University of Freiburg

Department of Computer Science

Master's Thesis

# Finetuning Open-Source LLMs for Scientific Idea Generation

**Shanmugapriya Kanagasabapathi**

University of Freiburg
Department of Computer Science

**Master's Thesis in Computer Science**

# Finetuning Open-Source LLMs for Scientific Idea Generation

Author: Shanmugapriya Kanagasabapathi
Advisor: Hanne Raum
Examiner 1: Prof. Dr. Joschka Bödecker
Examiner 2: Prof. Dr. Mario Krenn (University of Tübingen)

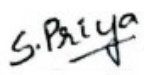
Submission Date: 31.03.2026

# Declaration of Originality

I hereby declare, that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work.

I also hereby declare that my thesis has not been prepared for another examination or assignment, either in its entirety or excerpts thereof.

Tübingen, 31.03.2026



(Shanmugapriya Kanagasabapathi)

# Abstract

Large language models (LLMs) and agentic systems are increasingly used in numerous fields of science. Recent artificial intelligence (AI) systems for automated creative idea generation and implementation can lead to scientifically interesting insights, but rely on closed-source LLMs. Being dependent on these static models, such AI systems often suffer from a low ratio between generated and useful ideas and cannot be easily adapted to specialized tasks. In this thesis, we investigate whether finetuning open-source LLMs can improve scientific idea generation in the context of AI-Mandel, an agentic system for generating and implementing experimental ideas in quantum optics. We compare leading open-source LLMs on idea generation tasks and demonstrate that finetuning such models can lead to significant improvement in idea quality when evaluated by other more powerful LLMs. At the same time, our results show that improved idea quality alone does not directly translate into higher agentic system efficiency: to achieve that, diversity and related constraints must also be explicitly enforced. Our results highlight the potential and limitations of finetuned open-source LLMs for scientific idea generation.

# Contents

# Acknowledgements

I would like to thank everyone who supported me throughout the writing of this thesis.

- I would first like to express my sincere gratitude to Prof. Dr. Mario Krenn for giving me the opportunity to write my thesis in the Machine Learning in Science II group of the University of Tübingen and for his continuous support, guidance, and encouragement throughout the development of this master's thesis.
- My sincere thanks also go to Sören Arlt for providing the foundation for this project, advice and for his helpful feedback.
- I would further like to thank Hanne Raum for her time, support and feedback during the course of this work.
- I am also deeply grateful to Prof. Dr. Joschka Bödecker for supervising this thesis and for allowing me to work independently.
- In addition, I would like to acknowledge the ML Cloud Cluster Administration at the University of Tübingen for their support and for providing the computational resources necessary for this research.

  Finally, I would like to thank my fiancée, Jonathan Klimesch for his constructive comments, constant encouragement, patience, and support throughout this journey.

# 1 Introduction

Large language models (LLMs) are becoming increasingly important in scientific workflows. Across different fields like chemistry [1, 2, 3, 4, 5], physics [6, 7, 8, 9, 10], mathematics [11], or computer science [12], LLM agents can support human researchers, speed up processes or help in the design of new experiments. So-called AI scientists or artificial scientists can automate large fractions of the scientific process and present human researchers with final results with only minimal human intervention [13, 14, 15, 16, 17].

While in most of these cases, the scientific objective or research idea still comes from human researchers, there exist first prototypes of AI systems that can also generate their own ideas [13] or predict the impact of research topics [18]. One example of such a system is AI-Mandel, an agentic system that not only generates ideas in quantum optics but also validates these ideas through PyTheus [19], a tool for highly efficient discovery of quantum optics experiments. While AI-Mandel can already produce publishable contributions to quantum physics [20, 21], it relies on general-purpose LLMs that are not specifically trained for scientific idea generation tasks. As a result, the ratio of generated ideas to practically useful, high-quality proposals remains low. Furthermore, most agentic systems in the context of scientific workflows rely on closed-source models such as GPT, Gemini, or Claude. These models can change in behavior and quality and cannot be finely controlled or improved for new tools like PyTheus that are added to the agentic system.

To improve the yield and quality of AI-generated scientific ideas while reducing dependence on closed-source models, this thesis presents a first study of finetuning open-source LLMs for scientific discovery in quantum optics. Our research aims to answer the main research question: How does finetuning an open-source LLM for scientific idea generation compare against its out-of-the-box baseline with in-context learning?

We first conduct a comprehensive comparison of leading open-source LLMs to determine their suitability for scientific idea generation in quantum optics. After choosing a specific LLM, we then move on to evaluate a set of prompting techniques in order to create a meaningful baseline model for the finetuning process. In both cases, we assess quality against a baseline with in-context learning using LLM judges. The diversity of outputs from finetuned models is analyzed using dimensionality reduction techniques applied to

sentence embeddings. We then integrate our finetuned models into the agentic system of AI-Mandel and evaluate their interplay with other agents.

In chapter 2, we give an overview of related work on agentic systems for scientific discovery, with a focus on idea generation and finetuning LLMs. In chapter 3, we discuss the existing AI-Mandel system and provide a short introduction to LLMs, finetuning techniques and evaluation metrics. In chapter 4, we describe our methodology behind the comparison of open-source LLMs and prompt selection before moving on to explaining our approach to finetuning and the diversity evaluation of the outputs of the finetuned model. We report the results of our evaluation metrics in chapter 5 and discuss them in detail in chapter 6. We conclude in chapter 7 by outlining future development plans.

# 2 Related Work

LLM-driven agentic systems have recently emerged as a promising path towards artificial research assistants. These systems are networks of LLM agents with different prompts that can communicate with each other and use different tools. Such systems can autonomously generate ideas, check their novelty by performing literature search, implement experiments to test them, and finally compile results into papers [14, 16, 17].

Specifically for idea generation tasks, these agentic systems often follow a similar principle as evolutionary search. From a pool of existing ideas, an LLM is prompted to recombine these ideas into something new. Generated ideas then get rated for novelty, interest and feasibility, possibly with the support of tools such as online literature search [15, 22, 23].

An alternative to LLM-based idea generation is recommendation systems based on knowledge graphs extracted from scientific papers. Here, neural networks can be trained on keyword pairs extracted from paper titles to predict which combinations will become important in the future or are especially impactful. By formulating keyword combinations as full texts, their potential can be evaluated by human experts [24, 25, 26, 27, 28, 18, 29, 30, 31, 22].

While previous works exclusively rely on closed-source, general-purpose LLMs, this thesis focuses on finetuning open-source LLMs for scientific idea generation. Recent papers introduce several methods for incorporating feedback with different levels of detail into the finetuning process.

Supervised finetuning (SFT) via next-token prediction on high-quality texts is already a form of fine-grained feedback. Rather than relying on coarse scalar rewards, SFT provides dense, token-level supervision, implicitly encoding domain-specific preferences [32, 33, 34]. Methods such as direct preference optimization (DPO) [35] and reinforcement learning from human feedback [36] further improve models based on binary feedback signals.

A number of methods try to improve upon pure SFT without relying on binary feedback. Wang et al. [37] introduce Text2Grad, a general-language approach that uses sentence-wise binary feedback to generate token-level rewards and update the parts of the model responsible for errors. In the paper Towards Aligning Language Models with Textual

Feedback, Lloret et al. [38] introduce ALignment with Textual feedback (ALT), where a model is finetuned to suppress toxic responses in general conversations. Here, the finetuning is done by next token prediction on triplets of prepended text feedback, prompt and model output. Similar to ALT, the research of Wang et al. [39] introduces LETI (Learning to Generate from Textual Interactions) for code generation tasks, where the finetuning is now performed on quadruplets of binary ranking, text feedback, prompt and model output. As an extension to the popular Reinforcement Learning from Human Feedback (RLHF) algorithm [36], Wu et al. [40] present fine-grained RLHF with rewards on different scales using categorical feedback at output-level, sentence-level and sub-sentence level for detoxification and long question and answer tasks. Liu et al. [41] use reward modeling with ordinal feedback as an alternative to binary feedback in a general-purpose, instruction-following objective. It uses preference feedback with three (A better, tie, B better) or five (A much better, A better, tie, etc.) levels to train a reward model that can then be used for finetuning. Finally, Koroleva et al. [42] use distributional scores, where ratings from multiple raters are not averaged into a single score but given as a distributional feedback signal to the model during training. In their general-purpose text test cases, the model can accurately capture polarizing texts where the average might be misleading.

Recently, LLMs have also become popular as scalable, automated judges that can rate other LLMs and often show surprisingly strong alignment with human judges [43]. While the potential of LLM judges is already being exploited in many fields [44], there is equally strong evidence that these LLM judges are often prone to different kinds of biases that must be considered when using these techniques [45, 46].

In this thesis, we mainly concentrate on the combination of SFT and DPO, and apply them to a task that has not been tried before: finetuning LLMs to generate ideas in quantum optics. Our contributions lie in testing and finetuning open-source models and prompting techniques in a field that requires the acquisition of strong domain knowledge in a relatively complex scientific context that goes far beyond tasks like simple toxicity estimation [38] from other works. To evaluate our finetuned models at scale, we make use of LLMs as judges, carefully addressing the biases that have been brought forward in other recent works.

# 3 Background

In section 3.1 of this background chapter, we will provide a short overview of the principles behind large language models (LLMs), introduce common open-source models, and explain the main prompting techniques that are relevant for this thesis. To provide a background for our finetuning methodology, we discuss common finetuning algorithms and techniques such as LoRA and quantization in section 3.2. In section 3.3, we then briefly discuss how agentic systems work and describe AI-Mandel, the system that we aim to improve. Finally, in section 3.4, we introduce metrics that are used later to evaluate models and prompting techniques.

## 3.1 Large Language Models

In recent years, large language models that rely on billions of parameters and are trained on very large amounts of text have been at the center of discussions about artificial intelligence. Since we make heavy use of this type of model, the following sections summarize how they work, what models are available at the moment, and how to use them.

### 3.1.1 Transformer Architecture

Large language models mainly rely on the transformer architecture [47] whose key mechanism is self-attention. It enables the model to learn the relative importance of parts of an input sequence. First, the input sequence is divided into tokens $x_i$, which are subsequently transformed into embeddings $e_{x_i}$, vectors that numerically capture the meaning of each token.

Given an input embedding sequence $E = [e_{x_1}, \ldots, e_{x_n}] \in \mathbb{R}^{n \times d}$ with $n$ token embeddings $e_{x_i}$ of dimension $d$, the sequence is sometimes additionally augmented with a positional embedding $PE$ that accounts for the order of tokens: $H = E + \mathrm{PE}$. The attention mechanism then calculates three arrays – the query $Q$, the key $K$ and the value $V$ – via the trainable weight matrices $W_Q$, $W_K$ and $W_V \in \mathbb{R}^{d \times d_k}$:

$$Q = HW_Q, \quad K = HW_K, \quad V = HW_V. \tag{3.1}$$

Query $Q$ and key $K$ are used to compute weights for the value $V$ through so-called scaled dot-product attention:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{3.2}$$

where $d_k$ is an architecture parameter. Multi-head attention using $h$ heads enables the model to simultaneously focus on many different patterns and positions within the input sequence. It effectively performs the attention mechanism from equation 3.2 $h$ times, concatenates the outputs, and processes it with another trainable weight matrix $W_O \in \mathbb{R}^{(hd_k)\times d}$:

$$\begin{aligned} \text{head}_i &= \text{Attn}\Big(HW_Q^{(i)},\, HW_K^{(i)},\, HW_V^{(i)}\Big) \\ \text{MHA}(H) &= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\, W_O. \end{aligned} \tag{3.3}$$

One transformer block then builds on this mechanism by computing:

$$\begin{aligned} H' &= \text{LN}(H + \text{MHA}(H)) \\ H^{\text{out}} &= \text{LN}(H' + \text{FFN}(H')), \end{aligned} \tag{3.4}$$

where $LN$ is a layer normalization, $FFN$ is a feed-forward neural network and both multi-head attention and feed-forward network have residual connections around them.

Transformer architectures then usually stack $L$ such blocks before calculating an output via:

$$\begin{aligned} \ell_i &= H_i^{(L)} W_{\text{vocab}} + b \\ P(x_{i+1} \mid x_{\le i}) &= \text{softmax}(\ell_i), \end{aligned} \tag{3.5}$$

where $W_{\text{vocab}} \in \mathbb{R}^{d\times|\mathcal{V}|}$ is the output matrix that maps to the vocabulary $\mathcal{V}$ and returns a probability distribution over this vocabulary for the next token $x_{i+1}$.

### 3.1.2 Training Procedure

The training of a transformer-based LLM is usually divided into two stages [48, 49, 50]. The first stage is *pre-training*, which is next-token prediction on a large preprocessed text corpus. Given an input token sequence $X = [x_1, \ldots, x_n]$, the model is trained to minimize the negative log-likelihood (cross-entropy loss) of each token conditioned on all previous tokens:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{n} \log p_\theta(x_t \mid x_{<t})\,. \tag{3.6}$$

In the second, *post-training* stage, LLMs are often trained on smaller, high-quality datasets to improve answer quality, filter unwanted behavior, and equip them with specific behaviors or knowledge such as instruction-following, chat behavior or coding capabilities. The specific technique depends on the task at hand and varies between supervised finetuning [32, 33] or different reinforcement learning strategies [51, 36].

### 3.1.3 Open-Source Models

While the most powerful LLMs are only accessible through company APIs, there is an emerging trend to publish the trained weights of smaller models. This trend started with the publication of GPT-2 in 2019, with the largest model reaching 1.5 billion parameters [52]. In recent years, several companies have published open-source LLMs reaching up to 685 billion parameters [53]. Notable models that are used or tested in this thesis are Llama-3.1 with 8 billion [48], Gemma-3 with 27 billion [49], and GPT-OSS-20B with 20 billion parameters [50].

GPT-OSS-20B uses a transformer-based, decoder-only architecture inspired by GPT-2 and GPT-3. It has 24 transformer blocks with 20.9 billion parameters and a context length of 131,072 tokens. The model is an autoregressive Mixture-of-Experts (MoE) transformer, meaning that it has feed-forward blocks that are divided into 32 parts, the so-called experts. A linear map in front of these divisions acts as a router and computes a token-wise score for each expert. The token is then processed by the top-4 experts, weighted by their respective score [50].

During post-training, GPT-OSS-20B was trained with reinforcement learning to solve problems using Chain-of-Thought reasoning (CoT), and was designed to be used in agentic systems. In addition, it was trained for tool use, including browsing tools, Python programming, and other developer functions.

### 3.1.4 Prompting Techniques

LLMs generate outputs conditioned on prompts. These inputs can be given in the form of text, but also in other modalities such as images, audio or combinations thereof. A deeper understanding of how prompt phrasing connects to output quality is part of active research [54, 55]. In this thesis, we make use of different prompting techniques and compare different prompts for the specific task of idea generation in quantum optics. Here, we give a short overview of the used prompting strategies.

*In-Context Learning* describes the prompting technique where the model learns from instructions or examples given in the prompt itself. In zero-shot prompting the model is given only instructions, but no examples [56]. On the other hand, few-shot prompting includes examples directly in the prompt, thereby conditioning the model output on this extra information [57].

In *chain-of-thought prompting*, the input usually contains a statement like "Let's think through this step-by-step", conditioning the model to output an answer together with detailed reasoning [58, 59, 60].

In *self-criticism prompting*, the model is conditioned to critically assess its own answer and improve it further. By generating its own questions that are supposed to test the answer or by giving a confidence score, the model can sometimes self-correct and improve its output [61, 62].

*Role prompting* assigns the model a specific persona, for example, a quantum researcher who wants to come up with new research ideas. This can improve writing and style, and sometimes also increase accuracy [63, 64].

## 3.2 Finetuning Large Language Models

While the training procedure described in section 3.1.2 yields general-purpose models that are well-suited for generic tasks, training LLMs on smaller datasets collected for specific, more narrow tasks can often significantly improve the quality of these models. Here we will give an introduction to the main finetuning techniques and tools used later in this thesis, specifically Supervised Finetuning in section 3.2.1, Direct Preference Optimization in section 3.2.2, as well as Low-Rank Adapters and Quantization in sections 3.2.3 and 3.2.4.

### 3.2.1 Supervised Finetuning

Supervised Finetuning (SFT) uses the same next-token prediction objective as pretraining (equation 3.6), but on a task-specific dataset [32, 33]. In our case, the specific task is idea generation for quantum optics experiments.

Instead of finetuning the model based on data from external sources such as humans or other models, SFT can also be done on the model's own outputs, potentially revised based on external feedback [34]. In this case, SFT does not clone the behavior of one model into another, but instead steers the behavior of the finetuned model into a subset of its capabilities based on that external feedback.

### 3.2.2 Direct Preference Optimization

One of the most successful finetuning methods for instilling desired behaviors into LLMs is reinforcement learning from human feedback (RLHF) [36]. Here, an explicit reward model is fit to a dataset of human preferences and the LLM is then trained with reinforcement learning to generate outputs that get a high reward without drifting too far from its initial behavior. This finetuning pipeline is more complex than SFT as it requires training multiple models within a reinforcement learning setting.

Recently, Direct Preference Optimization (DPO) was introduced as a simplified alternative to RLHF [35]. DPO optimizes for the same objective as RLHF, but does so in a simpler way, without an additional reward model and without the need for reinforcement learning. Similar to equation 3.6, DPO is based on $\pi_\theta(y \mid x)$, the probability of the model $\pi_\theta$ to assign the output $y$ to the prompt $x$. This can be expressed as an autoregressive factorization over the individual tokens $y_t$ conditioned on the prompt and previous output tokens $y_{<t}$ with $n$ as the output length:

$$\pi_\theta(y \mid x) = \prod_{t=1}^{n} p_\theta(y_t \mid x, y_{<t})\,. \tag{3.7}$$

DPO now uses this sequence probability in the formulation of a preference margin that considers two outputs for one prompt: $y^+$ as a desired response and $y^-$ as an undesired response for prompt $x$. The margin is introduced between the finetuned model $\pi_\theta$ and a frozen reference model $\pi_{\text{ref}}$ that is usually the initial model of the DPO process.

$$\begin{aligned}\Delta_\theta(x, y^+, y^-) = &\Big(\log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x)\Big) \\ &- \Big(\log \pi_{\text{ref}}(y^+ \mid x) - \log \pi_{\text{ref}}(y^- \mid x)\Big).\end{aligned} \tag{3.8}$$

By minimizing

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}\left[\log \sigma\big(\beta\, \Delta_\theta(x, y^+, y^-)\big)\right] \tag{3.9}$$

with $\mathcal{D}$ as the preference dataset or desired and undesired response pairs, $\beta$ as a scaling hyperparameter and $\sigma$ as the sigmoid function. With this objective function, DPO pushes the finetuned model $\pi_\theta$ to prefer $y^+$ over $y^-$ while not drifting too much from its reference $\pi_{\text{ref}}$. This prevents incoherent sentences while improving the expected reward.

### 3.2.3 Low-Rank Adapters

Full finetuning of LLMs with billions of parameters is computationally expensive. In this thesis, we use low-rank adapters (LoRA) [65], a popular technique to make finetuning less resource demanding. Instead of finetuning a full weight matrix $W_0 \in \mathbb{R}^{d\times k}$, only a low-rank update to selected weight matrices is finetuned:

$$h = W_0 x + \Delta W\, x = W_0 x + BAx. \tag{3.10}$$

The matrices $A \in \mathbb{R}^{r\times k}$ and $B \in \mathbb{R}^{d\times r}$ share dimension $k$ or $d$ with $W_0$ but have a small rank $r$. The same procedure can be applied to multiple model layers. The main assumption of LoRA is that it is sufficient to only adapt a low-dimensional subspace of the model parameters to finetune the model. LoRA can reach similar performance compared to full finetuning while using significantly fewer parameters, thereby saving compute and memory costs.

### 3.2.4 Quantization

Another popular technique that we use in this thesis to reduce the memory burden of finetuning LLMs are quantization techniques such as MXFP4 [66]. Here, model weights are grouped into small blocks of for example, 32 numbers. Using an 8-bit scaling factor $S_{\text{block}_i}$ for each block $i$, each number at index $j$ within block $i$ can then be reduced to a 4-bit representation $P_{ij}$. Using the scaling factor, the original weights can be closely approximated by $X_{ij}$ using dynamical dequantization of each 4-bit number (or number block):

$$X_{ij} = S_{\text{block}_i} \times P_{ij}. \tag{3.11}$$

This dequantization happens whenever the respective layer containing the corresponding number block is called in the forward pass. Reducing the memory requirements for the model weights in turn leads to memory-efficient finetuning.

## 3.3 AI-Mandel for Scientific Discovery in Quantum Optics

In this thesis, we investigate the finetuning of open-source LLMs in the context of AI-Mandel [13], an agentic system that generates and implements ideas for quantum optics experiments. To understand later prompts, idea outputs and evaluations of our finetuned models on a high-level, this section provides a short overview of the AI-Mandel physics background and its architecture.

AI-Mandel is specialized on quantum optics experiments for the generation and manipulation of photons. Photons are the individual particles of light and are an important resource for quantum communication [67, 68], quantum computing [69], and quantum-enhanced measurements [70]. Generated ideas often involve quantum networks, where different devices are connected through quantum states instead of classical signals. They are often implementable as table-top experiments with different kinds of photon sources and standard linear optics elements such as beam splitters.

The agentic system not only generates ideas, but its agents also have access to the discovery tool PyTheus [19], which can automatically find possible implementations of a given idea. The system architecture is divided into an *idea generation module* and an *idea implementation module* as shown in Fig. 3.1.

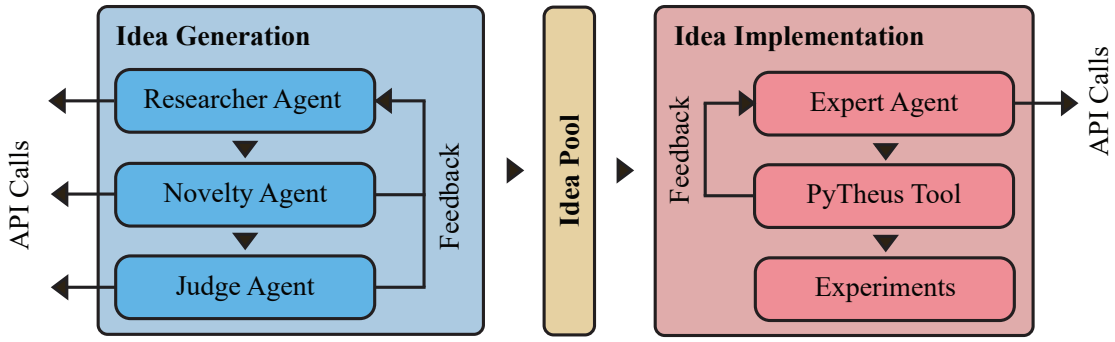



Figure 3.1: The AI Mandel architecture consists of two modules: The idea generation and idea implementation modules. Both of these hosts different agents that generate ideas, give feedback, or try to implement the ideas using PyTheus, an automated experiment discovery tool. The two modules communicate through an idea pool and use the LLM agents through black-box API calls.

The *idea generation module* consists of three agents: the researcher, novelty and judge agent. The *researcher agent* uses in-context learning by conditioning the LLM on extra information given in the prompt, such as information about the domain, components that the ideas can use, and examples of existing experiments. It is tasked to generate novel, interesting ideas that are implementable with PyTheus. The task of the *novelty agent* is to compare the outputs of the researcher with existing ideas in the idea pool and other reference ideas, to estimate their novelty. The *judge agent* gets extra information about PyTheus and is responsible for evaluating if a given idea can be implemented within the domain of that tool.

Both the *novelty agent* and the *judge agent* can accept or reject ideas with additional feedback. In case of rejection, the *researcher agent* gets the feedback and needs to come up with an improved idea. Ideas that successfully pass through all the idea generation

agents get added to an idea pool that is, in turn, the source of examples for the *novelty agent*.

The *idea implementation module* consists of the *expert agent* and the PyTheus [19] discovery tool. It takes an idea from the idea pool, translates it into the `JSON` configuration file that is needed for PyTheus and then tries to run the discovery tool. PyTheus returns errors and debugging information if the configuration file can not be executed. The *expert agent* can then try again and improve the configuration file. If PyTheus executes, the resulting experiment is saved for human interpretation.

All agents are implemented as instances of GPT-o4-mini and are therefore based on closed-source models only accessible through APIs. Using these agents, AI-Mandel resulted in 187 distinct ideas for experiments, out of which 184 could be implemented with PyTheus, 7 were categorized as especially promising by domain experts, and 2 have been developed into full, independent research papers [20, 21]. In this thesis, we are mainly focused on the researcher agent that generates ideas. While AI-Mandel uses closed-source models that cannot be easily adapted to more specific tasks or different kinds of ideas, this thesis explores the potential of replacing these models with finely controllable open-source LLMs which can steer AI-Mandel outputs towards higher-quality and different idea subfields.

## 3.4 Ranking Techniques

In this section, we provide a short overview of ranking techniques that will be used in later parts of this thesis to quantify the performance of different models and prompts.

### 3.4.1 Elo Ranking

The Elo ranking was initially developed for ranking chess players based on relative win rates [71]. In recent years, Elo gained popularity for ranking LLM models based on pairwise comparisons that yield a record of win rates between pairs of models [72, 73, 74]. Each model starts with a default ranking $R_D$, which gets updated with each comparison. For any pairwise comparison between model $i$ with ranking $R_i$ and model $j$ with ranking $R_j$, Elo provides an expected score using the scaled, logistic function

$$E_i = \frac{1}{1 + 10^{(R_j - R_i)/400}} \tag{3.12}$$

and

$$E_j = 1 - E_i. \tag{3.13}$$

A pairwise comparison based on human or LLM preference then yields actual binary scores $S_i$ and $S_j$ where the winner gets 1 and the loser gets 0. The model rankings are updated using

$$\begin{aligned} R'_i &= R_i + K(S_i - E_i) \\ R'_j &= R_j + K(S_j - E_j), \end{aligned} \tag{3.14}$$

where $K$ is an update factor that decides how fast the ranking can change.

### 3.4.2 Bradley-Terry Model

Since the Elo ranking can exhibit high volatility and is based on a sequential order of the pairwise comparisons [75], an alternative generalization and statistically sound model for pairwise comparisons is the Bradley-Terry Model [76] of which Elo is an online approximation.

Here, the probability $\Pr(i > j)$ that model A beats model B is defined by

$$\Pr(i > j) = \frac{p_i}{p_i + p_j}, \tag{3.15}$$

where $p_i$ and $p_j$ are the strengths of model $i$ and $j$. The Bradley-Terry method then estimates these strengths by maximum likelihood estimation which entails the maximization of the log-likelihood function over the parameter vector $\mathbf{p} = [p_1, ..., p_n]$ defined by

$$l(\mathbf{p}) = \sum_{i,j} \left[ w_{ij} \ln(p_i) - w_{ij} \ln(p_i + p_j) \right], \tag{3.16}$$

where $w_{ij}$ is the number of times model $i$ beat model $j$. This expression has only a single maximum [77], so the strengths $p_i$ and $p_j$ can be iteratively updated using a formula like:

$$p'_i = \frac{\sum_j w_{ij} \dfrac{p_j}{p_i + p_j}}{\sum_j \dfrac{w_{ji}}{p_i + p_j}}. \tag{3.17}$$

The Bradley-Terry method thereby results in statistically estimated model strengths that can be used as a global model ranking.

## 3.5 Diversity Measures

In order to check for a loss of diversity or overfitting to specific ideas from the finetuning process, we develop a diversity estimation that is based on sentence embeddings and dimensionality reduction techniques. Here we provide an overview of both.

### 3.5.1 Embeddings

Text embeddings [78] are vector representations of sentences, and these embeddings are designed to perform well on retrieval and similarity tasks. A large corpus of text can be automatically converted into a labeled dataset by assuming that adjacent sentences are similar, whereas random pairs of sentences are dissimilar.

Text embedding models are trained by batching $M$ pairs of similar sentences and computing an $M \times M$ logit matrix where each entry is the similarity metric $s_{ij}$ between sentence $i$ and sentence $j$ multiplied by an additional exponential scaling constant $\tau$:

$$s_{ij} = \mathrm{sim}(i, j)\ \exp(\tau). \tag{3.18}$$

Here, the similarity between two text embeddings $v_i$ and $v_j$ for the sentences $i$ and $j$ can be quantified using the cosine similarity:

$$\mathrm{sim}(i, j) = \frac{v_i \cdot v_j}{\|v_i\|\ \|v_j\|}. \tag{3.19}$$

A symmetric, contrastive cross-entropy loss [78] then favors similar embeddings between sentences of the same pair while penalizing similar cross-pair embeddings:

$$\begin{aligned} l_r &= \frac{1}{M} \sum_{i=1}^{M} \left( -\log \frac{\exp(s_{ii})}{\sum_{j=1}^{M} \exp(s_{ij})} \right) \\ l_c &= \frac{1}{M} \sum_{j=1}^{M} \left( -\log \frac{\exp(s_{jj})}{\sum_{i=1}^{M} \exp(s_{ij})} \right) \\ \mathcal{L} &= \frac{l_r + l_c}{2}. \end{aligned} \tag{3.20}$$

This loss $\mathcal{L}$ ensures that generated embeddings respect the semantic similarities between sentences.

Text embeddings can serve as a diversity measure by calculating the spread of text embeddings as the average distance from the embedding centroid [79]:

$$\text{Spread}(X) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{v}_i - \bar{\mathbf{v}}\|_2 \qquad \text{with} \qquad \bar{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}_i, \tag{3.21}$$

where $\mathbf{v}_i$ is the vector embedding of text $i$, $\bar{\mathbf{v}}$ is the embedding centroid, and $N$ is the number of texts. Alternatively, they can be visualized as a lower-dimensional 2D projection as discussed in the next section.

### 3.5.2 UMAP

To visualize text embeddings in a 2D space, we use UMAP (Uniform Manifold Approximation and Projection) [80], a common dimensionality reduction technique. It first builds a weighted neighborhood graph for the high-dimensional data points, where weights indicate pairwise distances. It then initializes lower-dimensional points and optimizes the position of these using attractive and repulsive updates along the edges until lower- and higher-dimensional neighborhood graphs match.

UMAP usually exhibits better run-time than competing methods such as t-SNE [81, 82] and better preserves global structures of high-dimensional embedding spaces. We use it to visualize the embeddings of our generated idea outputs in 2D space.

# 4 Methods

Our approach of finetuning LLMs consists of four separate steps. First, we test different open-source LLMs on the idea generation task and select the best one to be finetuned (section 4.1). Second, we compare different idea generation prompts and select the best-performing one as the prompt for dataset generation (section 4.2). We then finetune the selected model with different algorithms based on datasets generated from the selected prompt (section 4.3) and finally evaluate model performance via different metrics that consider idea quality and diversity.

## 4.1 Model Selection

We consider three different open-source LLMs as base models that could be finetuned to improve idea generation capabilities: Llama3.1-8B [48], Gemma-3-27B [49] and GPT-OSS-20B [50]. Our methodology to select between these three models is shown in Fig. 4.1.

To select a model that works best for idea generation in the context of AI-Mandel [13], we first construct an idea generation prompt inspired by the original AI-Mandel researcher. Box A.1 shows one version of this prompt. We use all prompting techniques described in section 3.1.4: role prompting to assign the model the persona of a quantum researcher, self-criticism prompting to make it refine its generated ideas, chain-of-thought prompting by making use of internal reasoning capabilities and explicitly asking for the reasoning behind generated ideas and in-context learning by providing it information about what experiments PyTheus [19] can handle.

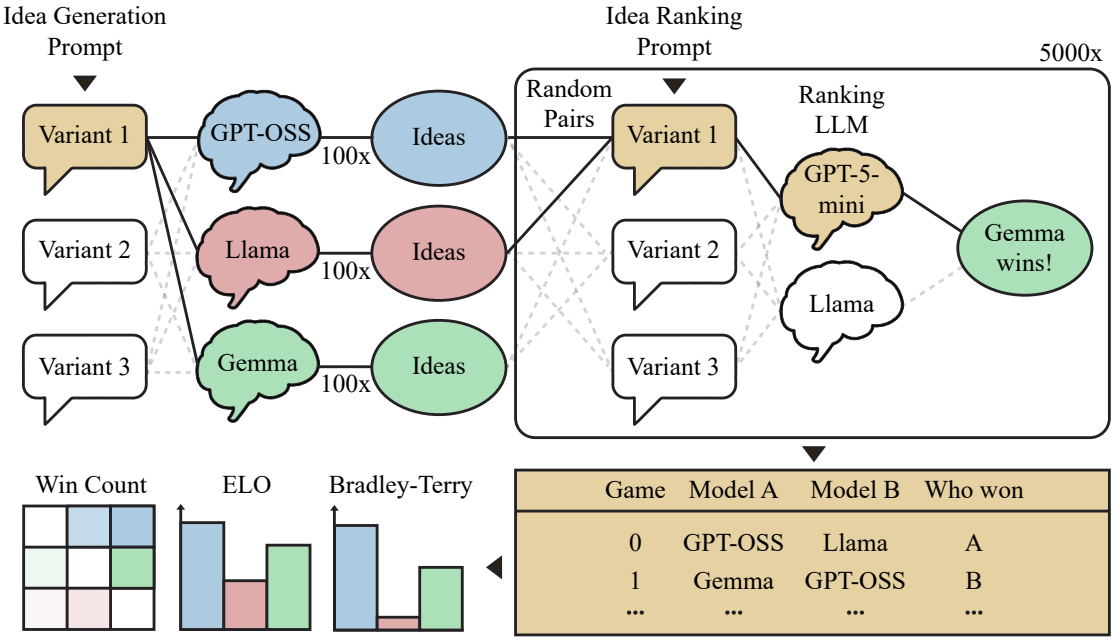



Figure 4.1: The model selection process. For a given idea generation prompt variant, we generate 100 ideas for each open-source LLM. We then randomly select 5000 idea pairs and use a given idea ranking prompt variant together with one of two LLM judges to determine the winner of the pairwise comparison. Yellow indicates one possible combination of idea generation prompt, ranking prompt and judge variants which results in a dataset of comparisons. This data is then evaluated using different metrics.

For each LLM under consideration, we then use the idea generation prompt to generate 100 ideas, each consisting of a maximum of 8192 tokens. From these idea pools, we then form 5000 random pairs of ideas from different models and insert these into an idea ranking prompt that is input to an LLM judge.

As we make use of LLMs as judges, we face the issue of potential biases in the judgments. It is known that even powerful LLMs are prone to simple prejudices [46]. Most of the known biases such as sentiment preference, tendencies to assign more credibility to statements made by an authority or preference for certain model names are not relevant in our use case: As the idea generation prompt is always the same, the generating model will not suddenly incorporate different sentiment into the outputs. Similarly, there are never statements about authority, model names and other explicit biases. Common biases that do apply in our case are positional preferences in ratings, verbosity bias that leads to preferring longer outputs, and inclinations to attend to specific wordings or phrases.

Our model selection methodology already takes these biases into account: To prevent

preference for specific wordings, we paraphrase the idea generation prompt in three different variants, shown in the boxes A.1, A.2, and A.3 in the appendix. For the same reason, we also paraphrase the idea ranking prompt in multiple variants, shown in the boxes B.1, B.2, and B.3 in the appendix.

To mitigate potential bias towards idea length, we additionally summarize the ideas from all models using GPT-OSS-20B to a single sentence and evaluate model performance using these similar-length ideas as input to the ranker prompt.

Now, to mitigate biases where LLM judges favor answers generated by themselves or within their model family, we deploy two different LLM judges, GPT-5-mini and Llama3.1-8B. These judges get the ranking prompts with the inserted idea pairs from two different models as input and are then asked to choose between the two ideas based on novelty and concreteness.

Each combination of idea generation prompt, idea ranking prompt, and LLM judge variant produces a database of 5,000 games. We evaluate these games using the three metrics introduced in Section 3.4: win count, Elo, and Bradley–Terry. Because pairwise comparisons are sampled at random to avoid potential bias, the win count alone may be misleading because it does not account for variation in opponent strength across randomly sampled pairwise comparisons. Elo and Bradley–Terry therefore serve as complementary metrics. Both aim to estimate the underlying quality of opponents, which is only indirectly observed through these random comparisons. Elo can be viewed as an iterative and order-dependent approximation of pairwise strength, whereas Bradley-Terry provides a more statistically principled approach based on an explicit probabilistic model. We report all three metrics for completeness.

Based on the highest scores of these three metrics across all combinations of prompts and LLM judges we then choose the best-performing open-source LLM for the finetuning process.

## 4.2 Prompt Selection

As a second step in our pipeline, we compare different idea generation prompts with diverse prompting techniques, contexts and information to create a solid baseline for the finetuning procedures. The prompt selection methodology is shown in Fig. 4.2.

We use a two-stage procedure where we first rank prompt variants based on pairwise games of the corresponding idea outputs. In the second stage, we combine these prompts based on their ranking to get a final prompt combination that we fix for the subsequent dataset generation and finetuning procedures.

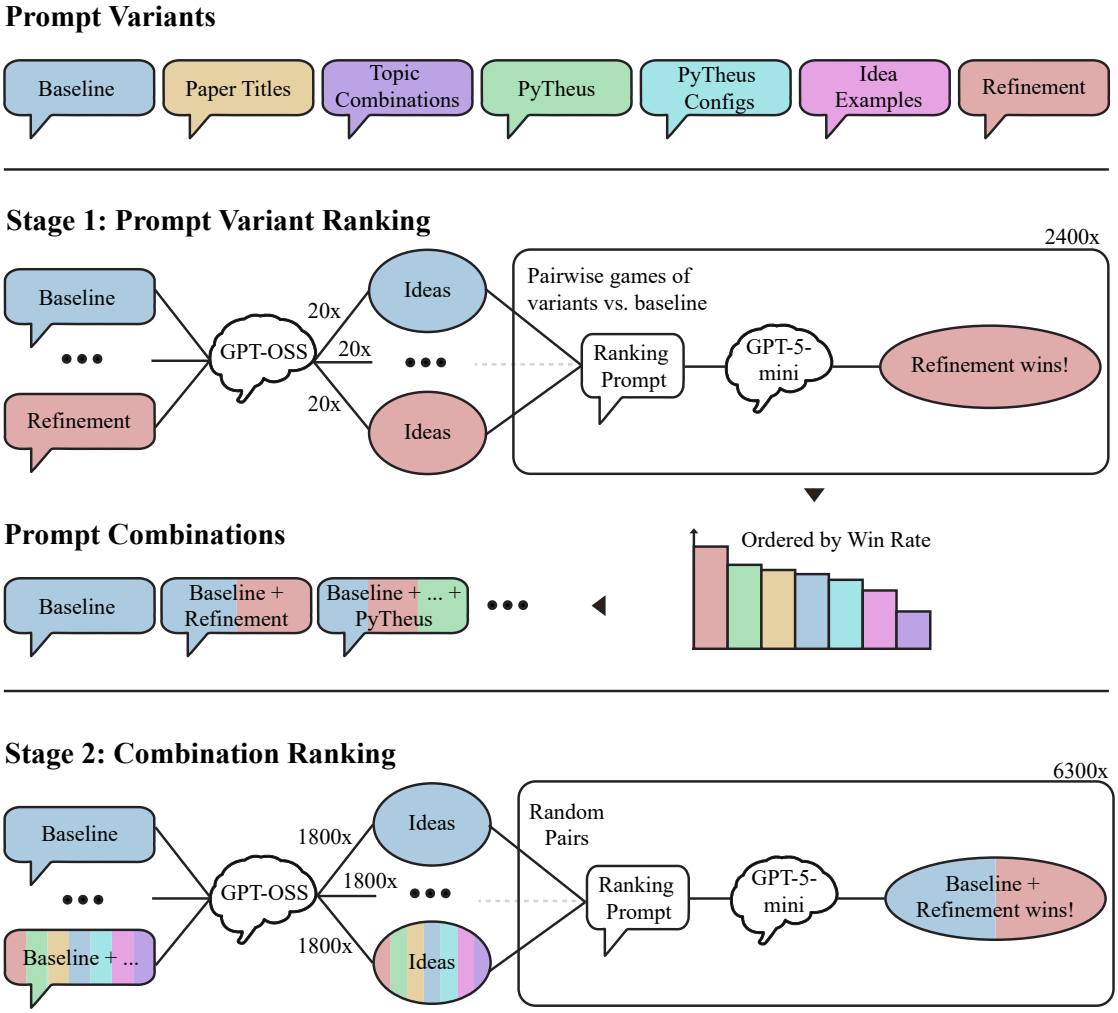



Figure 4.2: The prompt selection process. We start with seven prompt variants that contain different information. For each prompt variant we generate 20 ideas using GPT-OSS-20B. From these ideas, we randomly select 2400 pairwise comparisons (each prompt plays 400 games against the baseline) and use GPT-5-mini to choose a preference. We order the seven variants by their win rate and then add them incrementally to the baseline prompt, according to the win rate ordering. We then generate 1800 ideas per variant combination using GPT-OSS-20B and randomly select 6300 pairwise comparisons (each variant against each other 300 times) in which we choose a winner using GPT-5-mini. The resulting prompt is used in all downstream tasks.

For stage one, besides the baseline idea generation prompt in box A.1, we test six other variants:

- Prompt with three randomly selected arXiv paper titles from quantum physics (Box C.1).
- Prompt combining two randomly selected quantum physics topics into one idea (Box C.2).
- Prompt with detailed PyTheus introduction (Box C.3).
- Prompt with detailed PyTheus introduction and configuration examples (Box C.4).
- Prompt with idea examples that were already explored by PyTheus (Box C.5).
- Prompt asking the model to critically refine its own ideas for novelty (Box C.6).

We use our selected LLM from the previous model selection process to generate 20 idea outputs per prompt variant. As for the model selection, we use GPT-5-mini as a ranker LLM that gets a ranking prompt (Box D.1) with a randomly chosen idea pair as input and chooses a preference based on novelty, feasibility and scientific interest [83]. Each prompt is ranked in 400 pairwise comparisons against the baseline prompt, resulting in 2400 total comparisons. We then start with the baseline prompt and create variant combinations by incrementally adding the context of different prompt variants in the order of their win rate. This incremental combinatorial procedure tries to test some of the possible prompt combinations while taking into account the monetary and computational cost of running too many comparisons.

For stage two, we repeat the ranking procedure with these combinations. For each one we generate 1800 ideas which are then used in 6300 randomized pairwise games where each prompt variant plays against each other 300 times. We use GPT-5-mini as the LLM judge which determines a winner for each comparison. We determine the number of pairwise games to ensure statistical significance of the performance gap between the two best-performing prompts and demonstrate this significance by computing z- and p-values under the null hypothesis of both prompts being equally strong. From the resulting comparison database, we can select the best performing variant combination as the baseline for the subsequent dataset generation and finetuning procedures.

## 4.3 Finetuning Procedure

Having selected our open-source LLM and an idea generation prompt, we now describe our data generation process and the algorithms used to finetune the model (Fig. 4.3).

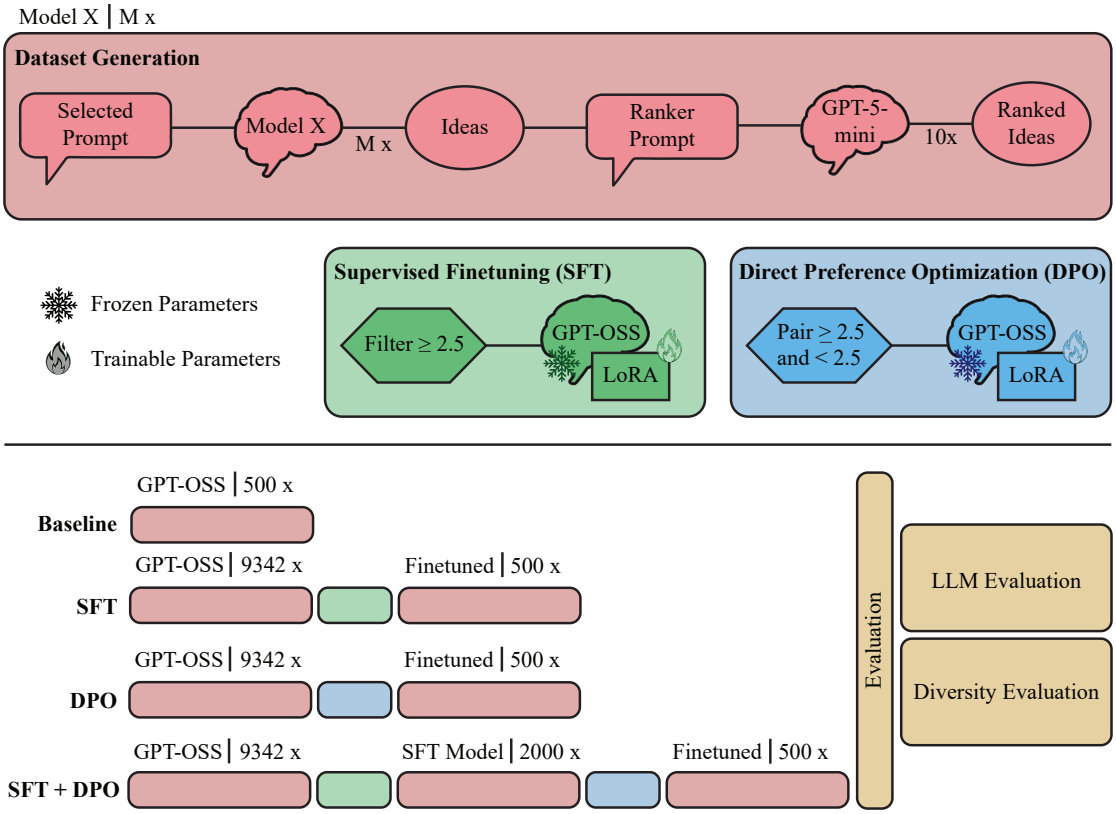



Figure 4.3: The finetuning process. We combine three different building blocks of dataset generation, SFT and DPO in four different model evaluations. In the dataset generation block we use our selected idea generation prompt together with a given model $X$ to generate $M$ ideas that are then inserted into a ranker prompt and scored by GPT-5-mini by averaging over 10 rankings. In the SFT block, we first filter the input idea dataset for ideas that are scored higher than 2.5 and subsequently finetune GPT-OSS-20B using LoRA and an SFT loss. For the DPO block we pair ideas that are scored above and below 2.5 and finetune GPT-OSS-20B using LoRA and a DPO loss.

### 4.3.1 Idea Dataset Generation

With the selected model and idea generation prompt we generate a finetuning dataset with 9,342 ideas (Fig. 4.3). We then rate these ideas by inserting each into a ranker prompt (Box D.1) that instructs GPT-5-mini to rate each idea based on its novelty, feasibility and scientific interest. The rating is an integer in the range 1 to 3, where 1 is the worst and 3 the best.

When rating ideas only a single time, GPT-5-mini's randomness would sometimes lead to very different results. Fig. 4.4 shows the accumulative mean of 100 GPT-5-mini ratings

for four example ideas. As a tradeoff between accuracy and cost we rate every idea 10 times and use the mean of these as the final rating. This results in an idea database with robust rankings that can be used for finetuning.

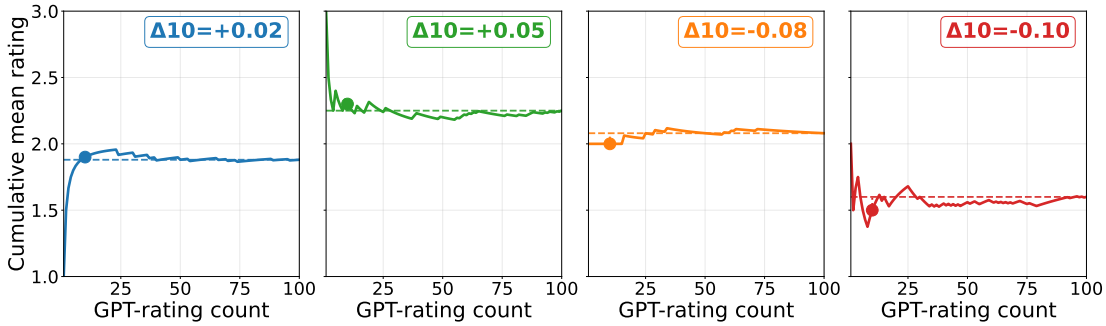



Figure 4.4: Cumulative mean rating of GPT-5-Mini over 100 ratings on 4 example ideas. The legends indicate $\Delta 10$, the offset of a mean over the first 10 ratings versus the mean over 100 ratings.

### 4.3.2 Finetuning Algorithms

For finetuning our selected model, we concentrate on Supervised Finetuning (SFT) and Direct Preference Optimization (DPO). As shown in Fig. 4.3, we always start from the initial dataset of generated ideas based on which we test three different finetuning variants: pure SFT, pure DPO and DPO on top of SFT.

For SFT, we constrain the dataset to contain only good ideas by selecting only ideas that were rated 2.5 or above. This results in around 3000 ideas. To manage the memory cost of GPT-OSS-20B, we make use of the popular `transformers` library from HuggingFace [84]. We use the GPT-OSS-20B model with MXFP4 quantization and apply LoRA from the `peft` library [85]. For LoRA, we use standard parameters with rank 8 and target all linear modules of the frozen GPT-OSS-20B model which results in around 15 million LoRA parameters. For training, we use Nvidia H100 GPUs with 80GB memory and trainer classes and objective functions from the `trl` library [86]. We run the SFT finetuning training with AdamW [87], a batch size of 16 and a learning rate of 0.0002 over 9 epochs. After each epoch, we evaluate the model by estimating the idea quality of 60 generated outputs by our LLM judge GPT-5-mini and finally use the model that achieves the best idea quality estimate for further evaluation.

For DPO, we combine ideas from the dataset into 5000 pairs where one idea was rated 2.5 or above and the other was rated below 2.5. We use the same training settings as for SFT. Again, we run the finetuning training with AdamW, a batch size of 16 and a learning rate of 0.0002 over 10 epochs. We use the final model for further evaluation.

For DPO on top of SFT, we take the final model and the tuned LoRA weights from the SFT training and generate a new dataset of around 2000 ideas from that model using the same procedure as for the initial dataset generation. We then run the same procedure as for the pure DPO model on this dataset.

Our finetuning procedure therefore, results in three finetuned models that we then evaluate against the baseline model using the methods described in the following section.

## 4.4 Evaluation Methods

In this section, we discuss the techniques that we use to evaluate the quality and diversity of the outputs from the three finetuned models. We first evaluate the quality of the ideas according to the LLM judge considering known LLM-as-a-judge biases. We then evaluate how the finetuning affects the diversity of the idea space and finally integrate our finetuned SFT + DPO model into the AI-Mandel system to test how many generated ideas are accepted by the other agents within that system.

### 4.4.1 Idea Quality Evaluation

For the idea quality evaluation, we generate 500 outputs for each finetuned model using the same procedure as for dataset generation with an average over 10 rankings per idea output (Fig. 4.3). Based on this, we compare the rating distribution and the average rating of all our models in Fig. 5.7.

We then use different statistical methods to estimate the uncertainty and significance of these results. We first estimate model performance uncertainty by generating $1,000$ bootstrap resamples, sampling with replacement from the 500 idea outputs for each model. This bootstrap distribution is then used to quantify uncertainty in the average rating via 95% confidence intervals, reported in Fig. 5.8.

In addition, we perform permutation tests on the 500 idea outputs to assess whether the pairwise difference between finetuned models and the baseline model is statistically significant under the null hypothesis of no difference in performance. We repeatedly shuffle the ratings from each finetuned model with the ratings from the baseline and compute the average rating difference. We then report the p-value as the fraction of shuffled datasets that produce a difference at least as large as the actual observed one. The results are shown in Fig. 5.9.

Ideally, the ranking should reflect the quality of the idea content instead of being influenced by the length or certain phrases within the idea. To mitigate such potential LLM biases we additionally run all evaluations a second time using GPT-OSS-20B generated

one-sentence versions of all ideas. The results of this additional verification are shown in Fig. 5.7, Fig. 5.8, and Fig. 5.10.

### 4.4.2 Diversity Evaluation

A valid concern when finetuning the model via SFT and DPO is that the model memorizes or overfits to a small set of high-quality ideas that then represent its entire learned knowledge. To test for such a collapse in idea diversity, we make use of the OpenAI embedding model and convert 500 generated ideas from the GPT-OSS-20B baseline and each finetuned model to an embedding space. We evaluate the diversity of the idea space by making use of the average distance from the embedding centroid, introduced in section 3.5.1.

In addition, we map the embeddings to a 2D space by making use of the UMAP algorithm and inspect the resulting idea clusters of finetuned and baseline models. We then remove specific idea clusters from the training dataset and retrain the models from scratch using this smaller dataset. By evaluating if the finetuned model still produces ideas in the removed cluster, we test its out-of-distribution capabilities and ensure that the finetuning does not overwrite previous knowledge of the removed idea cluster.

The results of all diversity evaluations are reported in section 5.3.2.

### 4.4.3 AI-Mandel Integration

In order to further test and evaluate our finetuning procedure, we run our SFT + DPO model in the context of AI-Mandel, the agentic system for idea generation in quantum optics described in section 3.3 and Fig. 3.1. For the purpose of testing our finetuned models, the idea generation module is sufficient as only this process populates the idea pool which is later used by the implementation module in an independent process. We replace the API call of the researcher agent once with the baseline GPT-OSS-20B model and once with our finetuned SFT + DPO model. Since we have not yet finetuned our models on prompts with extra feedback, we do not make use of AI-Mandel's feedback mechanism and only compare how many researcher agent ideas from each model are accepted by the novelty and judge agents. Here, we follow the original AI-Mandel system and use the GPT-5-mini OpenAI API for the novelty and judge agents.

We run two variants of this test, each with 500 iterations: First, without the idea pool, where novelty and judge agents only get the output of the previous agent as input. Second, with the idea pool, where all existing ideas that were previously accepted are added as additional context to the prompt of the novelty agent. This ensures diversity by checking incoming researcher ideas against already accepted candidates. We statistically evaluate the difference in acceptance counts using a two-sided Fisher's exact test [88].

This test evaluates the null hypothesis that the acceptance outcome is independent of model choice, that is, that the finetuned model and the baseline have equal acceptance probabilities.

We present all results of the AI-Mandel integration in section 5.3.3.

# 5 Results

In this chapter, we present the results of the methodology outlined in chapter 4. We start with results from the model selection process in section 5.1. In section 5.2, we present the results from the two stages of the prompt selection process and in section 5.3, we report the results from the idea quality evaluation and finally the diversity test and AI-Mandel integration results.

## 5.1 Model Selection Results

For model selection, we follow the procedure shown in Fig. 4.1 and explained in section 4.1. We first use ranker prompt variant 1 and all 3 idea generation prompts and subsequently combine idea generation prompt variant 1 with ranker prompt variants 2 and 3. These 5 combinations are tested with both LLM judges GPT-5-mini and Llama-3.1-8B. Fig. 5.1, 5.2, and 5.3 respectively show the Elo, Bradley-Terry and win rate ranking results for all prompt and judge combinations on the original ideas. Additionally, they show the results for running these prompt combinations with GPT-5-mini as the judge on single-sentence summaries generated by GPT-OSS-20B. In all figures, the ordering of the model rankings is the same: GPT-OSS-20B has the highest ranking, then comes Gemma-3-27B before Llama-3.1-8B.

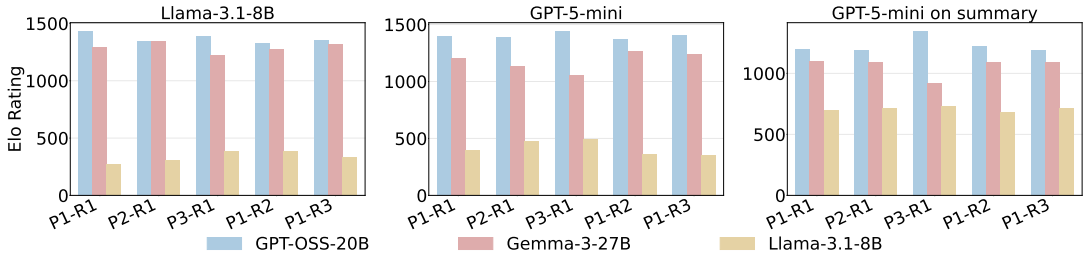



Figure 5.1: Elo scores for the pairwise comparisons of the model selection process. Colors indicate the LLMs used to generate the ideas. P1, P2, P3 are the idea generation prompt variants 1 to 3, while R1, R2, R3 indicate the three ranker prompt variants. Each plot corresponds to one ranker LLM on original or summarized ideas.

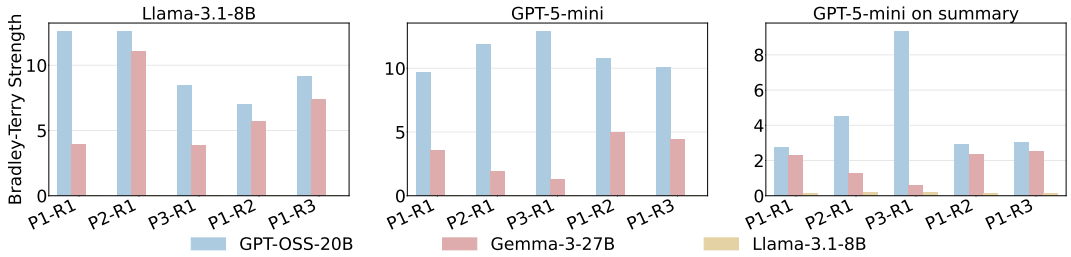



Figure 5.2: Bradley-Terry strengths for the pairwise comparisons of the model selection process. Colors indicate the LLMs used to generate the ideas. P1, P2, P3 are the idea generation prompt variants 1 to 3, while R1, R2, R3 indicate the three ranker prompt variants. Each plot corresponds to one ranker LLM on original or summarized ideas.

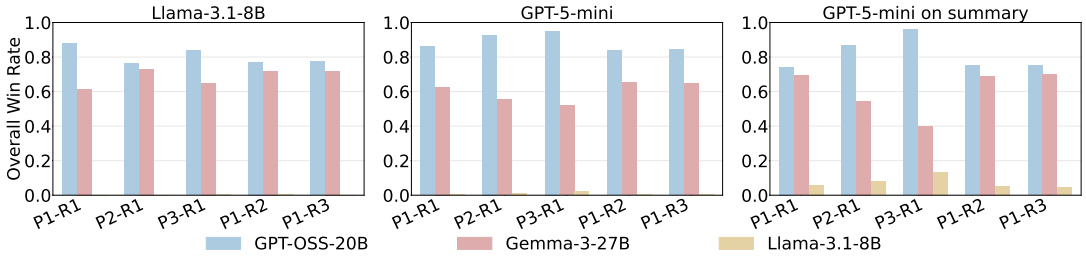



Figure 5.3: Win rates for the pairwise comparisons of the model selection process. Colors indicate the LLMs used to generate the ideas. P1, P2, P3 are the idea generation prompt variants 1 to 3, while R1, R2, R3 indicate the three ranker prompt variants. Each plot corresponds to one ranker LLM on original or summarized ideas.

Fig. 5.4 shows all 25,000 pairwise comparisons from the model selection process with all prompt combinations from Fig. 5.1 for the two ranker models Llama-3.1-8B and GPT-5-mini. In each pairwise comparison, these models choose between an idea at position A and an idea at position B of the ranker prompts in boxes B.1, B.2, and B.3. The length of the ideas in the respective positions is given on the x-axis and y-axis. Llama-3.1-8B chooses ideas at position A significantly more often than at position B, while GPT-5-mini slightly prefers position B. With pink and blue indicating the respective winner of the comparison, the upper left and lower right being uniformly colored indicates that longer ideas consistently outperform shorter ones.

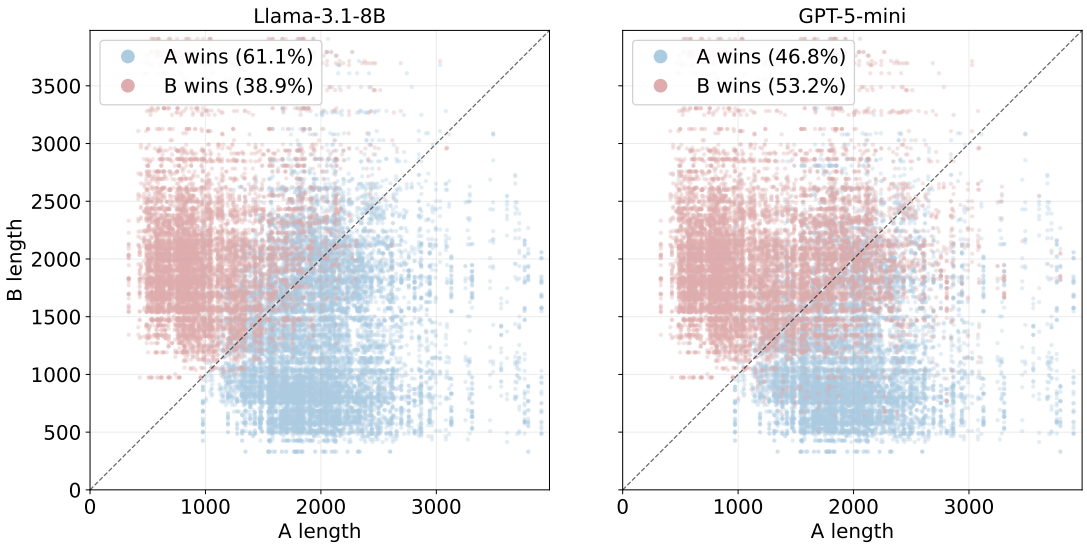



Figure 5.4: Idea length at position A versus idea length at position B of 25,000 pairwise comparisons ranked by Llama-3.1-8B on the left and GPT-5-mini on the right for the model selection process. Blue color indicates that idea at position A won the pairwise comparison, while pink indicates that the idea at position B won.

From the results in this section, we choose the open-source model GPT-OSS-20B for the idea generation process and as a finetuning baseline.

## 5.2 Prompt Selection Results

By using GPT-OSS-20B as our idea generation model and following the two-stage prompt selection process shown in Fig. 4.2 and explained in section 4.2, we first present a win rate ordering of our seven compared prompt variants in Fig. 5.5. Three prompt variants outperform the baseline while three others do not lead to improvements in LLM judge rankings. There is a significant win rate difference between the self-refinement prompt that performs best overall and the second-best performing prompt variant that contains an additional introduction to the PyTheus tool. The prompt variant in which GPT-OSS-20B is asked to combine two random quantum optics ideas performs the worst.

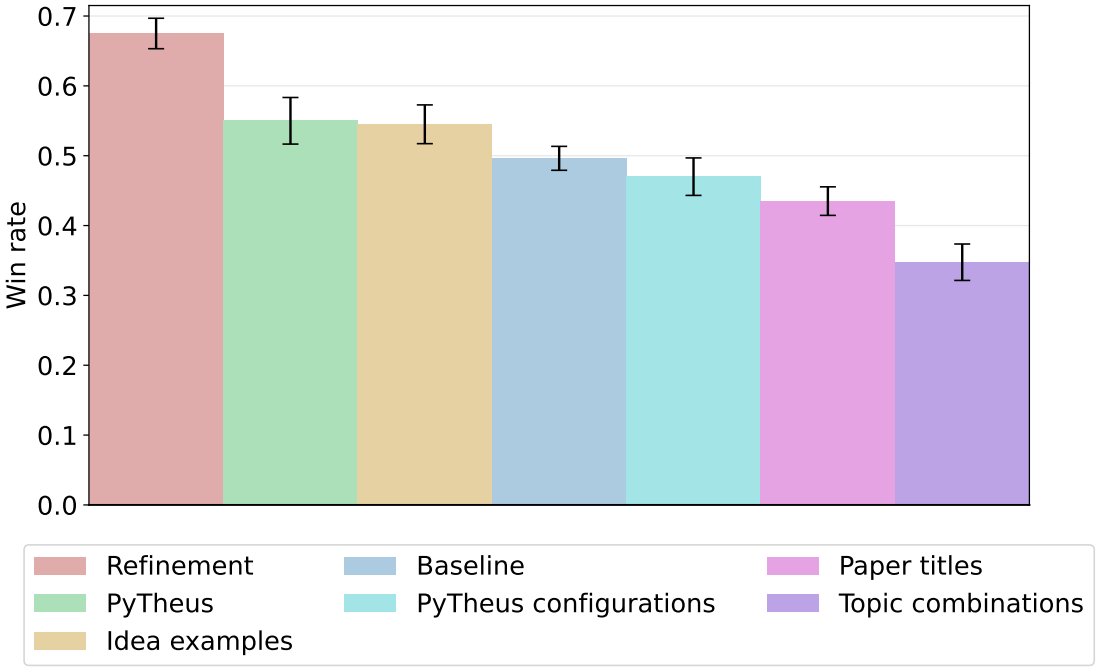



Figure 5.5: Win rate results for the seven prompt variants of the first stage of the prompt selection process. With each prompt, we generated ideas using GPT-OSS-20B and evaluated them according to our prompt selection process from section 4.2.

Fig. 5.6 shows the results of the second stage of the prompt selection process: the comparison between incremental combinations of the seven prompt variants introduced in section 4.2. Adding the request for self-refinement to the baseline prompt yields the best performance, while a combination of all variants performs the worst. The baseline prompt performs the second-worst.

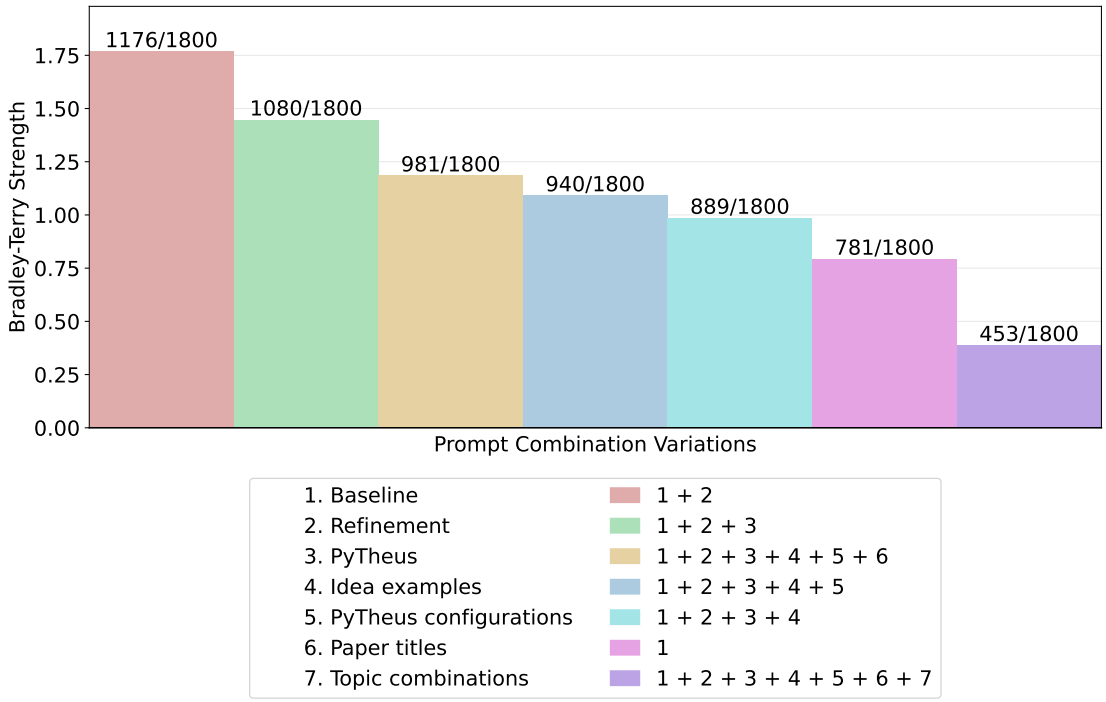



Figure 5.6: Bradley-Terry strengths for the seven incremental prompt combinations of the second stage of the prompt selection process. With each prompt, we generated ideas using GPT-OSS-20B and evaluated them according to our prompt selection process from section 4.2.

To test whether the best performing prompt is statistically significantly stronger, we can assume the null hypothesis that the two top performing prompts are equally strong and then calculate the p-value, the probability that we would observe the difference between those two prompts in Fig. 5.6 under this hypothesis. We perform the calculation in chapter F of the appendix. For the top two performing prompts, we get a p-value of around 0.001.

We further discuss the prompt ranking figures and the significance calculation in section 6.2 and choose the best-performing combination of baseline and self-refinement prompt for the idea dataset generation.

## 5.3 Evaluation Results

In this section, we will first present the idea quality rankings from the GPT-OSS-20B baseline model and all three SFT, SFT + DPO, and DPO finetuned models in section 5.3.1. Section 5.3.2 then evaluates the idea space of these four models to investigate if

the finetuning leads to a loss in idea diversity. Finally, in section 5.3.3, we then integrate the finetuned SFT + DPO model into AI-Mandel and report how many of the generated ideas pass the other AI-Mandel agents.

### 5.3.1 Idea Quality Evaluation Results

The finetuning procedure from section 6.3 and Fig. 4.3 results in a baseline model and three finetuned models: SFT, SFT + DPO, and DPO. To provide some better understanding of the ideas that these models are finetuned to generate, box 5.1 shows a manually shortened idea example generated by our finetuned SFT + DPO model. The original, high-detail idea can be found in box E.1 of the appendix.

**Hybrid high-dimensional entanglement swapping for quantum networks**

We propose a much more compact entanglement-swapping experiment whose central innovation is the Bell-state measurement in a hybrid polarization-orbital-angular-momentum space. Two independent SPDC sources generate photon pairs entangled simultaneously in polarization and OAM. Photons 2 and 3 are brought to a linear-optical Bell-state measurement that acts on the combined four-dimensional local Hilbert space, using q-plates to couple polarization and OAM, log-polar mode conversion to separate OAM components, and a 4×4 interferometer to project onto a subset of hybrid Bell states. A successful coincidence event then swaps the entanglement onto photons 1 and 4, preparing a nonlocal hybrid Bell state shared across the two remaining photons.

The key advance is not simply entanglement swapping, but swapping entanglement between photons that each encode two qubits in distinct degrees of freedom. This moves beyond standard polarization-only schemes and creates an experimentally accessible route to high-dimensional photonic network links with increased information capacity per photon. The proposal combines only static linear optics and existing high-efficiency components, making it realistic while still targeting a qualitatively new resource state. If realized, it would constitute the first demonstration of hybrid high-dimensional entanglement swapping, with direct relevance for multiplexed quantum communication, high-dimensional teleportation, and quantum-network architectures.

Box 5.1: Example of a manually shortened idea generated by the SFT + DPO model. The detailed idea can be found in box E.1 of the appendix.

We report results of all three evaluations described in our methods section 4.4.1. Fig. 5.7 shows the rating distribution as box and swarm plots for the 500 ideas from each of the four models, once for the original full outputs and once for one-sentence summaries generated by GPT-OSS-20B. For the full outputs, all finetuned models obtain higher mean and median ratings than the GPT-OSS baseline. They form a performance ladder from SFT over SFT + DPO to pure DPO. For the one-sentence summary, all finetuned models still outperform the baseline model in terms of mean and median. However, the SFT + DPO model now outperforms both DPO and SFT models.

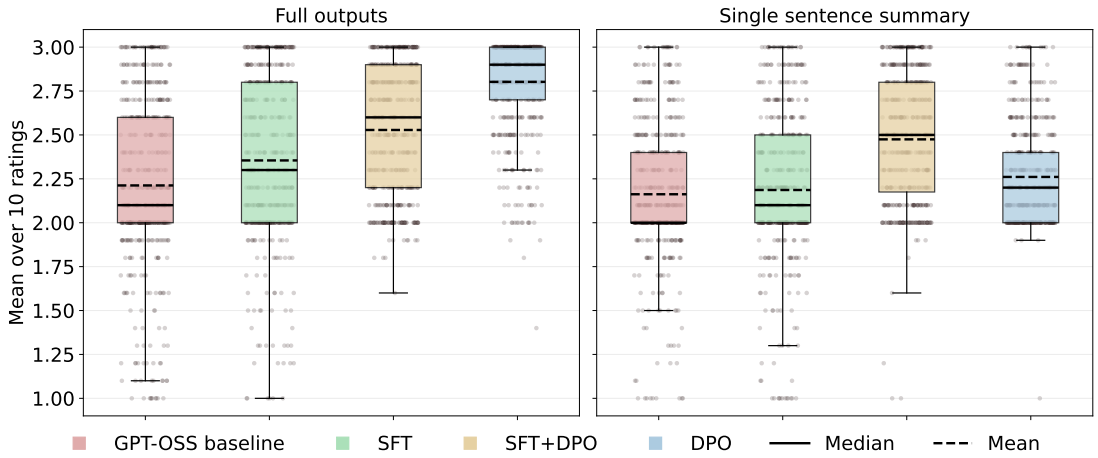



Figure 5.7: Idea quality rating distribution over 500 original and summarized ideas from the GPT-OSS-20B baseline model and all three finetuned models: SFT, SFT + DPO, and DPO. The boxes indicate the interquartile range IQR and the whiskers mark outliers outside of $\pm 1.5 \cdot \text{IQR}$.

Fig. 5.8 quantifies the uncertainty of the rating averages by showing the mean and 95% confidence intervals that are the result of our bootstrapping process explained in section 4.4.1. Again, the figure shows results for both the full outputs and the one-sentence summaries generated by GPT-OSS-20B. For the full-length ideas, the 95% confidence intervals are narrow and clearly separated across all models. For single-sentence summaries, the intervals are still narrow, but less well separated and overlap for the case of SFT and GPT-OSS-20B baseline models.

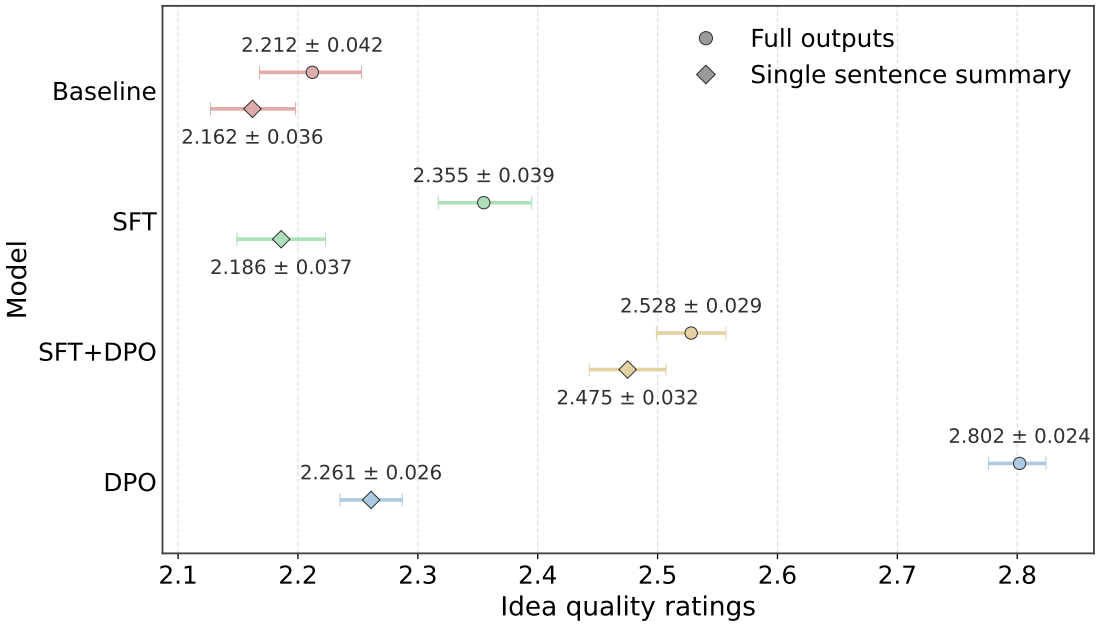



Figure 5.8: Idea quality rating uncertainty estimation via bootstrapping. The plot shows the mean idea quality rating of the 500 outputs from each model for full outputs and single sentence summaries. The bars indicate the 95% confidence intervals computed from the bootstrapping process.

Figures 5.9 and 5.10 test the significance of our rating results by showing one-sided permutation tests on the mean rating difference between each finetuned model and the baseline using $1,000$ random permutations per comparison of 500 original and summarized outputs per model. For the full-length ideas in Fig. 5.9, the probability of observing the result with the null hypothesis of the model not being better than the baseline (called p-value) is less than 0.1% for all finetuned models. For the summarized ideas in Fig. 5.10, only the SFT + DPO and DPO models have a p-value of less than 0.1%. The permutation test for the SFT model results in a p-value of around 17.5% instead.

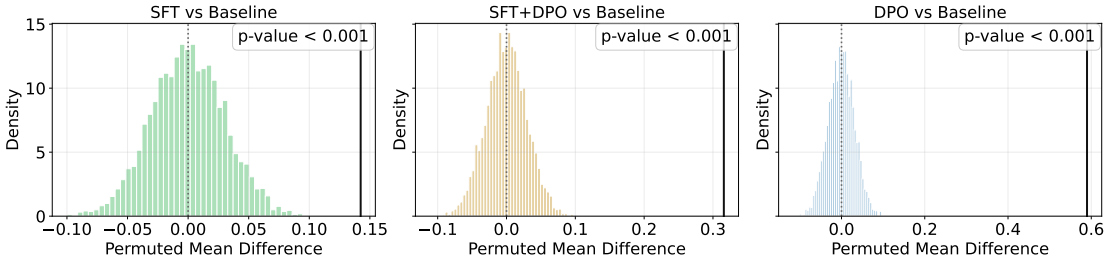



Figure 5.9: One-sided permutation tests (finetuned model > GPT-OSS-20B baseline) on 500 outputs per model and 1000 permutations. Histograms show null distributions of permuted mean differences (finetuned model - GPT-OSS-20B baseline), and vertical lines mark observed differences.

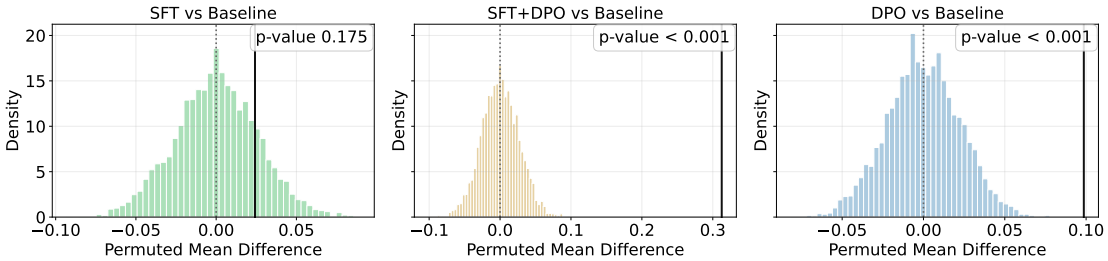



Figure 5.10: One-sided permutation tests (finetuned model > GPT-OSS-20B baseline) on 500 one-sentence summaries per model and 1000 permutations. Histograms show null distributions of permuted mean differences (finetuned model - GPT-OSS-20B baseline), and vertical lines mark observed differences.

These results show the idea quality rating distribution, quantify the uncertainty of the rating averages via the 95% confidence intervals, and test the significance of each finetuned model difference compared to the GPT-OSS-20B baseline in terms of the p-values from the permutation tests. The results will be further discussed in section 6.3.1.

### 5.3.2 Diversity Evaluation Results

As described in our methods section 4.4.2, we now present evaluations of the idea diversity for the GPT-OSS-20B baseline and all finetuned models.

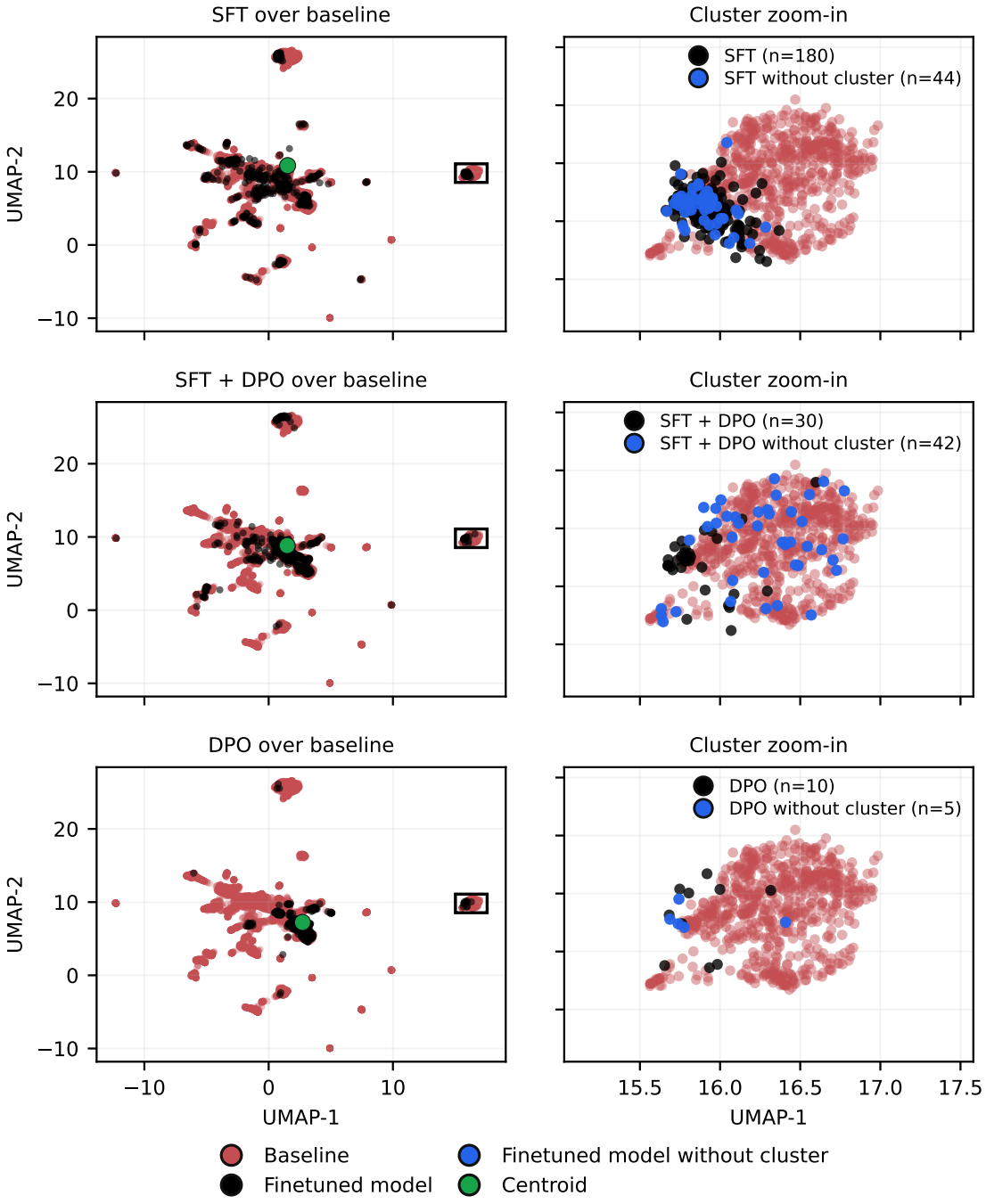



Figure 5.11: The left column shows two-dimensional UMAP projections of the idea embedding space of 500 ideas from each of the three finetuned models on top of 9,342 generated by the GPT-OSS-20B baseline. The right column shows a zoomed-in view of the boxed cluster on the right together with output embeddings from a model that was trained on a dataset including that cluster and one trained excluding that cluster.

Fig. 5.11 visualizes two-dimensional UMAP projections of the idea embedding spaces for 500 ideas from each of the three finetuned models overlaid on the embedding space of 9,342 ideas from the GPT-OSS-20B baseline. In the left column, the baseline ideas are shown in red and the corresponding finetuned model ideas in black. The SFT ideas are distributed across a region that broadly overlaps with the baseline projection, whereas the SFT+DPO and DPO ideas appear more concentrated in smaller regions of the shared space. The right columns zoom into the highlighted cluster. Black points represent ideas from models trained with this cluster included in the training data, and blue points represent ideas from models trained without that cluster. In all three cases, the cluster holds ideas for both types of training: with or without that cluster.

Table 5.1 quantifies idea diversity by reporting the average distance of idea embeddings from the embedding centroid. The GPT-OSS-20B baseline and SFT models show similar mean and median distances, while SFT + DPO and DPO have substantially lower distance values.

| Model | Mean Distance | Median Distance | Minimum Distance | Maximum Distance |
|---|---|---|---|---|
| Baseline | 7.625 | 5.291 | 0.058 | 20.069 |
| SFT | 7.609 | 5.323 | 1.014 | 15.466 |
| SFT + DPO | 3.691 | 1.673 | 0.157 | 17.639 |
| DPO | 2.157 | 2.078 | 0.047 | 19.013 |

Table 5.1: Idea diversity quantification by mean and median distance of all idea embeddings from the embedding space centroid.

Overall, both the qualitative and quantitative diversity evaluations provide consistent evidence of differences in idea diversity across models, further discussed in section 6.3.2.

### 5.3.3 AI-Mandel Integration Results

Here, we report the results from the AI-Mandel integration test described in the methods section 4.4.3. Table 5.2 shows the acceptance rates of the finetuned SFT + DPO model compared to the GPT-OSS-20B baseline over 500 runs per model. It also tests their significance by reporting the p-value of two-sided Fisher's exact tests on the acceptance rates. The p-value gives the probability of observing the results under the null-hypothesis that the acceptance rates are independent of the model choice. The exact equations for computing the p-values can be found in chapter G of the appendix.

| Setting | Agent | SFT + DPO | Baseline | $p$-value |
|---|---|---|---|---|
| With idea pool | Novelty | 9/500 (1.8%) | 14/500 (2.8%) | 0.399 |
| | Judge | 7/500 (1.4%) | 11/500 (2.2%) | 0.477 |
| Without idea pool | Novelty | 76/500 (15.2%) | 63/500 (12.6%) | 0.273 |
| | Judge | 70/500 (14.0%) | 54/500 (10.8%) | 0.150 |

Table 5.2: AI-Mandel acceptance results over 500 runs per model and setting. Reported $p$-values are from two-sided Fisher's exact tests comparing SFT + DPO against the baseline model with and without the idea pool over the novelty and judge agent. The null-hypothesis for the p-value is that the acceptance rates are independent of the model choice.

Overall, acceptance rates are low in both variants. In the setting with the idea pool, the novelty agent accepts 9 out of 500 ideas from the finetuned SFT + DPO model and 14 out of 500 ideas from the baseline GPT-OSS-20B model, corresponding to a p-value of 0.399. For the judge agent in the same setting, 7 out of 500 ideas from the finetuned model and 11 out of 500 ideas from the baseline model are accepted, corresponding to a p-value of 0.477.

In the setting without the idea pool, the novelty agent accepts 76 out of 500 ideas from the finetuned model and 63 out of 500 ideas from the baseline model, corresponding to a p-value of 0.273. For the judge agent in this setting, 70 out of 500 ideas from the finetuned model and 54 out of 500 ideas from the baseline model are accepted, corresponding to a p-value of 0.150. The AI-Mandel integration results are discussed in detail in section 6.3.3.

# 6 Discussion

This chapter focuses on the interpretation of the results presented in chapter 5. Section 6.1 motivates the reasoning behind choosing GPT-OSS-20B as the baseline model for subsequent analyses. Section 6.2 justifies the choice of idea generation prompt and explains why the selected prompt was adopted based on the previously presented results. In section 6.3, we then discuss the performance results of the finetuned models with respect to idea quality, diversity and the integration of these models into AI-Mandel.

## 6.1 GPT-OSS-20B for Idea Generation

Based on the results in section 5.1, we use GPT-OSS-20B as our main baseline model for dataset generation and finetuning. The fact that Figs. 5.1 - 5.3 consistently show GPT-OSS-20B as the best performing model across all five combinations of idea generation prompts, ranker prompts, and LLM judges, providing a robust indication for the idea generation quality of that model.

The consistency of the model ordering across all prompt and judge combinations indicates that these results go beyond a simple bias of an LLM judge towards certain phrases or textual features instead of idea content. Even though Fig. 5.4 indicates a correlation between the idea length and its success rate, the fact that all ideas summarized to a single sentence result in the same ordering rules out a simple length bias of the LLM judge.

Although Fig. 5.4 suggests a position-related imbalance for the Llama-3.1-8B judge, GPT-5-mini exhibits an imbalance in the opposite direction while yielding a similar overall model ordering. This indicates that any position biases do not significantly affect the final ranking in our setup.

Nonetheless, our analysis is mainly based on LLM judges whose ratings are difficult to trace to their actual reasoning. Human evaluations of idea quality would be ideal, but hard to achieve for time and cost reasons. Even with LLM judges, much of our methodology is constrained by monetary and computational cost. One possible extension of our work could be explicitly asking the LLM judges for additional reasoning as this often leads to improvements in terms of human alignment [89].

The purpose of our model selection process is to find a strong idea generation model that makes the subsequent finetuning more meaningful because it is applied to the strongest available open-source baseline rather than a weaker model. For this purpose, the robustness of the choice of GPT-OSS-20B seems sufficient.

## 6.2 Idea Generation Prompt

Similar to the model selection process, the goal of the prompt selection process is to find a strong enough baseline that makes the subsequent finetuning more meaningful. Fig. 5.5 shows that extending the baseline idea generation prompt with additional context can lead to clear improvements in LLM judge rankings.

In this first stage of the prompt selection process, the refinement, PyTheus and idea example prompts perform better than the baseline prompt in box A.1 of the appendix. The refinement prompt might perform well because its self-criticism prompting technique makes use of multiple reasoning iterations. The PyTheus prompt adds a clear context for quantum optics which might be ranked higher by the LLM judges because they are instructed to rate ideas for novelty, feasibility and scientific interest in that specific field. Similarly, the idea examples turn the problem into a few-shot learning problem where GPT-OSS-20B can make direct use of good idea examples within the right context.

The PyTheus configuration prompt and the prompts with random paper and topic combinations perform worse than the baseline prompt. PyTheus configurations are more of a technical detail independent of idea quality and might therefore shift the focus to the wrong direction, leading to worse results than with the baseline prompt. The prompts with random paper titles and topic combinations might restrict the model to specific fields that are far from the PyTheus context and might therefore be rejected by LLM judges that specifically rate ideas for the PyTheus context.

The results of the second stage of the prompt selection process in Fig. 5.6 show that targeted prompt combinations can lead to large improvements in LLM judge rankings while more extensive combinations tend to perform worse while often still outperforming the baseline prompt, which ranks second to last. Our statistical significance test for the performance gap between the two best-performing prompts results in the p-value of 0.001. This means that under the assumptions that the two prompts are equally good, the chance of seeing this gap between the prompts in Fig. 5.6 is around 0.1%. The best prompt therefore significantly outperforms the second-best prompt when using GPT-5-mini as the LLM judge.

Our method to combine the different prompts is restricted by the monetary and computational costs of using GPT-5-mini as an LLM judge via the OpenAI API. While more extensive prompt engineering would be in principle possible, we use our results from the

prompt selection process to fixate an idea generation prompt that provides a meaningful baseline to compare finetuned models. Since the prompt variant of the baseline prompt with additional requests to refine its idea for novelty performs best, we choose this one for all subsequent steps.

## 6.3 Finetuning for Scientific Idea Generation

Here, we discuss the main results from the finetuning process and the evaluations of the resulting models. In section 6.3.1, we first argue that finetuning leads to an improvement in the LLM-rated idea quality. In section 6.3.2, we discuss how our finetuned models can be further improved by explicit enforcement of idea diversity in the objective function. Section 6.3.3 concludes that improvements in LLM-rated idea quality do not directly translate into higher success rates in agentic systems such as AI-Mandel and that a beneficial integration of finetuned models into such systems requires further investigation.

### 6.3.1 The Effect of Finetuning on Idea Quality

The results of the LLM judge ratings in section 5.3.1 clearly show that finetuning improves the quality of generated ideas as perceived by the LLM judge across all evaluated finetuning strategies. The bootstrapping uncertainty estimation and the permutation tests provide evidence for the statistical significance of the shift towards ideas that the LLM judge considers more novel, feasible, and scientifically interesting.

Rating the summarized one-sentence ideas shrinks the performance margin, which is expected since the shorter ideas contain much less information. It also changes the performance ordering of the models: SFT + DPO performs better on summaries than pure DPO whereas DPO is clearly the best performing model on full-length ideas. This shows that different finetuned models learn to use different text features. Some might learn to put idea quality into detailed explanations that get lost when summarizing, while others might specialize in densely expressed ideas that then work better in summaries.

The fact that all finetuned models still outperform the GPT-OSS-20B baseline indicates that the quality gain goes beyond a simple LLM judge bias towards longer ideas. These results show that our finetuning does not just result in longer ideas. Instead, it improves their substance in a way that remains clear even in a single sentence. To make this summary test more robust, future work could extend this by additionally using other LLMs for summarization as well as generating sets of summaries of different lengths.

As before, our evaluation heavily relies on LLM judges and would ideally be complemented with human expert opinions. However, idea quality in itself can be a rather subjective

metric that is evaluated differently by different systems or humans. Here, we rely fully on an LLM judge that might have specific biases. Besides the menial biases of output length or position, human experts might have other biases that lead to a different, but still subjective result. Experts in a specific field of quantum optics might prefer ideas in their field better and therefore rank these ideas higher. Deciding which biases are more important might not be reasonable: Idea quality improvement might be a purely subjective process that must be approached differently for each person or system.

In our finetuning methodology, we rely mainly on two major finetuning algorithms: SFT and DPO. In addition, our rating system for idea quality is a simple ordinal score that fails to capture the full depth of the concept of idea quality. Future work could explore other finetuning techniques such as proximal policy optimization [51], test other hyperparameter settings of the LoRA setup, and define more sophisticated rating mechanisms. The ordinal score could for example be extended into multiple scores for each metric such as novelty or scientific interest, and could be complemented with existing impact estimation methods [18].

### 6.3.2 The Effect of Finetuning on Idea Diversity

Besides idea quality, our discussion must also include idea diversity. The results for this type of evaluation are presented in section 5.3.2. The UMAP projections on the left side of Fig. 5.11 and table 5.1 indicate that the diversity of outputs is approximately preserved in the case of the SFT model, but decreases for SFT + DPO and DPO models. Especially the pure DPO finetuning seems to reduce the spread of the embeddings into small sub-regions of its initial distribution. This can be seen both qualitatively in the UMAP projections, as well as quantitatively in the reduction of centroid distance.

The right side of Fig. 5.11 suggests that finetuning does not automatically restrict idea diversity to the examples seen during finetuning. Even when the model is finetuned without any ideas from a cluster that the baseline model already knows, it can still generate ideas in that cluster afterward. This is evident from the zoomed cluster plots, where the cluster always holds both: blue points (ideas from the model finetuned without the cluster) and black points (ideas from the model finetuned with the cluster). This serves as a sanity check to verify that the finetuned model does not overfit to the ideas in the training set.

The observation of the general loss of diversity and the fact that the model can preserve clusters that are not part of the training set during finetuning complement each other. The cluster test suggests that the finetuned model retains a similar support over the idea space, since it can still access clusters that were excluded from finetuning. However, the global UMAP plots indicate that the distribution within this support changes as a result of finetuning. In other words, finetuning does not eliminate entire regions of the idea

space, but it does shift idea mass away from some clusters towards others, creating a denser concentration of ideas in selected regions that might have a higher probability of yielding high-quality ideas.

Taken together with the previous results from section 6.3.1, our finetuned models seem to improve idea quality, but often at the price of reducing idea diversity or picking certain regions of the idea space. This could be prevented by explicitly introducing diversity-preserving terms in the objective function as is often done using the baseline model as a reference policy from which the finetuned model should not diverge too much. To not only preserve, but also extend the idea diversity the idea dataset would need to be generated using a more powerful LLM from which the finetuned model can then learn to extend its own idea repertoire.

In our methodology, we measure idea diversity using semantic similarity. It is not clear if this similarity metric can sufficiently capture nuances in ideas from quantum optics and how well the semantic idea clusters correspond to distinct quantum optics experiments. To fully assess diversity, future work should therefore complement this embedding-based analysis by either human expert evaluation or by making use of more specialized embedding models that are trained on the specific domain at hand.

### 6.3.3 The Effect of Finetuning on AI-Mandel

Ideally, the improvement in idea quality from the finetuned models would directly translate into higher performance within the AI-Mandel agentic system. Higher performance corresponds to a higher ratio between generated and actually useful ideas. Section 5.3.3 shows the results of testing the finetuned models on this ratio with and without the idea pool that acts as a test for diversity. Without the idea pool, the finetuned SFT + DPO model achieves higher acceptance counts than the GPT-OSS-20B baseline model for both the novelty and judge agents. This is consistent with the earlier evaluation results, in which finetuning improved idea quality. However, the reported p-values from our significance tests in section 5.3.3 are all in two-digit percentage numbers and indicate that the observed acceptance rates are not rare under the null-hypothesis of acceptance rates that are independent of model count. With our current sample size the results are therefore not statistically significant. Higher sample counts would improve the tests, but do not yet justify their API call costs at this stage.

With the idea pool enabled, the baseline model achieves slightly higher acceptance counts than the finetuned model. This is in line with the earlier diversity evaluation, which suggested that the baseline model produces more diverse ideas across runs, whereas finetuning primarily improves the quality of individual proposals. In AI-Mandel, part of the value of the researcher agent lies in proposing sufficiently different ideas across runs.

Finetuning improves the quality of individual proposals, but if it reduces diversity, the overall system-level gain can vanish once novelty constraints are enforced.

Important next steps are therefore to integrate diversity metrics directly into the objective function to mitigate the negative side-effect of the finetuning. Our AI-Mandel integration also still lacks the feedback loops with which the researcher can refine its ideas. Extending the functionality of our finetuned models by training them to handle these feedback loops is one of the next steps towards a full-scale integration into AI-Mandel.

# 7 Conclusions and Outlook

Recent progress in the field of LLMs and agentic systems raises fundamental questions about the automation of science. While the number of so-called AI scientists steadily increases, most of these systems still receive their scientific objective from human researchers. Building true artificial scientists requires progress on the whole chain of scientific research, from idea generation to idea implementation and data analysis. Agentic systems like AI-Mandel can already perform these tasks, but rely on closed-source commercial LLMs that can not be finely controlled or improved for new tasks or tools. In addition, the ratio between generated ideas and useful ones remains low, leading to high cost and computing time.

This thesis investigates the use of finetuned open-source LLMs for idea generation in the context of AI-Mandel. Our work aims to compare finetuned open-source LLMs to their non-finetuned counterpart in terms of idea quality, diversity and their functionality within AI-Mandel.

By making use of different LLM judges, we find that GPT-OSS-20B outperforms other open-source LLMs on the task of scientific idea generation. After carefully engineering an appropriate idea generation prompt via LLM-based ranking of randomized pairwise comparisons, we find that different finetuning techniques indeed lead to a significant improvement in idea quality when rated by GPT-5-mini as an LLM judge. This quality improvement even holds when ranking the ideas based on summarized versions that are stripped of potential phrases or formatting that might trigger potential LLM judge biases. However, our diversity evaluation and AI-Mandel integration indicate clear issues with idea diversity for some of the tested finetuning techniques.

The next steps in making these results more reliable are the following:

- Extend the objective functions of the finetuning process with new terms that explicitly enforce idea diversity.
- Complement and verify LLM judges with human expert ratings and feedback and extend LLM evaluations with other LLM models, additional rating reasoning, and more sophisticated rating mechanisms.
- Explore other finetuning algorithms such as proximal policy optimization and

prompting techniques such as self-consistency prompting, in which multiple reasoning outputs are generated and compared for consistency.

Further progress on open-source LLMs as idea generators will contribute to advances in the automation of science by reducing reliance on closed-source models and opening possibilities for fine-grained control over model behavior.

# A Idea Generation Prompt for Model Selection

```
You are a visionary researcher in quantum optics.  You lead a
team of scientists and want to provide ideas for them.  Your team
consists of theoretical quantum optics researchers who are amazing
in taking your ideas and creating wonderful stand-alone proposals
for experiments.  The stand-alone proposals created by your team
members are often published in top-journals such as Phys.Rev.Lett.
(PRL). That requires that the idea is scientifically novel and
concrete proposals from your ideas should be interesting for
individual experts in the field or the field of quantum physics
researchers as a whole.

Your team is especially exceptionally good in executing your ideas
to fully detailed experimental proposals if your ideas are targeted
for the following domain:

Concrete quantum networks systems (e.g., generalizations of
entanglement swapping, quantum teleportation, etc) and foundational
quantum optics experiments.  Your ideas should be implementable
with probabilistic photon-pair sources (such as SPDC), or
probabilistic and deterministic single-photon sources, and standard
linear optics elements.  Your team cannot design experiments that
require dynamic feedback control.  If your idea is in that realm,
your team will figure out a great way to develop a full proposal.

Respond exactly with the following format:
Thought:  (the reasoning behind the idea)
Final idea:  (the actual idea if you are happy with it)
Do not add any other text.  Do not output multiple Thoughts and
Final ideas.
```

Box A.1: Idea-generation prompt variant 1.

```
You are a visionary leader in quantum optics, guiding a team of
theoretical researchers who transform your insights into fully
developed experimental proposals.  These proposals often become
publications in top journals such as Physical Review Letters,
requiring your ideas to be scientifically novel, concrete, and
compelling for experts in quantum optics and physics.

Your team excels at converting your concepts into detailed
experimental designs, particularly in the domains of:

Quantum network systems (e.g., extensions of entanglement swapping,
quantum teleportation, etc.)

Foundational quantum optics experiments

Constraints:

Implementable using probabilistic photon-pair sources (SPDC),
probabilistic and deterministic single-photon sources, and standard
linear optics elements.

No dynamic feedback control is possible.

Output format:
Thought:  (your reasoning behind the idea)
Final idea:  (the concrete idea you are satisfied with)

Do not add any additional text.  Do not provide multiple ideas.
```

Box A.2: Idea-generation prompt variant 2.

```
Generate a single, novel idea for a quantum optics experiment
that can be implemented using probabilistic photon-pair sources
(SPDC), probabilistic or deterministic single-photon sources,
and standard linear optics elements.  The idea should not rely
on dynamic feedback control.

The output should be in the following format:
```

```
Thought:  (explain the reasoning that leads to the idea)
Final idea:  (state the single clear idea you chose)
No additional commentary.  No lists of multiple ideas.
```

Box A.3: Idea-generation prompt variant 3.

# B Idea Ranker Prompt for Model Selection

```
You are an expert quantum physicist judging the novelty and
concreteness of research ideas in quantum optics.  This is the
prompt which generated output A and output B.

prompt start

PROMPT

prompt end

beginning of outputs

Output A:
a

Output B:
b

end of outputs

Now, compare the two outputs and decide which is better overall.
Respond with 'A' if Output A is better, or 'B' if Output B is
better.
Do not add any other text.
```

Box B.1: Idea-ranking prompt variant 1.

```
You are a quantum physics expert evaluating the originality
and clarity of research ideas in quantum optics.  This is the
```

```
instruction that produced output A and output B.

prompt start

PROMPT

prompt end

beginning of outputs

Output A:
a

Output B:
b

end of outputs

Choose which output is overall better.  Reply with 'A' if Output A
is better, or 'B' if Output B is better.  Give no additional text.
```

Box B.2: Idea-ranking prompt variant 2.

```
Compare the following two research ideas in quantum optics,
generated from the same prompt.  Evaluate their novelty and
concreteness.

prompt start

PROMPT

prompt end

beginning of outputs

Output A:
a

Output B:
```

```
b

end of outputs

Respond with 'A' if Output A is better, or 'B' if Output B is
better.  Do not add any other text.
```

Box B.3: Idea-ranking prompt variant 3.

# C Prompt Selection Prompt

```
You are a visionary researcher in quantum optics.  You lead a
team of scientists and want to provide ideas for them.  Your team
consists of theoretical quantum optics researchers who are amazing
in taking your ideas and creating wonderful stand-alone proposals
for experiments.  The stand-alone proposals created by your team
members are often published in top-journals such as Phys.Rev.Lett.
(PRL). That requires that the idea is scientifically novel and
concrete proposals from your ideas should be interesting for
individual experts in the field or the field of quantum physics
researchers as a whole.

Your team is especially exceptionally good in executing your ideas
to fully detailed experimental proposals if your ideas are targeted
for the following domain:

Concrete quantum networks systems (e.g., generalizations of
entanglement swapping, quantum teleportation, etc) and foundational
quantum optics experiments.  Your ideas should be implementable
with probabilistic photon-pair sources (such as SPDC), or
probabilistic and deterministic single-photon sources, and standard
linear optics elements.  Your team cannot design experiments that
require dynamic feedback control.

If your idea is in that realm as described above, your team will
figure out a great way to develop a full proposal.

Here are some paper titles as initial inspiration.  You can decide
for yourself which ones are actually interesting to you.

START OF PAPER TITLES FOR INSPIRATION
```

```
Paper 1:  Toward Neural Network Simulation of Variational Quantum
Algorithms
-----------------------------------------------------
Paper 2:  Real-world data encryption with continuous-variable
measurement device-independent quantum key distribution
-----------------------------------------------------
Paper 3:  A novel hybrid protocol for semiquantum key distribution
and semiquantum secret sharing
-----------------------------------------------------

END OF PAPER TITLES FOR INSPIRATION

Respond exactly with the following format:
Thought:  (the reasoning behind the idea)
Final idea:  (the actual idea if you are happy with it)
Do not add any other text.  Do not output multiple Thoughts and
Final ideas.
```

Box C.1: Example of an idea generation prompt with random paper titles for prompt selection.

```
You are a visionary researcher in quantum optics.  You lead a
team of scientists and want to provide ideas for them.  Your team
consists of theoretical quantum optics researchers who are amazing
in taking your ideas and creating wonderful stand-alone proposals
for experiments.  The stand-alone proposals created by your team
members are often published in top-journals such as Phys.Rev.Lett.
(PRL). That requires that the idea is scientifically novel and
concrete proposals from your ideas should be interesting for
individual experts in the field or the field of quantum physics
researchers as a whole.

Your team is especially exceptionally good in executing your ideas
to fully detailed experimental proposals if your ideas are targeted
for the following domain:

Concrete quantum networks systems (e.g., generalizations of
entanglement swapping, quantum teleportation, etc) and foundational
quantum optics experiments.  Your ideas should be implementable
with probabilistic photon-pair sources (such as SPDC), or
probabilistic and deterministic single-photon sources, and standard
```

```
linear optics elements.  Your team cannot design experiments that
require dynamic feedback control.  If your idea is in that realm,
your team will figure out a great way to develop a full proposal.

Here is one last, but absolutely necessary condition:
Your proposed research idea should combine these two scientific
concepts in a meaningful way.
"topological photonic state" and "non hermitian skin effect"

Try your best to combine these concepts even if it means suggesting
experiments that are very different from the examples you have
seen.  The main goal is to suggest something interesting.

Respond exactly with the following format:
Thought:  (the reasoning behind the idea)
Final idea:  (the actual idea if you are happy with it)
Do not add any other text.  Do not output multiple Thoughts and
Final ideas.
```

Box C.2: Example of an idea generation prompt with random topic combinations for prompt selection.

```
Your colleague (the 'Expert') has access to a tool (pytheus) that
can design quantum optics experiments based on a clear target.

Here is some information on the capabilities and limitations of
pytheus:
1) Experiment Generation:  Pytheus can generate experimental
setups for concrete quantum networks (e.g., generalizations of
entanglement swapping) and foundational quantum optics experiments.
2) Abstract State Encoding:  Pytheus encodes quantum states
abstractly.  For example, the state |0>+|1> has no inherent
physical implementation--it could correspond to photon polarization
(a discrete 2-dimensional system) or orbital angular momentum
(a discrete high-dimensional system).  However, this physical
encoding is not part of Pytheus's output and must be specified
by the experimenter afterward.  Consequently, suggestions involving
hyper-entanglement or hybrid-entanglement cannot be meaningfully
interpreted by Pytheus, as it operates purely on abstract quantum
```

```
state descriptions without reference to their physical realization.
3) Limitations in Feedback Control:  Pytheus cannot design
experiments that require dynamic feedback control, including many
quantum error-correction systems.

Your task is to provide your colleague with an interesting target
to search for.  Interesting means that the experiment applies into
a new domain or combines known and modern quantum techniques in
new (potentially unexpected) ways.  But make sure it is suitable
for pytheus.  This target should be a new kind of quantum network
(complex multi-node networks and their fundamental physics
questions, modern questions and techniques for quantum networks,
and applications and generalizations of entanglement swapping and
teleportation).  Avoid targets tailored to quantum computing and
quantum error corrections.

Ideally this would be an idea for an experiment that can be
generalized and expanded upon beyond a single experimental
setup towards a whole range of diverse but related experiments.
Crucially (!):  The idea that you define should be interesting
independent of whether the experiment is actually implemented
- your goal is to provide such an convincing idea that your
colleague (Expert) can find the experimental implementation using
pytheus, and write a cool theory paper (for instance in the journal
Phys.Rev.Lett.).

Respond exactly with the following format:
Thought:  (the reasoning behind the idea)
Final idea:  (the actual idea if you are happy with it)
Do not add any other text.  Do not output multiple Thoughts and
Final ideas.
```

Box C.3: Idea generation prompt with PyTheus introduction for prompt selection.

```
Your colleague (the 'Expert') has access to a tool (pytheus) that
can design quantum optics experiments based on a clear target.

Here is some information on the capabilities and limitations of
pytheus:
```

```
1) Experiment Generation:  Pytheus can generate experimental
setups for concrete quantum networks (e.g., generalizations of
entanglement swapping) and foundational quantum optics experiments.
2) Abstract State Encoding:  Pytheus encodes quantum states
abstractly.  For example, the state |0>+|1> has no inherent
physical implementation--it could correspond to photon polarization
(a discrete 2-dimensional system) or orbital angular momentum
(a discrete high-dimensional system).  However, this physical
encoding is not part of Pytheus's output and must be specified
by the experimenter afterward.  Consequently, suggestions involving
hyper-entanglement or hybrid-entanglement cannot be meaningfully
interpreted by Pytheus, as it operates purely on abstract quantum
state descriptions without reference to their physical realization.
3) Limitations in Feedback Control:  Pytheus cannot design
experiments that require dynamic feedback control, including many
quantum error-correction systems.

Pytheus accepts JSON configurations that specify the desired target
state and various experimental constraints.

Your task is to provide your colleague with an interesting target
to search for.  Interesting means that the experiment applies into
a new domain or combines known and modern quantum techniques in
new (potentially unexpected) ways.  But make sure it is suitable
for pytheus.  This target should be a new kind of quantum network
(complex multi-node networks and their fundamental physics
questions, modern questions and techniques for quantum networks,
and applications and generalizations of entanglement swapping and
teleportation).  Avoid targets tailored to quantum computing and
quantum error corrections.

Ideally this would be an idea for an experiment that can be
generalized and expanded upon beyond a single experimental
setup towards a whole range of diverse but related experiments.
Crucially (!):  The idea that you define should be interesting
independent of whether the experiment is actually implemented
- your goal is to provide such an convincing idea that your
colleague (Expert) can find the experimental implementation using
pytheus, and write a cool theory paper (for instance in the journal
```

Phys.Rev.Lett.).

Here are some previously explored Pytheus examples plus configurations. Use these examples as inspirations.

START OF IDEAS AND CONFIGURATIONS

EXAMPLE 1
$ES3d_sp$
explanation: "In this example we create entanglement between two qutrits with single photon sources. We want the photons entangled in a three dimensional Bell state.

$Psi = Psi_BD = |00> + |11> + |22>$ (without normalization)

The setup we are optimizing for here is one where photons are emitted by single photon emitters (identifiable 'single_emitters').

In this case, six photons are emitted and the setup (which we optimize) manipulates them before they enter the detectors.

We can visualize the setup in the following way (with the photons going from top to bottom):

emitters: [2][3][4][5][6][7]
setup: [result of optimization]
detectors: [0][1][anc. det.][anc. det.][anc. det.][anc. det.]

where [0] and [1] are the detectors where the Bell state should be created. [anc. det.] are ancillary detectors, which should be measured by a third party.

The setup to create a 3d Bell state setup would normally be easier to realize, but we want to create this state in a way that is inspired by entanglement swapping. This means, that [0] should be measured by Alice, [1] should be measured by Bob and the ancillary detectors should be measured by Charlie.

There should be no emitter that can send a photon to both Alice and

Bob. We choose that there are three emitters that can send photons to Bob and Charlie ([2],[3],[4]) and three emitters that can send photons to Alice and Charlie ([5],[6],[7]).

In the context of single photon emitters, the keyword 'removed_connections' excludes paths between the specified pairs of source and detector. We thus include all of these forbidden paths.

This would not be possible without ancillary particles (there number is given by 'num_anc'). The ancillary particles are measured by a third party.

The standard entanglement swapping experiment uses probabilistic photon pair sources whereas this uses single photon sources. 'config': 'description': 'Entanglement swapping between two qutrits with single photon sources', 'foldername': '$ES3d_sp$', 'bulk_thr': 0.0, 'edges_tried': 30, 'ftol': 1e-09, 'loss_func': 'cr', 'num_anc': 10, 'num_pre': 1, 'optimizer': 'L-BFGS-B', 'imaginary': False, 'safe_hist': True, 'samples': 10, 'target_state': ['00', '11', '22'], 'in_nodes': [], 'out_nodes': [], 'thresholds': [0.3, 0.1], 'heralding_out': None, 'single_emitters': [2, 3, 4, 5, 6, 7], 'amplitudes': [], 'tries_per_edge': 5, 'removed_connections': [[0, 2], [0, 3], [0, 4], [1, 5], [1, 6], [1, 7]], 'seed': None, 'unicolor': False, 'number_resolving': True, 'novac': None, 'loops': None, 'topopt': None, 'dimensions': [], 'brutal_covers': None, 'verts': [], 'anc_detectors': []
.....
END OF IDEAS AND CONFIGURATIONS

Generate only textual ideas and no configurations.

Respond exactly with the following format:
Thought: (the reasoning behind the idea)
Final idea: (the actual idea if you are happy with it)
Do not add any other text. Do not output multiple Thoughts and Final ideas.

Box C.4: Example of an idea generation prompt with random PyTheus configuration example for prompt selection.

```
You are a visionary researcher in quantum optics.  You lead a
team of scientists and want to provide ideas for them.  Your team
consists of theoretical quantum optics researchers who are amazing
in taking your ideas and creating wonderful stand-alone proposals
for experiments.  The stand-alone proposals created by your team
members are often published in top-journals such as Phys.Rev.Lett.
(PRL). That requires that the idea is scientifically novel and
concrete proposals from your ideas should be interesting for
individual experts in the field or the field of quantum physics
researchers as a whole.

Your team is especially exceptionally good in executing your ideas
to fully detailed experimental proposals if your ideas are targeted
for the following domain:
Concrete quantum networks systems (e.g., generalizations of
entanglement swapping, quantum teleportation, etc) and foundational
quantum optics experiments.  Your ideas should be implementable
with probabilistic photon-pair sources (such as SPDC), or
probabilistic and deterministic single-photon sources, and standard
linear optics elements.  Your team cannot design experiments that
require dynamic feedback control.  If your idea is in that realm,
your team will figure out a great way to develop a full proposal.
Here are some ideas that the team has already designed quantum
optics experiments for.  A new idea should also not be too similar
to the ideas from this list.

START OF LIST OF EXPLORED EXPERIMENTS

1.  **Entangling Two Photons That Never Interacted with Bell
Pairs**:  - Demonstrates the entanglement of two photons that
have never interacted directly, using Bell pairs.  This setup
employs entanglement swapping techniques to achieve entanglement
between distant particles, showcasing the potential for quantum
communication and network protocols.

2.  **Entanglement Swapping of Three Bell Pairs (Four Additional
Particles)**:  - This experiment demonstrates entanglement swapping
```

between three Bell pairs, requiring four additional particles. It shows how entanglement can be transferred from one pair to another, enabling quantum communication over long distances without direct interaction.

3. **Entanglement Swapping for Two Pairs of Qutrits (Six Additional Particles)**: - Produces entanglement swapping between two pairs of qutrits using six additional particles. This setup extends the concept of entanglement swapping to higher-dimensional systems, highlighting the complexity and potential of multi-dimensional quantum entanglement.

4. **Entanglement Swapping with Single-Photon Sources**: - Demonstrates entanglement swapping using single-photon sources. The experiment creates a three-dimensional Bell state, showing the feasibility of using deterministic single-photon sources for complex quantum communication tasks.
5. **Yeo-Chua Analyzer**: - This analyzer is used for the Yeo-Chua state, a quantum state used in certain teleportation protocols. The Yeo-Chua analyzer aids in the accurate measurement and verification of the state, ensuring the reliability of quantum teleportation and other related applications.

6. **Mean King's Problem Analyzer (Three-Dimensional Case)**: - Similar to the previous setup, this experiment addresses the Mean King's Problem in three dimensions. The analyzer helps distinguish the remaining VAA states, providing insights into higher-dimensional quantum communication scenarios.

7. **Heralded CNOT Gate (3,3)**: - This experiment showcases a heralded CNOT gate with three control qubits and three target qubits. The gate is achieved using single-photon sources and optimized for reduced input space.

8. **Post-Selected Toffoli Gate (3,3)**: - A post-selected Toffoli gate with three control and three target qubits. Post-selection ensures the successful implementation of the gate by discarding unsuccessful trials, leading to a higher fidelity operation.

```
END OF LIST OF EXPLORED EXPERIMENTS

Respond exactly with the following format:
Thought:  (the reasoning behind the idea)
Final idea:  (the actual idea if you are happy with it)
Do not add any other text.  Do not output multiple Thoughts and
Final ideas.
```

Box C.5: Example of an idea generation prompt with random idea examples for prompt selection.

```
You are a visionary researcher in quantum optics.  You lead a
team of scientists and want to provide ideas for them.  Your team
consists of theoretical quantum optics researchers who are amazing
in taking your ideas and creating wonderful stand-alone proposals
for experiments.  The stand-alone proposals created by your team
members are often published in top-journals such as Phys.Rev.Lett.
(PRL). That requires that the idea is scientifically novel and
concrete proposals from your ideas should be interesting for
individual experts in the field or the field of quantum physics
researchers as a whole.

Your team is especially exceptionally good in executing your ideas
to fully detailed experimental proposals if your ideas are targeted
for the following domain:

Concrete quantum networks systems (e.g., generalizations of
entanglement swapping, quantum teleportation, etc) and foundational
quantum optics experiments.  Your ideas should be implementable
with probabilistic photon-pair sources (such as SPDC), or
probabilistic and deterministic single-photon sources, and standard
linear optics elements.  Your team cannot design experiments that
require dynamic feedback control.  If your idea is in that realm,
your team will figure out a great way to develop a full proposal.

Be constructively critical of your own idea and check if it is
novel.  Refine it and aim for producing a truly novel idea.

Respond exactly with the following format:
Thought:  (the reasoning behind the idea)
```

```
Final idea:  (the actual idea if you are happy with it)
Do not add any other text.  Do not output multiple Thoughts and
Final ideas.
```

Box C.6: Idea generation prompt with self-refinement for prompt selection.

# D Idea Ranker Prompt for Dataset Creation and Evaluation

```
You are an expert quantum physicist judging the research idea in
quantum optics based on the criteria given below:

1.Scientific novelty

2.Feasibility with Pytheus - a discovery tool for quantum optics
experiments.  Pytheus capabilities include designing concrete
quantum networks systems (e.g., generalizations of entanglement
swapping, quantum teleportation, etc.)  and foundational quantum
optics experiments.  The ideas should be implementable with
probabilistic photon-pair sources (such as SPDCs), probabilistic
and deterministic single-photon sources, and standard linear optics
elements.  Pytheus cannot design experiments that require dynamic
feedback control.

3.Scientifically interesting

Here is the research idea:

BEGINNING OF RESEARCH IDEA

IDEA

END OF RESEARCH IDEA

Please respond with a single number between 1 and 3 indicating the
overall score of the research idea based on the criteria above,
where 1 means "poor", 2 means "moderate" and 3 means "good".  Do
```

```
not include any additional text or explanation, just provide the
number.
```

Box D.1: Ranking prompt for idea dataset generation.

# E Example Idea

**Hybrid High-Dimensional Entanglement Swapping between Polarization and Orbital Angular Momentum for Quantum Networks**

**Overview:**
We propose an experiment that performs entanglement swapping on photon pairs that are simultaneously entangled in both polarization and orbital angular momentum (OAM). Two independent SPDC sources generate Bell states of the form

$$|\Phi^+\rangle_{12} = \frac{1}{2}\big(|H,+1\rangle_1|H,-1\rangle_2+|H,-1\rangle_1|H,+1\rangle_2+|V,+1\rangle_1|V,-1\rangle_2+|V,-1\rangle_1|V,+1\rangle_2\big).$$

Photons 2 and 3 are directed to a $4 \times 4$ linear interferometer that implements a joint Bell-state measurement (BSM) on the combined polarization-OAM space. This is achieved by inserting a q-plate on each path to couple polarization and OAM, followed by a log-polar mode converter that maps OAM modes to distinct spatial modes. The $4\times4$ interferometer is built from a tritter ($3\times3$) and an additional beam-splitter network, allowing projection onto a subset of the 16 joint Bell states. Successful detection of a specific coincidence pattern heralds the BSM, collapsing photons 1 and 4 into a hybrid Bell state of the form

$$|\Phi^+\rangle_{14} = \frac{1}{\sqrt{2}}\big(|H,+1\rangle_1|H,-1\rangle_4 + |V,-1\rangle_1|V,+1\rangle_4\big).$$

**Key novelties:**
1. *Hybrid, high-dimensional entanglement:* The swapped state is a 4-dimensional Bell state spanning polarization and OAM, which is experimentally unprecedented.
2. *Increased channel capacity for quantum networks:* Each photon carries two qubits, enabling high-dimensional quantum key distribution or multiplexed quantum communication between network nodes.

3. *Linear-optics BSM with higher success probability:* The $4 \times 4$ interferometer implements a partial BSM that can reach a success probability of $25\%$, while still being realizable with static optics.
4. *Verification via hybrid CHSH tests:* By measuring joint observables in the hybrid basis, for example $\sigma_x \otimes L_z$, the entanglement can be quantified through a CHSH inequality that includes both degrees of freedom.

**Experimental feasibility:**
- *Sources:* Two type-II PPKTP crystals pumped by $532\,\mathrm{nm}$ pulsed lasers generate entangled pairs with bandwidth exceeding $10\,\mathrm{THz}$, ensuring sufficient spectral overlap for the BSM.
- *Optics:* Q-plates ($\Delta = \pi$) convert polarization and OAM with high efficiency; log-polar converters map $\ell = \pm 1$ to distinct spatial modes with fidelity above $90\%$.
- *Interferometer:* A tritter built from $3 \times 3$ beam splitters, followed by a $2 \times 2$ beam-splitter network, realizes the $4 \times 4$ unitary. Phase stability is achieved with common-path designs; no dynamic feedback is needed.
- *Detectors:* Superconducting nanowire single-photon detectors (SNSPDs) with $80\%$ efficiency and $< 50\,\mathrm{ps}$ jitter resolve the time bins and spatial modes.
- *Post-selection:* Coincidence logic selects the desired BSM pattern; accidental coincidences are suppressed by temporal filtering.

**Impact:**
This experiment would be the first demonstration of high-dimensional hybrid entanglement swapping, opening a new toolbox for quantum network nodes that can exploit multi-qubit entanglement per photon. The resulting hybrid Bell state can be directly integrated into quantum key distribution protocols or used as a resource for high-dimensional quantum teleportation across network nodes, thereby increasing channel capacity and resilience to decoherence. The proposal is fully concrete, theoretically grounded, and within the experimental capabilities of the team, making it a strong candidate for a PRL submission.

Box E.1: Example of an idea generated by the SFT + DPO model.

# F Significance Calculation for Prompt Selection

To test whether the best performing prompt in Fig. 5.6 is statistically significantly stronger than the second-best, we first calculate the log-scale of the Bradley-Terry strengths using $\beta_i = \log p_i$. We then compute the difference $d$ in $\beta$ values for the top two prompts $a$ and $b$:

$$d = \beta_{\mathrm{a}} - \beta_{\mathrm{b}} = 0.2. \tag{F.1}$$

For a model parameter difference from a logistic pairwise-comparison, the standard error is based on the covariance matrix of the model:

$$SE(d) = \sqrt{\mathrm{Var}(\beta_a) + \mathrm{Var}(\beta_b) - 2\,\mathrm{Cov}(\beta_a, \beta_b)} = 0.065. \tag{F.2}$$

The z-score can be computed as:

$$z = \frac{d}{SE(d)} = \frac{0.200298}{0.064710} \approx 3.10. \tag{F.3}$$

Under the null hypothesis that both prompts are equally strong, this z-value is approximately standard normal. The corresponding one-sided probability for this performance gap between the two prompts can therefore be computed with $\Phi$, the cumulative distribution function of the standard normal distribution:

$$p = 1 - \Phi(3.10) \approx 0.001. \tag{F.4}$$

# G Significance Calculation for AI-Mandel Integration

To test the significance of our AI-Mandel integration acceptance counts, we use the Fisher's exact test [88]. For each comparison, we construct a $2 \times 2$ table of models versus outcomes:

| | accepted | not accepted | row total |
|---|---|---|---|
| SFT+DPO | $a$ | $b$ | $a+b$ |
| Baseline | $c$ | $d$ | $c+d$ |
| | $a+c$ | $b+d$ | $n$ |

(G.1)

Conditioning on the row and column totals, the number of accepted ideas assigned to the SFT + DPO model is a hypergeometric random variable:

$$X \sim \text{Hypergeometric}(N = n,\ K = a + c,\ m = a + b). \tag{G.2}$$

Under the null hypothesis of equal acceptance probabilities, the probability of observing a table with value $X = x$ is

$$\Pr(X = x) = \frac{\binom{a+c}{x}\binom{b+d}{(a+b)-x}}{\binom{n}{a+b}}. \tag{G.3}$$

The probability of the observed table is therefore

$$\Pr(X = a) = \frac{\binom{a+c}{a}\binom{b+d}{b}}{\binom{n}{a+b}}. \tag{G.4}$$

The two-sided Fisher exact test p-value is obtained by summing the probabilities of all tables with the same margins whose probability is less than or equal to that of the observed table:

$$p = \sum_{x:\, \Pr(X=x) \le \Pr(X=a)} \Pr(X = x). \tag{G.5}$$

# Bibliography


[1] D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes, Autonomous chemical research with large language models, Nature **624**, 570–578 (2023).

[2] A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White, and P. Schwaller, Augmenting large language models with chemistry tools, Nature machine intelligence **6**, 525–535 (2024).

[3] Y. Zou, A. H. Cheng, A. Aldossary, J. Bai, S. X. Leong, J. A. Campos-Gonzalez-Angulo, C. Choi, C. T. Ser, G. Tom, A. Wang, *et al.*, El agente: An autonomous agent for quantum chemistry, Matter **8** (2025).

[4] H. Wang, M. Skreta, C.-T. Ser, W. Gao, L. Kong, F. Strieth-Kalthoff, C. Duan, Y. Zhuang, Y. Yu, Y. Zhu, *et al.*, Efficient evolutionary search over chemical space with large language models, arXiv preprint arXiv:2406.16976 (2024).

[5] A. Ghafarollahi and M. J. Buehler, Sciagents: automating scientific discovery through bioinspired multi-agent intelligent graph reasoning, Advanced Materials **37**, 2413523 (2025).

[6] M. Nägele and F. Marquardt, Agentic exploration of physics models, arXiv preprint arXiv:2509.24978 (2025).

[7] S. Cao, Z. Zhang, M. Alghadeer, S. D. Fasciati, M. Piscitelli, M. Bakr, P. Leek, and A. Aspuru-Guzik, Automating quantum computing laboratory experiments with an agent-based ai framework, Patterns **6** (2025).

[8] C. Yu, V. Uotila, S. Deng, Q. Wu, T. Shi, S. Jiang, L. You, and B. Zhao, Quasar: Quantum assembly code generation using tool-augmented llms via agentic rl, arXiv preprint arXiv:2510.00967 (2025).

[9] A. Sharma, Y. Fu, V. Ansari, R. Iyer, F. Kuang, K. Mistry, R. I. Aishy, S. Ahmad, J. Matres, D. R. Englund, *et al.*, Ai agents for photonic integrated circuit design automation, APL Machine Learning **3** (2025).

[10] D. Lu, J. M. Malof, and W. J. Padilla, An agentic framework for autonomous metamaterial modeling and inverse design, ACS Photonics **12**, 6071–6080 (2025).

[11] B. Georgiev, J. Gómez-Serrano, T. Tao, and A. Z. Wagner, Mathematical exploration and discovery at scale, 2025, URL https://arxiv. org/abs/2511.02864 .

[12] A. Novikov, N. Vũ, M. Eisenberger, E. Dupont, P.-S. Huang, A. Z. Wagner, S. Shirobokov, B. Kozlovskii, F. J. Ruiz, A. Mehrabian, *et al.*, Alphaevolve: A coding agent for scientific and algorithmic discovery, arXiv preprint arXiv:2506.13131 (2025).

[13] S. Arlt, X. Gu, and M. Krenn, Towards autonomous quantum physics research using llm agents with access to intelligent tools, arXiv preprint arXiv:2511.11752 (2025).

[14] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha, The ai scientist: Towards fully automated open-ended scientific discovery, arXiv preprint arXiv:2408.06292 (2024).

[15] F. Villaescusa-Navarro, B. Bolliet, P. Villanueva-Domingo, A. E. Bayer, A. Acquah, C. Amancharla, A. Barzilay-Siegal, P. Bermejo, C. Bilodeau, P. C. Ramírez, *et al.*, The denario project: Deep knowledge ai agents for scientific discovery, arXiv preprint arXiv:2510.26887 (2025).

[16] Y. Yamada, R. T. Lange, C. Lu, S. Hu, C. Lu, J. Foerster, J. Clune, and D. Ha, The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search, arXiv preprint arXiv:2504.08066 (2025).

[17] S. Schmidgall, Y. Su, Z. Wang, X. Sun, J. Wu, X. Yu, J. Liu, M. Moor, Z. Liu, and E. Barsoum, Agent laboratory: Using llm agents as research assistants, Findings of the Association for Computational Linguistics: EMNLP 2025 , 5977–6043 (2025).

[18] X. Gu and M. Krenn, Forecasting high-impact research topics via machine learning on evolving knowledge graphs, Machine Learning: Science and Technology **6**, 025041 (2025).

[19] C. Ruiz-Gonzalez, S. Arlt, J. Petermann, S. Sayyad, T. Jaouni, E. Karimi, N. Tischler, X. Gu, and M. Krenn, Digital discovery of 100 diverse quantum experiments with pytheus, Quantum **7**, 1204 (2023).

[20] S. Arlt, M. Krenn, and X. Gu, Automated discovery of high-dimensional multipartite entanglement with photons that never interacted, arXiv preprint arXiv:2510.10707 (2025).

[21] S. Arlt, M. Krenn, and X. Gu, Automated discovery of non-local photonic gates, arXiv preprint arXiv:2511.04648 (2025).

[22] J. Baek, S. K. Jauhar, S. Cucerzan, and S. J. Hwang, in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (2025) pp. 6709–6738.

[23] X. Hu, H. Fu, J. Wang, Y. Wang, Z. Li, R. Xu, Y. Lu, Y. Jin, L. Pan, and Z. Lan, Nova: An iterative planning and search approach to enhance novelty and diversity of llm generated ideas, arXiv preprint arXiv:2410.14255 (2024).

[24] A. Rzhetsky, J. G. Foster, I. T. Foster, and J. A. Evans, Choosing experiments to accelerate collective discovery, Proceedings of the National Academy of Sciences **112**, 14569–14574 (2015).

[25] M. Krenn and A. Zeilinger, Predicting research trends with semantic and neural networks with an application in quantum physics, Proceedings of the National Academy of Sciences **117**, 1910–1916 (2020).

[26] M. Krenn, L. Buffoni, B. Coutinho, S. Eppel, J. G. Foster, A. Gritsevskiy, H. Lee, Y. Lu, J. P. Moutinho, N. Sanjabi, *et al.*, Forecasting the future of artificial intelligence with machine learning-based link prediction in an exponentially growing knowledge network, Nature Machine Intelligence **5**, 1326–1335 (2023).

[27] T. Marwitz, A. Colsmann, B. Breitung, C. Brabec, C. Kirchlechner, E. Blasco, G. C. Marques, H. Hahn, M. Hirtz, P. A. Levkin, *et al.*, Predicting new research directions in materials science using large language models and concept graphs, arXiv preprint arXiv:2506.16824 (2025).

[28] F. Shi and J. Evans, Surprising combinations of research contents and contexts are related to impact and emerge with scientific outsiders from distant disciplines, Nature Communications **14**, 1641 (2023).

[29] Q. Wang, D. Downey, H. Ji, and T. Hope, in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2024) pp. 279–299.

[30] Z. Yang, X. Du, J. Li, J. Zheng, S. Poria, and E. Cambria, in *Findings of the Association for Computational Linguistics: ACL 2024* (2024) pp. 13545–13565.

[31] C. Si, D. Yang, and T. Hashimoto, Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers, arXiv preprint arXiv:2409.04109 (2024).

[32] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, Training language models to follow instructions

with human feedback, Advances in neural information processing systems **35**, 27730–27744 (2022).

[33] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. Christiano, Recursively summarizing books with human feedback, arXiv preprint arXiv:2109.10862 (2021).

[34] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, *et al.*, Constitutional ai: Harmlessness from ai feedback, arXiv preprint arXiv:2212.08073 (2022).

[35] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, Direct preference optimization: Your language model is secretly a reward model, Advances in neural information processing systems **36**, 53728–53741 (2023).

[36] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, Deep reinforcement learning from human preferences, Advances in neural information processing systems **30** (2017).

[37] H. Wang, L. Wang, C. Zhang, T. Mao, S. Qin, Q. Lin, S. Rajmohan, and D. Zhang, Text2grad: Reinforcement learning from natural language feedback, arXiv preprint arXiv:2505.22338 (2025).

[38] S. A. Lloret, S. Dhuliawala, K. Murugesan, and M. Sachan, in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (2024) pp. 20240–20266.

[39] X. Wang, H. Peng, R. Jabbarvand, and H. Ji, in *Findings of the Association for Computational Linguistics: NAACL 2024* (2024) pp. 223–239.

[40] Z. Wu, Y. Hu, W. Shi, N. Dziri, A. Suhr, P. Ammanabrolu, N. A. Smith, M. Ostendorf, and H. Hajishirzi, Fine-grained human feedback gives better rewards for language model training, Advances in Neural Information Processing Systems **36**, 59008–59033 (2023).

[41] S. Liu, Y. Pan, G. Chen, and X. Li, Reward modeling with ordinal feedback: Wisdom of the crowd, arXiv preprint arXiv:2411.12843 (2024).

[42] E. Koroleva and S. Mikhailova, Llm fine-tuning with distributional feedback: An approach for aligning with human preferences, (2025).

[43] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, *et al.*, Judging llm-as-a-judge with mt-bench and chatbot arena, Advances in neural information processing systems **36**, 46595–46623 (2023).

[44] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, *et al.*, A survey on llm-as-a-judge, The Innovation (2024).

[45] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, L. Kong, Q. Liu, T. Liu, *et al.*, in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2024) pp. 9440–9450.

[46] J. Ye, Y. Wang, Y. Huang, D. Chen, Q. Zhang, N. Moniz, T. Gao, W. Geyer, C. Huang, P.-Y. Chen, *et al.*, Justice or prejudice? quantifying biases in llm-as-a-judge, arXiv preprint arXiv:2410.02736 (2024).

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, Advances in neural information processing systems **30** (2017).

[48] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, *et al.*, The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).

[49] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, *et al.*, Gemma 3 technical report, arXiv preprint arXiv:2503.19786 (2025).

[50] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao, *et al.*, gpt-oss-120b & gpt-oss-20b model card, arXiv preprint arXiv:2508.10925 (2025).

[51] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[52] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, Language models are unsupervised multitask learners, OpenAI blog **1**, 9 (2019).

[53] A. Liu, A. Mei, B. Lin, B. Xue, B. Wang, B. Xu, B. Wu, B. Zhang, C. Lin, C. Dong, *et al.*, Deepseek-v3. 2: Pushing the frontier of open large language models, arXiv preprint arXiv:2512.02556 (2025).

[54] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff, *et al.*, The prompt report: A systematic survey of prompt engineering techniques, arXiv preprint arXiv:2406.06608 (2024).

[55] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, A systematic survey of prompt engineering in large language models: Techniques and applications, arXiv preprint arXiv:2402.07927 **1** (2024).

[56] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, Finetuned language models are zero-shot learners, arXiv preprint arXiv:2109.01652 (2021).

[57] S. Roy, R. Shu, N. Pappas, E. Mansimov, Y. Zhang, S. Mansour, and D. Roth, in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)* (2023) pp. 119–143.

[58] R. Tutunov, A. Grosnit, J. Ziomek, J. Wang, and H. Bou-Ammar, Why can large language models generate correct chain-of-thoughts?, arXiv preprint arXiv:2310.13571 (2023).

[59] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, *et al.*, in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38 (2024) pp. 17682–17690.

[60] W. Chen, X. Ma, X. Wang, and W. W. Cohen, Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, arXiv preprint arXiv:2211.12588 (2022).

[61] J. Huang, S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han, in *Proceedings of the 2023 conference on empirical methods in natural language processing* (2023) pp. 1051–1068.

[62] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, *et al.*, Self-refine: Iterative refinement with self-feedback, Advances in neural information processing systems **36**, 46534–46594 (2023).

[63] N. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, J. Yang, *et al.*, in *Findings of the Association for Computational Linguistics: ACL 2024* (2024) pp. 14743–14777.

[64] M. Zheng, J. Pei, L. Logeswaran, M. Lee, and D. Jurgens, in *Findings of the Association for Computational Linguistics: EMNLP 2024* (2024) pp. 15126–15154.

[65] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, Lora: Low-rank adaptation of large language models., Iclr **1**, 3 (2022).

[66] B. D. Rouhani, R. Zhao, A. More, M. Hall, A. Khodamoradi, S. Deng, D. Choudhary, M. Cornea, E. Dellinger, K. Denolf, *et al.*, Microscaling data formats for deep learning, arXiv preprint arXiv:2310.10537 (2023).

[67] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li, *et al.*, Satellite-to-ground quantum key distribution, Nature **549**, 43–47 (2017).

[68] S.-K. Liao, W.-Q. Cai, J. Handsteiner, B. Liu, J. Yin, L. Zhang, D. Rauch, M. Fink, J.-G. Ren, W.-Y. Liu, *et al.*, Satellite-relayed intercontinental quantum network, Physical review letters **120**, 030501 (2018).

[69] S. Bartolucci, P. Birchall, H. Bombin, H. Cable, C. Dawson, M. Gimeno-Segovia, E. Johnston, K. Kieling, N. Nickerson, M. Pant, *et al.*, Fusion-based quantum computation, Nature Communications **14**, 912 (2023).

[70] E. Polino, M. Valeri, N. Spagnolo, and F. Sciarrino, Photonic quantum metrology, AVS Quantum Science **2** (2020).

[71] A. E. Elo, *The rating of chessplayers, past and present*, 2nd ed. (Arco Pub., New York, 1986) oCLC: 15013480.

[72] A. Askell, Y. Bai, A. Chen, D. Drain, D. Ganguli, T. Henighan, A. Jones, N. Joseph, B. Mann, N. DasSarma, *et al.*, A general language assistant as a laboratory for alignment, arXiv preprint arXiv:2112.00861 (2021).

[73] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, *et al.*, Training a helpful and harmless assistant with reinforcement learning from human feedback, arXiv preprint arXiv:2204.05862 (2022).

[74] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, B. Zhu, H. Zhang, M. Jordan, J. E. Gonzalez, *et al.*, in *Forty-first International Conference on Machine Learning* (2024).

[75] M. Boubdir, E. Kim, B. Ermis, S. Hooker, and M. Fadaee, Elo uncovered: Robustness and best practices in language model evaluation, Advances in Neural Information Processing Systems **37**, 106135–106161 (2024).

[76] R. A. Bradley and M. E. Terry, Rank analysis of incomplete block designs: I. the method of paired comparisons, Biometrika **39**, 324–345 (1952).

[77] M. E. Newman, Efficient computation of rankings from pairwise comparisons, Journal of Machine Learning Research **24**, 1–25 (2023).

[78] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy, *et al.*, Text and code embeddings by contrastive pre-training, arXiv preprint arXiv:2201.10005 (2022).

[79] L. Estève, M.-C. de Marneffe, N. Melnik, A. Savary, and O. Kanishcheva, A survey of diversity quantification in natural language processing: The why, what, where and how, arXiv preprint arXiv:2507.20858 (2025).

[80] L. McInnes, J. Healy, and J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).

[81] L. Van der Maaten and G. Hinton, Visualizing data using t-sne., Journal of machine learning research **9** (2008).

[82] L. Van Der Maaten, Accelerating t-sne using tree-based algorithms, The journal of machine learning research **15**, 3221–3245 (2014).

[83] P. J. Silvia, Interest—the curious emotion, Current directions in psychological science **17**, 57–60 (2008).

[84] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (2020) pp. 38–45.

[85] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, and M. Tietz, PEFT: State-of-the-art parameter-efficient fine-tuning methods, `https://github.com/huggingface/peft` (2022).

[86] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, and Q. Gallouédec, TRL: Transformers Reinforcement Learning (2020).

[87] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[88] R. A. Fisher, On the interpretation of $\chi$ 2 from contingency tables, and the calculation of p, Journal of the royal statistical society **85**, 87–94 (1922).

[89] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, in *Proceedings of the 2023 conference on empirical methods in natural language processing* (2023) pp. 2511–2522.