refine the {section} only.
Recall the advice:

{per_section_tips[section]}

Make this complete in this pass, do not leave any placeholders.
Refine your writing in even more detail and clarity while
accounting for the feasibility of the project.
You have already written the following:

{written_section}

Pay attention to how it fits in with the rest of the paper.
Identify any redundancies (e.g.  repeated text), if there are any,
decide where in the paper things should be cut.
Identify where we can save space, and be more concise without
weakening the message of the text.
Again, follow the following format:

## section <TEXT>
```

```
Also, after the text DO NOT WRITE ANYTHING BELOW!!!
Fix any remaining errors as before:

{error_list}
```

Retry Prompt

```
The parsing of the Markdown message failed.  Here is the section
that failed:

{error}

Please try again.  The text should be placed at <TEXT> and after
the text <ou should NOT include anything else.
The format should look exactly like this:
## {section}
<TEXT>
```

This retrying prompt is repeated up to two times if the LLM query parsing goes wrong.

4.2.3 Review Prompts

Review Prompt

```
Below is a description of the questions you will be asked on the
review form for each paper draft and some guidelines on what to
consider when answering these questions.
When writing your review, please keep in mind that after decisions
have been made, reviews and meta-reviews of accepted drafts and
opted-in rejected drafts will be made available to the authors.
Remember it is only a draft, and it does not have to be perfect.

1.  Summary:  Briefly summarize the paper draft and its
contributions.  This is not the place to critique the paper draft;
the authors should generally agree with a well-written summary.
   - Strengths and Weaknesses:  Please provide a thorough
assessment of the strengths and weaknesses of the paper draft,
touching on each of the following dimensions:
   - Originality:  Are the methods or approaches novel?  Is
the work a new combination of well-known techniques that
provides value?  Is it clear how this work differs from previous
contributions?  Is related work adequately cited?
   - Quality:  Is the submission technically sound?  Are claims
```

```
well supported (e.g., by theoretical analysis or experimental
results)?  Are the methods appropriate for the problem being
addressed?  Is this a complete piece of work or work in progress?
Are the authors careful and honest about evaluating both the
strengths and weaknesses of their work?
  - Clarity:  Is the submission clearly written?  Is it well
organized?  (If not, please make constructive suggestions for
improving its clarity.)  Does it adequately inform the reader?
(Note that a superbly written paper draft provides enough
information for an expert reader to reproduce its results.)
  - Significance:  Are the results important?  Are others
(researchers or practitioners) likely to use the ideas or build
on them?  Does the submission address a significant problem in
signal processing or related fields?  Does it advance the state
of the art in a demonstrable way?  Does it provide unique data,
unique conclusions about existing data, or a unique theoretical or
experimental approach?

2.  Questions:  Please list and carefully describe any questions
and suggestions for the authors.
Consider aspects where a response from the authors could change
your opinion, clarify a confusion, or address a limitation.  This
can be very important for a productive rebuttal and discussion
phase.

3.  Limitations:  Have the authors adequately addressed the
limitations and potential technical or societal impact of their
work?  If not, please include constructive suggestions for
improvement.

In general, authors should be rewarded rather than punished for
being upfront about the limitations of their work and any potential
impact.  You are encouraged to think through whether any critical
points are missing and provide these as feedback for the authors.

4.  Ethical concerns:  If there are ethical issues with this paper
draft, please flag the paper draft for an ethics review.
For guidance on when this is appropriate, please review the IEEE
ethics guidelines.

5.  Soundness:  Please assign the paper draft a numerical rating
on the following scale to indicate the soundness of the technical
```

claims, experimental and research methodology, and whether the central claims of the paper draft are adequately supported with evidence.
  4: excellent
  3: good
  2: fair
  1: poor

6. Presentation: Please assign the paper draft a numerical rating on the following scale to indicate the quality of the presentation. This should take into account the writing style and clarity, as well as the contextualization relative to prior work.
  4: excellent
  3: good
  2: fair
  1: poor

7. Contribution: Please assign the paper draft a numerical rating on the following scale to indicate the quality of the overall contribution this paper draft makes to the research area being studied.
Are the questions being asked important? Does the paper draft bring a significant originality of ideas and/or execution? Are the results valuable to share with the broader signal processing community?
  4: excellent
  3: good
  2: fair
  1: poor

8. Overall: Please provide an "overall score" for this submission. Choices:
  10: Award quality: Technically flawless paper draft with groundbreaking impact on one or more areas of signal processing or related fields, with exceptionally strong evaluation, reproducibility, and resources, and no unaddressed ethical considerations.
  9: Very Strong Accept: Technically flawless paper draft with groundbreaking impact on at least one area of signal processing and excellent impact on multiple areas, with flawless evaluation, resources, and reproducibility, and no unaddressed ethical considerations.

```
  8:  Strong Accept:  Technically strong paper draft with novel
ideas, excellent impact on at least one area of signal processing
or high-to-excellent impact on multiple areas, with excellent
evaluation, resources, and reproducibility, and no unaddressed
ethical considerations.
  7:  Accept:  Technically solid paper draft, with high impact
on at least one sub-area of signal processing or moderate-to-high
impact on more than one area, with good-to-excellent evaluation,
resources, reproducibility, and no unaddressed ethical
considerations.
  6:  Weak Accept:  Technically solid, moderate-to-high impact
paper draft, with no major concerns with respect to evaluation,
resources, reproducibility, or ethical considerations.
  5:  Borderline accept:  Technically solid paper draft where
reasons to accept outweigh reasons to reject, e.g., limited
evaluation.  Please use sparingly.
  4:  Borderline reject:  Technically solid paper draft where
reasons to reject, e.g., limited evaluation, outweigh reasons to
accept, e.g., good evaluation.  Please use sparingly.
  3:  Reject:  For instance, a paper draft with technical flaws,
weak evaluation, inadequate reproducibility, and incompletely
addressed ethical considerations.
  2:  Strong Reject:  For instance, a paper draft with major
technical flaws, and/or poor evaluation, limited impact, poor
reproducibility, and mostly unaddressed ethical considerations.
  1:  Very Strong Reject:  For instance, a paper draft with
trivial results or unaddressed ethical considerations.

9.  Confidence:  Please provide a "confidence score" for your
assessment of this submission to indicate how confident you are
in your evaluation.  Choices:
  5:  You are absolutely certain about your assessment.  You are
very familiar with the related work and checked the math/other
details carefully.
  4:  You are confident in your assessment, but not absolutely
certain.  It is unlikely, but not impossible, that you did not
understand some parts of the submission or that you are unfamiliar
with some pieces of related work.
  3:  You are fairly confident in your assessment.  It is possible
that you did not understand some parts of the submission or that
you are unfamiliar with some pieces of related work.  Math/other
details were not carefully checked.
```

```
  2:  You are willing to defend your assessment, but it is quite
likely that you did not understand the central parts of the
submission or that you are unfamiliar with some pieces of related
work.  Math/other details were not carefully checked.
  1:  Your assessment is an educated guess.  The submission is
not in your area or the submission was difficult to understand.
Math/other details were not carefully checked.
Respond in the following format:

THOUGHT:
<THOUGHT>

REVIEW JSON:
‘‘‘json
<JSON>
‘‘‘

In <THOUGHT>, first briefly discuss your intuitions and reasoning
for the evaluation.
Detail your high-level arguments, necessary choices and desired
outcomes of the review.
Do not make generic comments here, but be specific to your current
paper draft.
Treat this as the note-taking phase of your review.

In <JSON>, provide the review in JSON format with the following
fields in the order:
- "Summary":  A summary of the paper draft content and its
contributions.
- "Strengths":  A list of strengths of the paper draft.
- "Weaknesses":  A list of weaknesses of the paper draft.
- "Originality":  A rating from 1 to 4 (low, medium, high, very
high).
- "Quality":  A rating from 1 to 4 (low, medium, high, very high).
- "Clarity":  A rating from 1 to 4 (low, medium, high, very high).
- "Significance":  A rating from 1 to 4 (low, medium, high, very
high).
- "Questions":  A set of clarifying questions to be answered by the
paper draft authors.
- "Limitations":  A set of limitations and potential negative
societal impacts of the work.
- "Ethical Concerns":  A boolean value indicating whether there are
```

```
ethical concerns.
- "Soundness":  A rating from 1 to 4 (poor, fair, good, excellent).
- "Presentation":  A rating from 1 to 4 (poor, fair, good,
excellent).
- "Contribution":  A rating from 1 to 4 (poor, fair, good,
excellent).
- "Overall":  A rating from 1 to 10 (very strong reject to award
quality).
- "Confidence":  A rating from 1 to 5 (low, medium, high, very
high, absolute).
- "Decision":  A decision that has to be one of the following:
Accept, Reject.
For the "Decision" field, don't use Weak Accept, Borderline Accept,
Borderline Reject, or Strong Reject.  Instead, only use Accept or
Reject.
This JSON will be automatically parsed, so ensure the format is
precise.
Here is the draft you are asked to review:

{text}
```

This review prompt combines Chain-Of-Thought, Reflection and Meta Prompting to further increase performance.

Meta Reviewer Prompt

```
You are an Area Chair at a conference regarding the topic {topic}.
You are in charge of meta-reviewing a draft of a paper that was
reviewed by {reviewer_count} reviewers.
Your job is to aggregate the reviews into a single meta-review in
the same format.
Be critical and cautious in your decision, find consensus, and
respect the opinion of all the reviewers.
```

By default, the reviewer count will be 3. This ensembling technique ensures the best quality, by combining all answers into one improved version with this persona prompt. It is important to note that this meta reviewer prompt will only be executed if the number of reviews is more than one, otherwise only one review is used.

Reviewer Reflection Prompt

```
Round {current_round}/{num_reflections}.
In your thoughts, first carefully consider the accuracy and
```

```
soundness of the review you just created.
Include any other factors that you think are important in
evaluating the draft.  Ensure the review is clear and concise,
and the JSON is in the correct format.  Do not make things overly
complicated.
In the next attempt, try and refine and improve your review.  Stick
to the spirit of the original review unless there are glaring
issues.

Respond in the same format as before:
THOUGHT:
<THOUGHT>

REVIEW JSON:
‘‘‘json
<JSON>
‘‘‘

If there is nothing to improve, simply repeat the previous JSON
EXACTLY after the thought and include "I am done" at the end of the
thoughts but before the JSON.
ONLY INCLUDE "I am done" IF YOU ARE MAKING NO MORE CHANGES.
```

Unless ”I am done” is included in the text, the model will reflect for 4 rounds.

Improvement Prompt

```
The following Review has been created for your research paper
draft:

"""
{review}
"""

Improve the following text using the review.
DO NOT ADD ANY OWN TEXT TO YOUR ANSWER!
JUST RESPOND WITH THE MODIFIED MARKDOWN!

{text}
```

This improvement will only happen if `--improvement` has been specified in the run. After the improvement, the Idea Generation is reviewed and saved again as `review_improved.txt`.

Chapter 5

Results

5.1 Performance Benchmark

5.1.1 Setup

This section presents the results of the model evaluations, focusing on key metrics such as the runtime duration, the number of completed generations, and the average performance of each model. Additionally, feasibility and interestingness were evaluated to measure the quality of the generated ideas.

Table 5.1 provides a summary of the performance across different models, with each model generating 10 ideas. The evaluation of the Large Language Models (LLMs) was based on the following key metrics:

- **Duration**: The total time (in minutes) taken by the model to execute a **single** generated idea.
- **Generations**: The number of successful **novel** idea generations out of 10 generation iterations carried out by the model.
- **Performance**: A combined, averaged, overall metric evaluating the overall quality of the generated ideas, rated on a scale from 1 to 10.
- **Feasibility**: The average practicality of the generated ideas in a research context in a team of 10 researchers, rated from 1 to 10.
- **Interestingness**: How novel or creative the generated ideas are (on average), rated from 1 to 10.

Moreover, the following prompt was prompted to state-of-the-art ChatGPT o1-preview to rate the ideas:

Results Review Prompt

```
You are an experienced researcher in the field of Optical Networks and
Systems.  A colleague has sent you an idea for a paper.
Rate the Idea based off certain criteria, not the execution.  Remember it
is only a draft, so it does not need to be perfect.
Review Form
Below is a description of the questions you will be asked on the review
form for each idea draft and some guidelines on what to consider when
answering these questions.
Respond in the following format:
THOUGHT:
<THOUGHT>

REVIEW JSON:
‘‘‘json
<JSON>
‘‘‘

In <THOUGHT>, first briefly discuss your intuitions and reasoning for the
evaluation.
Detail your high-level arguments, necessary choices and desired outcomes
of the review.
Do not make generic comments here, but be specific to the current idea.
Treat this as the note-taking phase of your review.
In <JSON>, provide the idea review in JSON format with the following
fields in the order:
- "Overall":  A rating from 1 to 10 (very strong reject to award
quality).
- "Feasibility":  A rating from 1 (not feasible at all for a small
researcher team of 10 people)to 10 (easily doable within a few days for a
small researcher team of 10 people).
- "Interestingness":  A rating from 1 (not interesting) to 10 (highly
interesting).
- "Confidence":  A rating from 1 to 5 (low, medium, high, very high,
absolute).
- "Decision":  A decision that has to be one of the following:  Accept,
Reject.

For the "Decision" field, don’t use Weak Accept, Borderline Accept,
Borderline Reject, or Strong Reject.  Instead, only use Accept or Reject.
This JSON will be automatically parsed, so ensure the format is precise.
Here is are the next multiple ideas in markdown format you are asked to
review:

’’’
{ideas}
```

5.1.2 Results

Model	Duration	Generations	Performance	Feasibility	Interestingness
Llama 3.1 70B	20	**7**	7,6	7	8,1
Llama 3.1 8B	**3**	**7**	6,4	5,9	7,6
Mistral 7B	6	4	6	5,3	7,3
Mistral-L 123B	242	4	7,3	**7,3**	**8,3**
Qwen 2.5 32B	12	5	7,2	6,6	8
Qwen 2.5 72B	485	1	**8**	7	9

Table 5.1: Average Performance of various models on 10 Idea Generations

Gemma 2 9B, Gemma 2 27B, Mixtral 8x7B, Mistral-Nemo and Phi 3.5 did not produce a complete idea in 10 generations and were therefore left out in the results.
The experiments were run on a single Ubuntu 22.04 Server with a 40GB-VRAM NVIDIA A100, a 32-Core Intel Xeon Processor and 192GiB RAM. With another machine setup, the idea generation duration will obviously change.

5.1.3 Analysis

First, unlike expectations, the amount of completed generations did not depend on model size, but rather on model architecture. The only model architecture delivering "consistent" outputs were the Llama 3.1 Models. This is very likely due to their high instruction-following capabilities [Dubey et al., 2024, Meta AI, 2024, Chiang et al., 2024], and directly translates to the JSON and Markdown formatting of the pipeline.

The Mistral-Large 2 123B produced the most feasible and interesting ideas, although not by a significant margin. The new Qwen 2.5 72B model produced a single, yet highest-performing result. Despite the public benchmark results showing significantly better performance than the direct competitor Llama 3.1 70B, this test did not reflect the results of Qwen 2.5 72B [Yang et al., 2024, Qwen Team, 2024].
Due to the limited GPU capability with the A100, both models are coming at a cost of time, while also only 4, respectively one out of 10 results were novel or had the correct format.

The smallest model by size - being Mistral **7B** - still managed to produce 4 idea generations in just 6 minutes, scoring slightly worse than the **8B** model from Meta. The new Qwen 2.5 32B model also showed great size-to-performance ratio, also running quickly on this machine.
Further experimentation and prompt refinement are recommended to maximize the potential of smaller models, while larger models may benefit from optimizing the balance between computational efficiency and generation quality.

5.2 Example Generations

In this part, three example generations of different quality - each by by another model - are reviewed and discussed. As this internship took part in the Digital Signal Processing Group, the chosen topic will be "Next-Generation Optical Networks and Systems" - although almost any research topic can be chosen.

5.2.1 First Example: Qwen 2.5 72B

Qwen 2.5 72B Example Idea Generation

Machine Learning-Enhanced Digital Signal Processing for Real-Time Noise Mitigation in Next- Generation Optical Networks

Abstract

In this paper, we propose a machine learning-enhanced digital signal processing (DSP) system designed to dynamically mitigate noise and enhance signal integrity in next-generation optical networks. Addressing the challenges posed by chromatic dispersion (CD) and polarization mode dispersion (PMD) is crucial for maintaining high data throughput and reliability in long-distance communication scenarios. These issues are particularly hard to solve due to their dynamic nature and the need for real-time correction. Our solution involves the use of deep neural network (DNN) models implemented on a Field-Programmable Gate Array (FPGA) platform, ensuring low-latency processing. We evaluate the performance of our approach using a testbed consisting of 5-10 nodes equipped with programmable transceivers and a central Software-Defined Networking (SDN) controller. The evaluation focuses on key metrics such as bit error rate (BER), signal-to-noise ratio (SNR), and overall data throughput, demonstrating significant improvements over traditional DSP techniques. Our key contributions include the development of an adaptive machine learning model that can effectively handle varying network conditions, ensuring robust noise mitigation in real-time environments. Additionally, we provide practical methods for training and deploying DNN models within the constraints of an FPGA platform, offering a realistic framework for implementation and testing.

Introduction

In this paper, we address the pervasive issue of noise in optical networks, a critical challenge that impacts data integrity and transmission efficiency. The increasing demand for high-speed, reliable communication, driven by applications such as cloud computing, streaming services, and the Internet of Things (IoT), underscores the relevance of this problem. Noise in optical networks can originate from various sources, including environmental factors, imperfections in fiber optics, and limitations in signal amplification techniques. These issues are particularly significant because they can lead to data corruption, reduced transmission rates, and increased error rates, thereby degrading overall network performance.

Related Work

Recent advancements in the field of machine learning have led to a surge in research focused on developing more efficient and scalable models. This section reviews key contributions that are relevant to our project, particularly those that address practical applications and methods that can be realistically developed and tested within the given constraints.

Background

Optical communication networks form the backbone of modern telecommunications infrastructure, enabling high-speed data transmission over long distances. These networks rely on the propagation of light through optical fibers, which offer significant advantages in terms of bandwidth and low loss. However, they also face several critical challenges, primarily due to noise and dispersion.

Method

In this section, we detail the methodology employed to develop and implement the machine learning- enhanced digital signal processing (DSP) system. The method is structured into four key components: data collection, model development, FPGA implementation, and performance evaluation.

Data Collection

We collected a comprehensive dataset of optical signals under various network conditions, including different levels of chromatic dispersion (CD), polarization mode dispersion (PMD), and other types of noise. This dataset was used to train and validate the DNN models.

Model Development

The deep neural network (DNN) model was designed to handle the dynamic nature of noise in optical networks. We explored various architectures, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to identify the most effective approach for real-time noise mitigation.

FPGA Implementation

The DNN models were implemented on a Field-Programmable Gate Array (FPGA) platform to ensure low-latency processing. We optimized the model's architecture and hyperparameters to fit within the resource constraints of the FPGA, while maintaining high performance.

Performance Evaluation

We evaluated the performance of our system using a testbed consisting of 5-10 nodes equipped with programmable transceivers and a central Software-Defined Networking (SDN) controller. The evaluation focused on key metrics such as bit error rate (BER), signal-to-noise ratio (SNR), and overall data throughput, comparing the results with traditional DSP techniques.

Experimental Setup

To validate our proposed method and ensure its effectiveness, we designed a comprehensive experimental setup that includes detailed descriptions of the hardware configuration, dataset, evaluation metrics, hyperparameters, and implementation details. This section aims to provide a clear and reproducible framework for testing our approach.

Hardware Configuration

The testbed consists of 5-10 nodes equipped with programmable transceivers and connected via optical fibers. A central SDN controller manages the network traffic and monitors the performance metrics.

Dataset

We collected a dataset of optical signals under various conditions, including different levels of CD, PMD, and other types of noise. The dataset was divided into training, validation, and test sets to ensure robust model evaluation.

Evaluation Metrics

The primary evaluation metrics are bit error rate (BER), signal-to-noise ratio (SNR), and overall data throughput. These metrics provide a comprehensive assessment of the system's performance in mitigating noise and maintaining high data transmission quality.

Hyperparameters

We optimized the hyperparameters of the DNN models, including learning rate, batch size, and number of layers, to achieve the best performance on the FPGA platform.

Implementation Details

The DNN models were implemented using a high-level synthesis (HLS) tool to generate the necessary hardware description language (HDL) code for the FPGA. We also developed a software interface to control the FPGA and collect performance data.

Remarks

The proposed highest-rated generation from the benchmark, generated by Qwen 2.5 72B shows the possibilities of the Pipeline.
The proposed idea, "Machine Learning-Enhanced Digital Signal Processing for Real-Time Noise Mitigation in Next-Generation Optical Networks," presents a promising approach to addressing significant challenges in optical communications.

Strengths

- **Relevance**: Tackles critical issues like chromatic and polarization mode dispersion, which are essential for improving data integrity and transmission rates in optical networks.
- **Innovation**: Integrates deep neural networks with DSP on FPGA platforms, offering a novel method for real-time noise mitigation.
- **Practical Implementation**: Utilizes a realistic testbed with programmable transceivers and an SDN controller, ensuring the feasibility of experimental evaluations.
- **Comprehensive Evaluation**: Focuses on key metrics such as BER, SNR, and throughput, providing a thorough assessment of the system's performance.

Areas for Improvement

- **Model Justification**: The choice of CNNs and RNNs lacks a detailed rationale. Clarifying why these architectures are best suited for this application would strengthen the proposal.
- **Dataset Details**: More information on the dataset's size and diversity is needed to ensure the models are trained on comprehensive and representative data.
- **FPGA Constraints**: Addressing how FPGA resource limitations will be managed, such as through model optimization techniques, is necessary for practical implementation. This lack of detail coincides with the limited feasibility of LLM-generated ideas, as shown by Si et al. [2024].
- **Baseline Comparisons**: Clearly defining the traditional DSP methods used for comparison will enhance the evaluation's credibility.
- **Implementation Specifics**: Providing more details on the high-level synthesis tools and software frameworks will improve the reproducibility of the project.

5.2.2 Second Example: Mistral 7B

Qwen 2.5 72B Example Idea Generation

Hybrid Distributed Swarm Intelligence-based Optimal Resource Allocation in Next-Generation Optical Networks

Abstract

This study introduces ’si_opt_hybrid’, a novel, hybrid distributed swarm intelligence algorithm for optimal resource allocation in next-generation optical networks. The algorithm combines Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), addressing the need for adaptable, scalable, and energy-efficient strategies under diverse traffic scenarios. By optimizing resources such as wavelength, power, and frequency, we aim to surpass existing centralized and decentralized optimization methods. The paper could benefit from additional scenarios or more comprehensive testing to validate its effectiveness and robustness in various environments. The findings from this research will catalyze the development of more efficient and adaptable resource allocation strategies in next-generation optical networks.

Introduction

In this introductory section, we will delve into the intricacies of the problem at hand: optimal resource allocation in next-generation optical networks. Specifically, we will discuss the relevance and necessity of devising efficient and adaptable resource allocation strategies that can handle diverse traffic scenarios effectively. Furthermore, we will highlight the challenges associated with this problem and explain why current centralized and decentralized optimization methods fall short when it comes to real-time adaptability, scalability, and energy efficiency. In addition, some sections might require additional explanations for readers unfamiliar with the specific techniques employed.

Related Work

In the realm of Natural Language Processing (NLP), researchers have been pursuing efficient and scalable machine learning models capable of handling large-scale data. To mitigate the high computational requirements in extensive Long Short-Term Memory (LSTM) networks, practical solutions like pruning [3] and knowledge distillation [4] have been proposed.

Background

In this study, we explore optimal resource allocation (ORA) within a graph $G(V, E)$, where V represents nodes and E denotes edges connecting these nodes. Each node $i \in V$ has an associated capacity C_i, representing its processing power, and a set N_i of available resources that can be allocated to different tasks. The edges represent the communication links between nodes, with each edge $e_j \in E$ having a bandwidth capacity e_j and energy consumption c_{e_j}.

Method

The proposed methodology entails creating and implementing a Hybrid-CNN-RNN model, a deep learning approach that combines Convolutional Neural Networks (CNN) with Recurrent Neural Networks (RNN). The resulting architecture, called Hybrid-CNN-RNN, is designed to work optimally on a large dataset of labeled images for improved accuracy and efficiency. Could you expand on the potential impact of the proposed approach in real-world applications? Would it be possible to provide additional explanations for sections that might require clarification?

Experimental Setup

In this section, we detail the experimental setup designed to evaluate the effectiveness of our proposed hybrid swarm intelligence algorithm, ’si_opt_hybrid’, under realistic next-generation optical network scenarios. The evaluation encompasses a wide variety of traffic patterns and conditions, ensuring the versatility of the algorithm for real-world applications.

Remarks

The idea, titled "Hybrid Distributed Swarm Intelligence-based Optimal Resource Allocation in Next-Generation Optical Networks," introduces an interesting approach to resource allocation in optical networks. However, compared to the previous example generated by the bigger Qwen 2.5 72B model, this proposal by the smaller Mistral 7B model exhibits several shortcomings that affect its overall quality and coherence.

Strengths

The project addresses a pertinent issue in optical networks by focusing on optimal resource allocation using a hybrid swarm intelligence algorithm that combines Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). The intent to surpass existing centralized and decentralized methods through adaptability and scalability is commendable. Additionally, the mention of key performance metrics such as wavelength, power, and frequency allocation indicates an awareness of important evaluation criteria in the field.

Areas for Improvement

Despite its strengths, the proposal demonstrates notable deficiencies when compared to the higher-quality example from Qwen 2.5 72B:

- **Inconsistent Content**: The "Related Work" and "Method" sections introduce unrelated topics such as Natural Language Processing (NLP) techniques and Hybrid-CNN-RNN models, which detract from the main focus on swarm intelligence.
- **Lack of Detailed Justification**: Insufficient explanation for selecting ACO and PSO specifically for this application, making it unclear how their combination enhances resource allocation in optical networks.
- **Incomplete Experimental Setup**: The "Experimental Setup" lacks detailed scenarios and baseline comparisons necessary to validate the algorithm's effectiveness comprehensively.

Conclusions

In summary, while the proposed hybrid swarm intelligence algorithm for resource allocation in optical networks is conceptually sound, its execution in the current proposal is hindered by inconsistencies and a lack of focused detail. This is yet another example about a lack of explanation on technologies and experiment details, as already shown in the Paper The AI Scientist [Lu et al., 2024].

5.2.3 Third Example: Llama 3.1 70B

Qwen 2.5 72B Example Idea Generation

Integration of Explainable AI Principles with Network Management Systems for Energy Efficiency Optimization in Optical Amplifiers within Next-Generation Optical Networks and Systems

Abstract

This paper introduces a novel approach to optimizing energy efficiency in optical amplifiers within next-generation optical networks and systems. By integrating Explainable AI (XAI) principles with Network Management Systems (NMS), operators can gain real-time insights into the decision-making process, identifying areas of inefficiency and optimizing energy consumption. Our proposed XAI-NMS system leverages feature attribution and model interpretability to enhance transparency and scalability in NMS solutions, addressing the challenge of optimizing energy efficiency while maintaining model performance and interpretability.

Introduction

The increasing demand for efficient and sustainable energy storage solutions has led to a surge in research focused on developing innovative battery technologies, with lithium-ion batteries emerging as a leading contender due to their high energy density, long cycle life, and relatively low self-discharge rate. However, the widespread adoption of lithium-ion batteries is hindered by several challenges, including high production costs, limited availability of raw materials, and concerns over safety and environmental sustainability.

Related Work

Research on human-computer interaction has led to various approaches for improving user experience, with a focus on practical applications and methods that can be realistically developed and tested within project constraints. Incorporating natural language processing (NLP) techniques has been shown to significantly enhance user engagement in controlled studies [1]. For instance, NLP-powered chatbots have demonstrated improved user interaction compared to traditional graphical user interfaces (GUIs) in tasks requiring complex decision-making [3].

Background

Explainable AI (XAI) has emerged as a crucial aspect of artificial intelligence research, aiming to provide insights into the decision-making processes of machine learning models. XAI principles can be applied to various domains, including network management, where complex decisions are made based on large amounts of data. The application of XAI in network management is particularly relevant, as it enables the development of more transparent and accountable Network Management Systems (NMS). NMS play a vital role in ensuring the reliability, efficiency, and security of modern communication networks.

Method

The proposed system for optimizing energy efficiency in optical amplifiers consists of several stages, designed to ensure robustness, accuracy, and practical feasibility. Our methodology adheres to the formalism introduced in the Background section. We applied our evaluation methodology through extensive simulations and experiments using realistic network topologies and traffic patterns.

Experimental Setup

To evaluate the performance of our proposed system, we used a combination of simulated and real-world data. The simulation environment was built using Python 3.8 and the PyTorch library, with a focus on scalability and flexibility. Specifically, we utilized the PyTorch Lightning framework to manage the training loop and experiment tracking.

Limitations and Future Work

Our proposed approach has several limitations, including potential overfitting in simulations and limited robustness. To address these limitations, future work should focus on developing more robust evaluation methodologies and exploring alternative approaches to optimize energy efficiency in optical amplifiers.

Conclusion

In this paper, we presented a novel approach for optimizing energy efficiency in optical amplifiers within next-generation optical networks and systems. By integrating Explainable AI (XAI) principles with Network Management Systems (NMS), our proposed XAI-NMS system provides real-time insights into the decision-making process, enabling operators to identify areas of inefficiency and optimize energy consumption. Through extensive simulations and experiments, we demonstrated significant energy savings (up to 25%) without compromising model performance or interpretability, showcasing the feasibility of our approach for real-world deployment.

Remarks

The proposed idea, titled "Integration of Explainable AI Principles with Network Management Systems for Energy Efficiency Optimization in Optical Amplifiers within Next-Generation Optical Networks and Systems", presents a highly relevant and innovative approach to enhancing energy efficiency in optical networks through the integration of Explainable AI (XAI) with Network Management Systems (NMS).

Strengths

- **Innovative Integration**: Combining XAI principles with NMS for energy optimization in optical amplifiers is a novel approach that addresses the increasing need for transparency and efficiency in network management.
- **Practical Applicability**: The focus on real-time insights and energy consumption optimization directly tackles critical challenges in next-generation optical networks, making the proposal highly relevant for current industry needs.
- **Comprehensive Evaluation**: The inclusion of simulations and experiments using realistic network topologies and traffic patterns demonstrates a thorough methodology for validating the system's effectiveness.
- **Clear Contribution**: The proposal outlines significant energy savings (up to 25%) without compromising model performance or interpretability, highlighting the potential impact of the research.

Areas for Improvement

- **Coherence in Sections**: The "Introduction" and "Related Work" sections introduce topics such as lithium-ion batteries and NLP techniques, which are not directly related to the core focus on XAI and NMS in optical networks. Ensuring that all sections remain aligned with the main objective would enhance the overall coherence of the proposal.

- **Methodological Clarity**: The ”Method” section references a Hybrid-CNN-RNN model for image datasets, which seems unrelated to energy efficiency optimization in optical networks. Clarifying the methodology to focus on XAI and its integration with NMS would strengthen the proposal’s focus.
- **Detailed Justification**: Providing a more detailed explanation of how XAI principles specifically contribute to energy efficiency optimization would offer clearer insights into the proposed system’s advantages.
- **Enhanced Experimental Details**: Expanding the ”Experimental Setup” section to include more specific scenarios, baseline comparisons, and detailed descriptions of the XAI techniques employed would improve the robustness of the evaluation.

Conclusions

Overall, the proposed integration of Explainable AI with Network Management Systems for optimizing energy efficiency in optical amplifiers demonstrates significant potential in advancing sustainable practices within optical networks. While this idea may not be as innovative as the first, it still offers inspiration and possible new methodology for future papers.

5.3 Summary of Findings

The benchmarking and qualitative analysis of the Large Language Models (LLMs) revealed significant insights into their performance and suitability for generating high-quality research ideas in the field of Optical Networks and Systems. Among the evaluated models, Llama 3.1 70B demonstrated capabilities in producing relevant and innovative ideas. Despite other models producing results with higher quality, Llama 3.1 70B proved to have the best instruction-following capabilities in very short time durations for idea generations.

Conversely, smaller models such as Mistral 7B showed limitations in feasibility and relevance, often deviating from the core focus of Optical Networks and Systems. Despite their faster runtime and lower resource consumption, these models struggled to maintain the same level of quality in idea generation, highlighting a trade-off between efficiency and output quality.

The Qwen 2.5 32B model presented a balanced performance, offering a reasonable compromise between runtime efficiency and idea quality, making it another viable option for scenarios with constrained computational resources.

The analysis of example generations further underscored the importance of model architecture and prompt engineering in achieving high-quality outputs. Higher-performing models produced well-structured and focused research ideas with clear contributions and practical applicability, whereas lower-performing models exhibited inconsistencies and introduced unrelated concepts, detracting from the main objectives. This disparity emphasizes the need for careful model selection and prompt design to align the generated ideas with specific research goals.

Another recurring issue next to the misformatted ideas and outputs of LLMs is the "convergence" of ideas; The more ideas have been generated, the more similar ideas happened, as thouroughly investigated by Si et al. [2024]. Ideas also lacked feasibility - even with prompt engineering tackling these issues.

To further improve the quality of generations, other models such as the leading open-source model, Llama 3.1 405B can be used. Unfortunately, in this case, there were not enough computational resources available to benchmark this model.

Overall, the findings suggest that while larger and more advanced models offer substantial benefits in generating high-quality research ideas, smaller models can still be effectively utilized with optimized prompts and targeted use cases. Future efforts should focus on enhancing the coherence and relevance of smaller models through refined prompt engineering regarding the parsing of ideas.

Exploring hybrid approaches that leverage the strengths of multiple models to achieve both efficiency and excellence in research idea generation, e. g. using a separate model for the review and improvement stage to avoid bias in the rating, could be another research topic. Last, but not least, the feasibility of ideas needs to be tackled substantially for the generations to be more applicable and usable.

Chapter 6

Conclusions

During the internship at the Fraunhofer Heinrich Hertz Institute, a Paper Idea Generation Pipeline has been successfully developed, which automates the process of generating novel research ideas using Large Language Models (LLMs). The pipeline integrates multiple open-source and closed-source LLMs and includes features like novelty checking, idea refinement, as well as automatic paper writeup and review. This project provided the opportunity to apply theoretical knowledge in the development of a practical tool, with the potential to streamline the brainstorming process within the institute.

One of the key achievements of the project was the successful integration and benchmarking of various open-source LLMs, such as Llama 3.1, Mistral, and Qwen 2.5. Additionally, closed-source models, including GPT-4o and Sonnet 3.5, were also implemented. This integration provided significant flexibility in model selection and enabled comprehensive analysis of the performance of different models.

An important insight gained from this project was the significant impact of prompt engineering techniques on the quality of the generated ideas. Techniques such as Chain-of-Thought and Reflection enhanced both the coherence and relevance of the outputs. Furthermore, incorporating multiple rounds of reflection allowed the models to refine their responses, resulting in ideas that were more unique and innovative.

However, several challenges were encountered. Smaller LLMs frequently failed to generate outputs in the correct JSON or LaTeX format, as demonstrated by Xia et al. [2024]. This limitation posed a problem, as correct formatting was essential for the pipeline's processing of JSON and the compiling of LaTeX. To mitigate this, robust error-handling mechanisms were implemented and the format was simplified to markdown.

Overall, this project provided valuable insights into the capabilities and limitations of LLMs in automating the idea generation process. It demonstrated the potential for such technologies to accelerate innovation within the institute.

In conclusion, this internship has been instrumental in my academic and professional development. It has equipped me with practical skills and insights that I will continue to apply throughout my career. I am deeply grateful for the opportunity to contribute to the institute's research efforts.

Chapter 7

Outlook

Looking ahead, several avenues for future work are apparent.
Firstly, enhancing the error-handling capabilities of the pipeline could significantly improve its reliability. Developing more robust JSON parsing mechanisms, potentially incorporating additional models or APIs, may increase the likelihood of correctly formatted JSON responses and reduce processing interruptions.

Furthermore, as research in the field progresses, refining the prompt engineering techniques could further enhance the quality of the responses across all models integrated into the pipeline. Continuous experimentation with methods such as Chain-of-Thought, Reflection, and Meta-Prompting may yield more consistent and accurate outputs, thereby improving the overall performance of the system.

Additionally, exploring the application of the pipeline in other research domains beyond Digital Signal Processing and Optical Networks and Systems could provide valuable insights into its strengths and limitations. Such exploration may reveal domain-specific challenges and facilitate the adaptation of the pipeline to a wider range of scientific fields.

As Large Language Models continue to advance, particularly smaller models, their improved performance could indirectly enhance the quality of generated ideas. Recent developments, such as Qwen 2.5, indicate that the quality gap between larger and smaller models - as well as between closed-source and open-source models - is diminishing. This trend is particularly advantageous for locally run projects like this one, where the resources for large-scale, closed-source models may be limited.

In summary, ongoing advancements in LLM technology, combined with targeted improvements to the pipeline's error handling and prompt engineering strategies, hold significant potential for augmenting its effectiveness. By addressing these areas, the pipeline can become a more robust and versatile tool, capable of contributing to innovation across various research domains.

Bibliography

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue

Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai, 2024. URL https://arxiv.org/abs/2403.04652.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. URL https://api.semanticscholar.org/CorpusID:268232499.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024. URL https://arxiv.org/abs/2403.04132. Leaderboard: https://lmarena.ai.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. Marg: Multi-agent review generation for scientific papers, 2024. URL https://arxiv.org/abs/2401.04259.

DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Paul Gauthier. Aider is ai pair programming in your terminal, 2024. URL https://github.com/paul-gautier/aider.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

Joosung Lee, Minsik Oh, and Donghun Lee. P5: Plug-and-play persona prompting for personalized response selection, 2023. URL https://arxiv.org/abs/2310.06390.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292.

Meta AI. Introducing llama 3.1: Our most capable models to date, July 2024. URL https://ai.meta.com/blog/meta-llama-3-1.

Microsoft. Phi-3 cookbook: Hands-on examples with microsoft's phi-3 models, June 2024. URL https://github.com/microsoft/Phi-3CookBook.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024. URL https://arxiv.org/abs/2402.06196.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2024. URL https://arxiv.org/abs/2307.06435.

OpenAI. Openai api, 2020. URL https://openai.com/index/openai-api.

OpenAI. Hello gpt-4o, 2024. URL https://openai.com/index/hello-gpt-4o.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers, 2024. URL https://arxiv.org/abs/2409.04109.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Mistral AI Team. Mixtral of experts. a high quality sparse mixture-of-experts., December 2023. URL https://mistral.ai/fr/news/mixtral-of-experts.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL https://arxiv.org/abs/1910.03771.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. FOFO: A benchmark to evaluate LLMs' format-following capability. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 680–699, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.40.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Chenrui Zhang, Lin Liu, Jinpeng Wang, Chuyuan Wang, Xiao Sun, Hongyu Wang, and Mingchen Cai. Prefer: Prompt ensemble learning via feedback-reflect-refine, 2023. URL https://arxiv.org/abs/2308.12033.

Yifan Zhang, Yang Yuan, and Andrew Chi-Chih Yao. Meta prompting for ai systems, 2024. URL https://arxiv.org/abs/2311.11482.