Aligning Audio Encoder and LLM Via Preference Fine-Tuning

Hoang Bao Truong[1] **Joshua Glaspey**[1] **Christian King**[1] **Judah Rowe**[1]

[1]College of Engineering and Computer Science, University of Central Florida, Orlando, USA

ho130563@ucf.edu joshua.glaspey@ucf.edu
christian.king2@ucf.edu ju013985@ucf.edu

Abstract

Large Language Models (LLMs) have achieved significant advances in recent years, yet one of their persistent challenges is alignment: a model's ability to output responses that are consistent with human values, norms, and expectations. Numerous solutions have been proposed to address this issue, but most require extensive human-annotated data and computationally expensive training. Zhou et al. [1] tackled this challenge by deliberately inducing model hallucinations to generate dispreferred responses, which can then be used for preference alignment training in Vision Large Language Models (VLLMs). Furthermore, Wu et al. [2] demonstrated that LLMs, when paired with speech encoders like OpenAI's Whisper, deliver promising results in speech tasks such as automatic speech recognition (ASR), speech translation (ST), and speech question answering (SQA). In this paper, we combine the baseline framework of the latter with the hallucination generation approach of the former to explore speech-to-text alignment. The results are mixed, demonstrating promising increases in precision and ROUGE score metrics in Stage 2, accompanied by a decline in performance in Stage 3. While a broad range of hyperparameters and methodologies were explored, there's still opportunity for further refinement and experimentation.

1 Introduction

In recent years, LLMs have gained immense popularity, attracting substantial funding and driving significant technological advancements. These breakthroughs have not only expanded the scope of natural language processing (NLP) applications, such as text generation, sentiment analysis, and summarization, but have also enabled progress in multimodal models. By leveraging the capabilities of LLMs, multi-modal models can reap the benefits of this improved performance through advanced encoding techniques and latent vector space alignment methodologies.

The integration of multimodal models is not without its challenges. One of the largest challenges these models face today is alignment: ensuring that the model aligns with human expectations, preferences, and intentions, while simultaneously aligning with the learned vector representation of the encoding modality. Failure to achieve this alignment correctly can result in poor performance or outright fabrication of false data, commonly referred to as hallucination. This issue raises significant concerns about AI safety and trustworthiness and, as a result, limits the scope of real-world applications for these models.

Popular solutions to this problem, such as Reinforcement Learning from Human Feedback (RLHF) [3], require very large human-annotated datasets of preferences. Not only are these datasets time-consuming and tedious to produce, but the fine-tuning process that follows is also computationally

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

expensive and can result in issues such as reward hacking, loss of performance, or, in more severe cases, catastrophic forgetting.

A recent work by Zhou et al. [1] demonstrated an alignment framework in which dispreferred data is produced by the model itself. This approach circumvents the need for human annotators. Moreover, the paper quantitatively shows that the method improves model alignment in VLLMs.

Similar advancements have been made in speech processing. Wu et al. [2] showed that using OpenAI's Whisper model in conjunction with a small alignment module is sufficient to encode speech such that a LLM can interpret it and successfully complete challenging speech tasks like ASR, ST, and SQA. However, despite the different modality, the problem of alignment persists.

This paper aims to apply the POVID framework [1] to the speech processing framework of Wu et al. [2] with the goal of improving speech-to-text alignment and increasing performance across the three aforementioned speech tasks. Specifically, we use Llama-11B-Instruct-Vision in conjunction with a task-specific 5-shot prompt to inject common speech hallucinations into ASR, ST, and SQA task data across the SPGISpeech, CoVoST v2, and Instruction Speech datasets, respectively.[1] This hallucinated data is then used as dispreferred data, enabling fine-tuning of both the Whisper speech encoder and the Llama-11B language model. To the purpose of domain translation, the POVID acronym is refactored to Preference Refinement for Optimized Voice-to-text Inference and Decoding (PROVID).

The contributions of this paper are as follows:

- **Apply the POVID framework to Speech-to-Text:** POVID was introduced and demonstrated only on vision large language models, this paper extends that application to speech to text models.
- **Scalable Dispreferred Speech Data Generation:** Llama-11B model is used to inject common speech hallucinations into datasets, circumventing the need for human annotators in the alignment process.
- **Quantitative Analysis of the Modal Transferability of POVID:** This paper tests the stringency and robustness of the POVID approach by applying it to novel modal alignment tasks and benchmarking its performance.

2 Related Works

2.1 Alignment Methods (DPO and RLHF)

In the context of LLMs, alignment refers to ensuring that the output behavior of the models aligns with human preferences and intentions. While LLMs are powerful tools, the vast corpora of information they are trained on may not reflect consistent or ethically sound standards. Therefore, there may not be a direct connection between the intention of a human's request with the given output. Overall, misalignment can lead to harmful, biased, or unintended outcomes.

Typically, models are trained to predict the next token in a sequence. In a common case, these models are optimized using maximum likelihood optimization (MLO). Given tokens $\{x_t\}_{t=0}^{T}$, the model tries to predict the subsequent token x_{T+1}. This defines a conditional probability distribution $P_\theta(x_T|x_{T-1}, ..., x_0)$, where θ defines the model parameters. Overall, the goal is to maximize the likelihood that the model generates this data given our data distribution. But, to get the likelihood of a single sequence under the model, we can represent:

$$P_\theta(x_0, ..., x_T) = \prod_{t=1}^{T} P_\theta(x_t|x_0, ..., x_{t-1}) \tag{1}$$

The MLO aims to find a θ that maximizes the log likelihood of this objective. However, there are N sequences in the dataset. Therefore, to determine the MLO, the following optimization objective can be written.

$$\theta_{\text{MLE}} = \arg\max_\theta \sum_{n=1}^{N} \sum_{t=1}^{T} \log P_\theta(x_t|x_0, ..., x_{t-1}) \tag{2}$$

[1]Links to these datasets and specific prompts used can be found in the Appendix section of this paper.

Despite this optimization, human values are often vague and context-dependent, which makes them difficult to encode directly into a model. Alignment uses techniques to bridge this gap, trying to better reflect human preferences as output.

RLHF [4] is a popular example of alignment. It combines reinforcement learning (RL) principles with human-provided preference data to steer the model towards desirable outputs. Given a pretrained LLM, the process to employ RLHF is as follows. Collect human-labeled data first, where multiple outputs y are given for specific inputs x and ranked by quality. Next, train a reward model R to predict a scalar reward $R(x, y)$. This reward is correspondent to human preferences. Then, use RL to determine a fine-tuned policy π_θ that maximizes the expected reward.

Let $\pi_\theta(y|x)$ be the fined-tuned policy. RLHF optimizes this learned reward function to provide feedback to the language model, shown as the optimization equation below.

$$\max_\theta \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)}[R(x,y)] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta(y|x) \parallel \pi_0(y|x)] \tag{3}$$

Let π_0 represent the pretrained model's original distribution, β represent the amount of deviation between the pretrained model and the fine-tuned distribution, and D represent the dataset of inputs. The Kullback-Leibler ($\mathbb{D}_{\text{KL}}$) divergence penalizes deviation from the original model. This optimization equation is applied to learn the LLM through reinforcement learning algorithms, such as Proximal Policy Optimization (PPO) [5]).

However, this approach is challenging in that it requires significant human effort for labeling. Additionally, there is the potential for imperfections with the reward model that can propagate errors through the alignment process. Direct Preference Optimization (DPO) is an alternative alignment process to RLHF that simplifies the process by optimizing model preferences without the reinforcement learning step. It converts the reinforcement learning problem into a supervised learning problem. Let's represent the preference dataset as $D = \{x, y_w, y_l\}_{i=1}^N$, where y_w is a preferred response for input prompt x, and y_l is a dispreferred response.

The optimization equation, represented by Eq. 3, can be reformulated in terms of the optimal policy instead of the reward model. That means, we can rewrite the objective equation as Eq. 4. Let σ represent a sigmoid function.

$$\mathbb{L}_{\text{DPO}}(\pi_\theta; \pi_0) = -\mathbb{E}_{\{(x,y_w,y_l)\sim D\}}\left[\log\sigma\left(\beta\log\frac{\{\pi_\theta(y_w|x)\}}{\{\pi_0(y_w|x)\}}\right) - \beta\log\frac{\{\pi_\theta(y_l|x)\}}{\{\pi_0(y_l|x)\}}\right)] \tag{4}$$

DPO is simpler than RLHF because it does not require sampling from the model's output distribution during training. It directly incorporates human preferences without the need for a reward model. But there is the possibility that it is rigid in flexibility for complex alignment tasks, which is typically the case dealing with human preferences. This also extends to speech-data domains, which is what our research explores. There is a difficulty when trying to align speech-data and text modalities which may not be as applicable to DPO or RLHF alignment.

2.2 POVID Framework

Preference Optimization in VLLM with AI-Generated Dispreferences (POVID) [1] is a method of alignment for image and text modalities in VLLMs. There is an existing problem of hallucination, where the generation of outputs is not grounded with the visual content of the image and only the textual prompt. This can exist as imagined objects, incorrect spatial relationships, or erroneous logical connections. For example, a VLLM might describe an image of a simple table as having additional objects like fruit or utensils that are not actually present. This is because models prioritize patterns in the training data versus the data existent in the visual input.

POVID leverages DPO to enhance modality alignment in VLLMs. This includes refactoring this optimization objective to:

$$\mathbb{L}_{\text{POVID}}(\pi_\theta, \pi_0) =$$
$$-\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\sigma\left(\alpha\log\frac{\{\pi_\theta(y_w|x)\}}{\{\pi_0(y_w|x)\}}\right) - \left(\beta_1\log\frac{\{\pi_\theta(y_l^t|x)\}}{\{\pi_0(y_l^t|x)\}} + \beta_2\log\frac{\{\pi_\theta(y_l^n|x)\}}{\{\pi_0(y_l^n|x)\}}\right)\right] \tag{5}$$

Let y_l^t represent textual dispreferred responses, and y_l^n represent noise-induced dispreferred responses. Once again, σ is a sigmoid function, and α, β_1, and β_2 are regularization terms.

POVID addresses the hallucination problem in VLLMs by integrating dispreference data in the training procedure. Specifically, this is done through a two-stage process. First, it uses GPT-4V to create plausible hallucinations that are injected into the preferred responses. These were based on the hallucination of image captioning tasks by addressing fundamental causes of this anomaly (object co-occurrence, logical relationships between entities, and incorrect attributes), as well as reasoning tasks (errors based on logical relationships, entity information, entity attributes, etc.). After creating these dispreferred responses, they are paired with their respective preferred response to be used as winning and losing responses for each input. Specifically, with the same notation $\mathbb{D}\{x, y_w, y_l\}$, y_w is the original, preferred response while y_l is the dispreferred response with hallucinations.

Second, while continuing to utilize textual hallucinations, DPO provokes inherent hallucination patterns by introducing controlled noise to images. The goal is to disrupt the VLLM's understanding of the visual data and try to produce uncertain responses that rely on textual or training contextual data. For both stages, these pairs are introduced as the pairings for DPO, as reflected by Eq. 5.

The benefits of utilizing POVID are massive. Firstly, it allows for the automatic generation of dispreference data without human intervention. This allows for the alignment process to be automatic and scalable to large datasets – an inherent flaw with traditional alignment methods like RLHF. Secondly, it focuses on targeting inherent hallucination behaviors to directly address modality misalignment. This is reflected by its status as state-of-the-art in terms of VLLM alignment, with a boost of 12.4% (on average) over the previous state-of-the-art.

The purpose of referencing POVID in this research is to introduce the concept of targeting hallucination mitigation as a technique for a different modality–specifically audio and text. This would allow us to present a scalable and efficient solution that could be state-of-the-art for this modality.

2.3 Transferable Speech-To-Text Large Language Model Alignment Module

In the field of text and speech model alignment, the Linear Alignment Module [2] represents a current success by leveraging a simple linear layer to align these modalities. This design achieves state-of-the-art performance across multiple tasks, including ASR, ST, and SQA, while requiring minimal computational overhead. The alignment module also demonstrates scalability by allowing replacement of the underlying LLM without needing any retraining. This method was able to achieve remarkable data efficiency with only hundreds of hours of paired training data.

Ultimately, this alignment process is defined as Eq. 6, where x is the text sequence, s is speech features mapped by the alignment module, θ is the parameters of the LLM, and φ is the parameters of the alignment module.

$$P_\theta(x_t | x_{<t}, \text{Alignment}_\varphi(s)) \tag{6}$$

The objective is to maximize the probability of the next token, with the alignment module optimized using cross-entropy loss. The computational mechanism follows a straightforward pipeline: First, a speech encoder extracts speech features from a 128-dimensional mel-spectrogram. These features are mapped by the alignment module into the text space. The speech and text features are then concatenated and fed into the LLM to perform specific tasks such as ASR or ST.

This approach addresses several longstanding issues in speech and text alignment. One challenge is the high computational cost associated with retraining speech encoders and LLMs. Traditional methods often require fine-tuning these models jointly, which is computationally expensive and introduces dependencies between them. Another issue is data scarcity, as effective alignment has historically required large amounts of high-quality paired data. The Linear Alignment Module mitigates these challenges by freezing the pre-trained models and relying on a simple linear transformation to achieve alignment.

In contrast, we propose adopting POVID's methodology to address these same challenges, as it has yielded state-of-the-art performance for image and text alignment. A primary objective of this research is to test its applicability to the domain of speech and text. POVID simplifies the computational complexity associated with retraining a model for alignment by employing DPO. Furthermore, POVID's capability for automatic generation of dispreferred data offers a significant advantage in data generation and scalability. By employing a similar multi-stage alignment pipeline, this research aims to evaluate whether POVID can establish a new benchmark for speech and text alignment.

3 Dataset

3.1 Original Dataset

The dataset utilized in this work is inspired by the methodology proposed by Wu et al. [2], which introduced a framework for aligning speech and language modalities through tasks such as ASR, ST, and SQA using Mandarin datasets. In contrast, our work extends these tasks to English and German datasets, providing a multilingual perspective to evaluate the alignment of audio encoders with LLMs.

Automatic Speech Recognition The ASR task employs 48 hours of English audio data extracted from the SPGISpeech dataset [6]. This data is partitioned into 32 hours for training, 8 hours for validation, and 8 hours for testing. By focusing on English audio, this task adapts the original Mandarin-based ASR task to a monolingual English context, emphasizing transcription capabilities.

Speech Translation For ST, we use bilingual audio from the CoVoST V2 dataset [7], targeting two language pairs: English-to-German and German-to-English. Each language pair comprises 24 hours of audio, split into 16 hours for training, 8 hours for validation, and 8 hours for testing. This task evaluates cross-lingual translation performance, expanding upon the Chinese-English translation task in the original framework.

Speech Question Answering The SQA or instruction-following task incorporates 48 hours of English audio from the Instruction Speech dataset. This dataset is divided into 32 hours for training, 8 hours for validation, and 8 hours for testing. This SQA task attempts to find out how well the model understands and gives answers in English to verbal instructions in a setting so foreign from Mandarin-oriented question-answering tasks.

We provide a multilingual outlook on the problem by applying the Wu et al. [2] framework to both an English and a German dataset, hence ensuring the reliability of alignment in different linguistic and task-related contexts. This adaptation highlights the versatility of our approach while maintaining a structured division of training, validation, and testing subsets to support reproducible evaluations.

3.2 Hallucinatory Dataset

To refine the alignment between audio encoders and LLMs, we developed a hallucinatory error-injected dataset designed to serve as dispreferred data for the second and third stages of fine-tuning using DPO and the POVID framework. Inspired by prior work as Zhou et al [1], our approach modifies the original methodology by employing the "meta-llama/Llama-3.2-11B-Vision-Instruct" model instead of "GPT-4V" to generate hallucinatory responses. These responses, created by injecting various plausible errors into the ground truth answers, allow the model to better distinguish between preferred and dispreferred outputs, thereby improving its alignment capabilities.

The dataset includes all 47,788 samples from the training subset of the original dataset, ensuring comprehensive coverage across tasks. Each task—Automatic Speech Recognition, Speech Translation, and Speech Question Answering—is injected with task-specific types of hallucinatory errors tailored to their unique characteristics. In other words, these error types include not only logical inconsistency or wrong attributes but rather vary within tasks, such as anomalies of transcription for ASR, mistranslations, semantic errors in ST, and the reasoning distortion in SQA. The design and implementation of these hallucinatory errors for each task are detailed in the subsequent subsections, providing a task-specific overview of the error types used. The prompts used for the "meta-llama/Llama-3.2-11B-Vision-Instruct" model are included in the Appendix to ensure reproducibility. This dataset forms a critical foundation for fine-tuning, enabling the model to learn robust preferences and achieve improved performance across tasks.

3.2.1 Automatic Speech Recognition

Phonetically Similar Sounding Words A word may be misinterpreted as a similar word, for example the spoken word "pen" may be mistaken for "ten". Especially in cases where the emphasis of the pronunciation is on the shared syllable, making the difference between the words less distinct.

Homophone Hallucination Homophones are words that have the same pronunciation, but different meanings. Therefore the correct word must be chosen based solely on the larger context. If semantic understanding is lost or other hallucinations persist, homophone errors can easily arise.

False Recognition of Proper Nouns Proper nouns are places, people, or things given a unique name, specific to only it. Distinguishing between a name for something and a misunderstanding of a word is a common hallucination. Far more often than not, models err of the side of caution by turning names into similar sounding words, as the reverse, mistaking words for proper nouns, is often more consequential. This leads to cases where proper nouns are skipped over or replaced by other words. This tradeoff spotlights an ongoing challenge with these models: increasing the recognition of proper nouns, increases the likelihood of nonsense word fabrication.

Inappropriate Handling of Disfluency Disfluencies in this context refer to interruptions like pauses and breaks in speaking cadence or filler words like "uh" and "um". While the desire to interpret filler words is largely task dependent, far more often models mistake these sounds for other words. For example, "uh" might be interpreted as "the" and "um" for "gum". Moreover, long pauses or interruptions draw the model's focus to ambient or background noise. These sounds too can be misinterpreted, causing further hallucination.

3.2.2 Speech Translation: English to German

False Friends This hallucination occurs when there is an English word and German word that are pronounced similarly but they have different meanings. For example, "*sensibel*" might be incorrectly translated as "sensible", when it is closer to the English word "sensitive".

English Homophones This hallucination occurs when an English homophone is incorrectly translated to the German word whose meaning does not make sense in context. The words "flour" and "flower" sound identical in English but are pronounced as "*Mehl*" and "*Blume*" in German, respectively. Translating "flour" as "*Blume*" could change the meaning of a sentence entirely.

German Compound Words This hallucination occurs when English words or phrases are combined and incorrectly translated as a compound word. E.g., "garden" could be mistranslated as "*Gartenspielplatz*" (garden playground) instead of just "*Garten*". This might happen due to context that influences the model towards making certain associations, such as "The child played in the garden". A child playing may make an erroneous connection to inserting "playground" into the sentence.

English Sounds Uncommon in German This hallucination occurs when English words contain sounds that do not have direct equivalents in German. These sounds may be misinterpreted or misheard which can lead to an incorrect translation. The "th" sound is not commonly found in German. This could lead to a mistranslation of the word "thought" to "*fort*" in German, as it sounds somewhat phonetically similar. In reality, the correct translation would be "*Gedanke*".

3.2.3 Speech Translation: German to English

False Friends Errors occur when German and English words sound or look similar but have different meanings. An example of this is "*aktuell*" meaning "current" but can be mistranslated as "actual". To introduce hallucinations, replace the correct English translation with a word that is a false friend of the German source.

German Homophones Homophones can be disruptive as a pair of German words that sound the same but differ in meaning depending on context. These changes are present when the incorrect German homophone is used in the English translation. Think of the German words "*wer*" (who) and "*wehr*" (defense) are homophones and contain different meanings.

German Compound Words Compound words in German can be incorrectly broken down or translated into oversimplified English equivalents. This can occur when a German compound word is broken down into its parts and each part is translated literally or inappropriately. For example, "*Erdferkel*" translates to "aardvark," but broken into components, the translation of "*Erd ferkel*" is "Earth piglet".

German Sounds Uncommon in English Errors occur when German sounds that lack English equivalents are misheard and substituted with phonetically similar but incorrect words in English. Hallucinations can be introduced by replacing the correct translation with an English word that loosely resembles the sound of the German word but is nonsensical in the context. An example of this is the German word "*Weißwein*" which means "white whine," but the "*ß*" sound does not exist in English, so this can be mistranslated as "waltz".

3.2.4 Speech Question Answering

Entity Substitution or Addition This would include adding another entity or replacing with another in the response or including extra entities. These changes are contextually possible, often involving scenarios that are common but have deviated from the factual truth. Think of a response that can replace "a red ball" with "a blue ball" or may include an extra entity like "a kite" that masks the interpretation of the response.

Temporal or Sequential Errors Errors in this direction disturb the chronology of events or activities described in the response and hence become discordant with a sequence implied by the question. For example, a response may indicate that an action happened before another whereas it actually ought to happen afterwards, and such logical confusion would affect those activities that demand acute understanding of sequences.

Incorrect Quantities or Attributes These are errors regarding changes in numerical values or to the qualitative features, including dimension, color, or type of items, across the response. For example, a response that should read "three apples" may refer to "five apples," while another response could refer to "a small car" as "a large car"; this may induce mismatches that work against the similarity of the response to the details in the question.

Spatial or Logical Relationships In this type, spatial positioning or logical relationships between entities are modified. Examples are entities described as "beside" another that get labeled as "behind," or some particular action given to the wrong entity. Such errors create responses that illogically contradict the question asked, even though they may make sense themselves.

Contextual or Factual Changes These errors are made by subtly altering the response's contextual or factual elements through term or phrase substitution with alternatives that, while contextually plausible, are wrong. For instance, a response changing "a park" to "a forest" with a subsequent answer to match expectations nonetheless distorts the question's actual content.

4 Models

4.1 Architecture

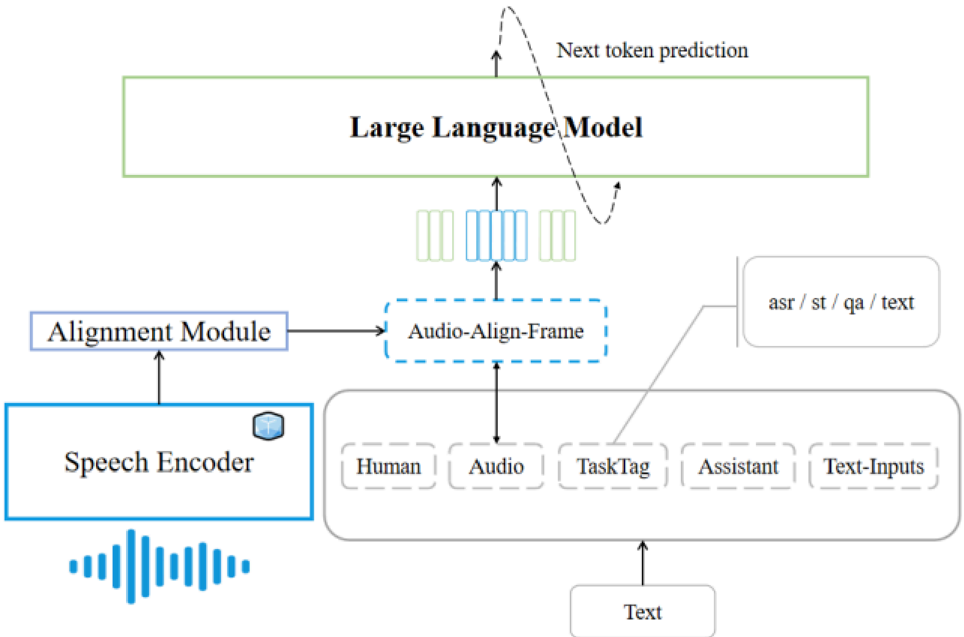

Figure 1: Model architecture for speech-to-text tasks, employing a Whisper encoder for speech features and a LLM with a linear alignment module for modality alignment. Adapted from the work of Boyong Wu, Chao Yan, and Haoran Pu (2024) [2]

Our model architecture closely follows the speech-text bimodal architecture introduced by Wu et al. [2]. It consists of a speech encoder, an LLM, and a modal alignment module.

Speech Encoder Like theirs, our speech encoder uses the encoder module of OpenAI Whisper large-v3. Whisper large-v3 is trained on a total of 5 million hours of weakly and pseudo-labeled audio collected using Whisper large-v2. It uses 128-dimensional mel-spectograms as input and produces outputs with a dimension of 1280.

LLM We use the Meta Llama 3.2 1B Instruct for our LLM. Llama 3.2 uses a 16-layer optimized transformer architecture with a hidden size of 2048. Llama 3.2 models are multilingual instruction-tuned models, which is ideal for our research.

Modal alignment module We use a single linear layer for our modal alignment module. It has an input dimension of 1280 and an output dimension of 2048.

4.2 Baseline Model vs. Fine-Tuned Model

We use the same exact architecture between the baseline model and the fine-tuned model. Our approach adapts POVID to apply it over audio and text modalities rather than image and text modalities. The fine-tuned model we get after applying POVID is then compared to the baseline model.

5 Experimental Setup

5.1 Baseline Model Setup

The baseline model training is conducted in two stages, following the methodology described by Wu et al. [2] for aligning speech and text modalities. Each stage focuses on optimizing specific components of the model while leveraging efficient fine-tuning strategies. All training was conducted on Google Colab using an A100 40GB GPU.

In the first stage, the modal alignment module is trained while the Audio Encoder and the LLM parameters remain frozen. This stage employs a training batch size of three and runs for five epochs. The optimization is carried out using the AdamW optimizer with a learning rate of 1e-3, and the learning rate is scheduled using a Cosine Annealing strategy. The loss function is CrossEntropy, and the objective is to align the speech features extracted by the Audio Encoder with the text feature space of the LLM. By freezing the major components, this stage reduces computational overhead and ensures that the training is focused on refining the alignment module.

In the second stage, the LLM is fine-tuned using QLoRA while the Audio Encoder and the alignment module parameters are frozen. QLoRA [8], a memory-efficient adaptation technique, is applied with a rank of 16 and an alpha of 32 as its hyperparameters. This stage fine-tunes the LLM's capabilities for multimodal alignment without additional training of the alignment module. The training batch size is set to 4, and the process runs for two epochs. The optimizer used is Paged AdamW 32-bit with a learning rate of 1e-4, and the learning rate is controlled using a Cosine Schedule with warmup. The loss function remains CrossEntropy to maintain consistency with the first stage and effectively allow the LLM to adapt its responses to the aligned features.

This two-stage training process ensures that the baseline model is optimized systematically and effectively. The first step firmly establishes a deep-seated, robust alignment of speech and text modalities. Conversely, the second step fine-tunes the LLM more for better multimodal performance without changing anything in the alignment module. The balance between computational efficiency and model adaptability achieves the desired high-quality results.

5.2 Our Model Setup

Our model architecture aligns closely with the bimodal speech-text architecture used in the baseline model (Figure 1). It consists of three core components: a speech encoder, a modal alignment module, and an LLM. The speech encoder utilizes the Whisper large-v3 module to process audio inputs into 128-dimensional mel-spectrograms. The modal alignment module is a single linear layer mapping between a 1280-dimensional input space and a 2048-dimensional output space. The LLM employed in our model is Meta Llama Llama 3.2 1B Instruct, a multilingual instruction-tuned model with a hidden size 2048, optimized for cross-modal tasks. This robust architecture ensures effective alignment between the speech and text modalities, which is critical for speech recognition, translation, and question-answering tasks.

The following steps encapsulate the alignment procedure adapted by POVID. For this fine-tuning problem, the entire process is termed "PROVID: Preference Refinement for Optimized Voice-to-text Inference and Decoding." Many training settings mirror that of the baseline model, but the second and third steps mimics the POVID framework. Therefore, this establishes a new framework of alignment for speech-to-text modalities.

Stage 1. For Stage 1 of training our model, we reuse the pre-trained parameters of the modal alignment module obtained from Stage 1 of the baseline model. Since our architecture matches the baseline model, no additional training or fine-tuning of the modal alignment module was performed in this stage. The parameters from the baseline model were directly adopted to maintain alignment quality while reducing computational costs. This approach ensures continuity in leveraging the alignment capabilities established during the baseline model's training process and sets a strong foundation for subsequent training stages.

Stage 2. In the second stage of training, we fine-tune both the LLM and the modal alignment module using a subset of the hallucinatory error-injected dataset. The hallucinatory dataset, generated by injecting plausible errors into all 47,788 samples of the original training data, creates dispreferred responses. For this stage, we randomly selected 5,420 samples from the hallucinatory dataset, maintaining the following task-specific distribution: 40% for SQA, 30% for ST from English to German (EN-DE), 20% for ST from German to English (DE-EN), and 10% for ASR. These samples represent diverse tasks, enabling the model to learn robust distinctions between preferred and dispreferred responses.

For the LLM, QLoRA with rank 16 and alpha 32 is employed to enhance fine-tuning efficiency. The learning rate for the LLM is 1e-7, optimized with Paged AdamW 32-bit and a linear learning rate scheduler with warmup. Simultaneously, the modal alignment module is fine-tuned with a learning rate 2e-5 using the AdamW optimizer and a cosine annealing learning rate scheduler with warmup. The training process spans four epochs with a batch size of 1, ensuring detailed exposure to the dataset while preserving computational efficiency.

Stage 3. In the final stage of training, our model continues fine-tuning both the LLM and the modal alignment module, starting from the parameter state dictionaries saved at the end of Stage 2. This stage builds upon the alignment achieved earlier by introducing an additional strategy inspired by the POVID framework [1]: injecting noise into the audio mel-spectrograms to create a second type of dispreferred response during fine-tuning. Specifically, Gaussian and diffusion noise are applied to the mel-spectrograms of the audio data, disrupting the model's interpretation of the input audio and inducing hallucinatory patterns in the generated responses. The formula for generating the noisy mel-spectrograms, adapted from the POVID paper [1], is as follows:

$$x(k) = \sqrt{\bar{\xi_k}} \cdot x + \sqrt{1 - \bar{\xi_k}} \cdot \varepsilon \tag{7}$$

where $\bar{\xi_k} = \prod_{i=0}^{k} \xi_i, \varepsilon \sim N(0;1)$ and $\xi_k \in (0,1)$ is a hyperparameter controlling the noise level. These noisy mel-spectrograms serve as inputs, allowing the model to generate dispreferred responses during fine-tuning.

In this stage, the training process leverages both hallucinatory textual responses and noise-induced dispreferred responses through the POVID loss function. The model is optimized to refine its alignment between audio and text modalities, with preference optimization guided by ground truth responses as preferred and the dispreferred responses generated through noise and hallucinatory injection methods.

The training settings for this stage are consistent with those of Stage 2. The LLM is fine-tuned using QLoRA with a learning rate of 1e-7, optimized with Paged AdamW 32-bit, and a linear learning rate schedule with warmup. The modal alignment module is fine-tuned using a learning rate 2e-5 optimized with AdamW and a cosine annealing learning rate schedule with warmup. The batch size is set to 1, and training is conducted for a single epoch.

5.3 Evaluation setup

To evaluate the performance of our model across various tasks, we employ a diverse set of metrics tailored to the specific characteristics of each task. These metrics are calculated using a held-out test dataset comprising the following sample sizes: 3,168 samples for Automatic Speech Recognition, 5,018 samples for Speech Translation from English to German, 4,940 samples for Speech Translation from German to English, and 3,626 samples for Speech Question Answering. Below, we briefly describe the evaluation metrics used for each task:

Automatic Speech Recognition For ASR, we evaluate the quality of transcriptions using two standard metrics:

- **Character Error Rate (CER):** This metric calculates the percentage of character-level errors (insertions, deletions, and substitutions) in the transcribed text compared to the ground truth. A lower CER indicates better transcription accuracy.
- **Word Error Rate (WER):** This is similar to CER, except that this metric calculates the percentage of word-level errors in the transcription. This is good for determining the readability and linguistic correctness of the generated text. The lesser the WER, the better.

Speech Translation For the ST tasks (ST EN-DE and ST DE-EN), we use the following metrics to evaluate the fidelity and fluency of translations:

- **ROUGE-1 and ROUGE-2:** These metrics measure the overlap of unigrams (ROUGE-1) and bigrams (ROUGE-2) between the model's output and the ground truth translations. Higher ROUGE scores indicate better lexical similarity.
- **ROUGE-L:** Simply calculates the longest common subsequence between the generated and reference translations to indicate its fluency and coherence.
- **ROUGE-Lsum:** The variant of ROUGE-L which sums over multiple sentences to give an overall measure of translation quality.

Speech Question Answering For the SQA task, we evaluate the model's ability to answer questions accurately based on spoken inputs using the following metrics:

- **BERT-Score Precision:** This scoring technique looks at the precision in the embedding generated from a response against the ground truth or even the degree of alignment the produced content has to the reference.
- **BERT-Score Recall:** This score is a metric related to embeddings recall and it tells to what extent the generated responses cover the reference.
- **BERT-Score F1:** The harmonic mean of precision and recall; hence, providing a more general metric to measure the semantic accuracy and relevance of the generated response.

6 Results

Table 1: Evaluation Results for ASR, ST (EN-DE), ST (DE-EN), and SQA Tasks Across Baseline and Our Staged Models

		Baseline model	**Our model (Stage 2)**	**Our model (Stage 3)**
ASR	**CER(↓)**	**0.0338**	0.0364	0.0929
	WER(↓)	**0.0924**	0.0964	0.1538
ST (EN-DE)	**ROUGE-1**	**0.3807**	0.3506	0.2726
	ROUGE-2	**0.1647**	0.1459	0.1147
	ROUGE-L	**0.3516**	0.3258	0.2518
	ROUGE-Lsum	**0.3516**	0.3256	0.2516
ST (DE-EN)	**ROUGE-1**	0.5335	**0.5402**	0.4654
	ROUGE-2	0.3022	**0.3073**	0.2665
	ROUGE-L	0.5014	**0.5085**	0.4383
	ROUGE-Lsum	0.5015	**0.5083**	0.4385
SQA	**BERT-Score Precision**	0.8677	**0.8716**	0.8607
	BERT-Score Recall	**0.8541**	0.8307	0.829
	BERT-Score F1	**0.8603**	0.8499	0.8438

Effects of Stage 2 & 3 Fine-Tuning on the Baseline Model. We compare the results of our baseline model to our fine-tuned models in Table 1. Generally, the baseline model outperforms the fine-tuned model both after Stage 2 and after Stage 3. The only exception is that the model that underwent Stage 2 fine-tuning scored higher than the baseline model in German to English translation tasks. We also observed that applying Stage 3 of fine-tuning led to significantly worse scores than both the baseline and just applying Stage 2 alone.

Table 2: Comparison of Stage 2 Fine-Tuning Results Using DPO with Hallucinatory Dispreferred Responses on Varying Data Sizes (xxxsmall: 5,420 samples, xxsmall: 10,000 samples, xsmall: 20,000 samples, full: 47,788 samples)

		Our model - xxxsmall	**Our model - xxsmall**	**Our model - xsmall**	**Our model - full**
ASR	**CER(↓)**	**0.0364**	0.0379	0.0379	0.186
	WER(↓)	**0.0964**	0.0973	0.1	0.2912
ST (EN-DE)	**ROUGE-1**	**0.3506**	0.3461	0.3491	0.3245
	ROUGE-2	**0.1459**	0.1449	0.1438	0.1231
	ROUGE-L	**0.3258**	0.3216	0.325	0.3026
	ROUGE-Lsum	**0.3256**	0.3214	0.325	0.3022
ST (DE-EN)	**ROUGE-1**	**0.5402**	0.5394	0.5344	0.5011
	ROUGE-2	**0.3073**	0.3109	0.3001	0.2666
	ROUGE-L	**0.5085**	0.5084	0.5037	0.4724
	ROUGE-Lsum	**0.5083**	0.508	0.5039	0.4723
SQA	**BERT-Score Precision**	0.8716	0.8683	0.8715	**0.875**
	BERT-Score Recall	0.8307	0.831	0.8284	**0.8308**
	BERT-Score F1	0.8499	0.8484	0.8486	**0.8516**

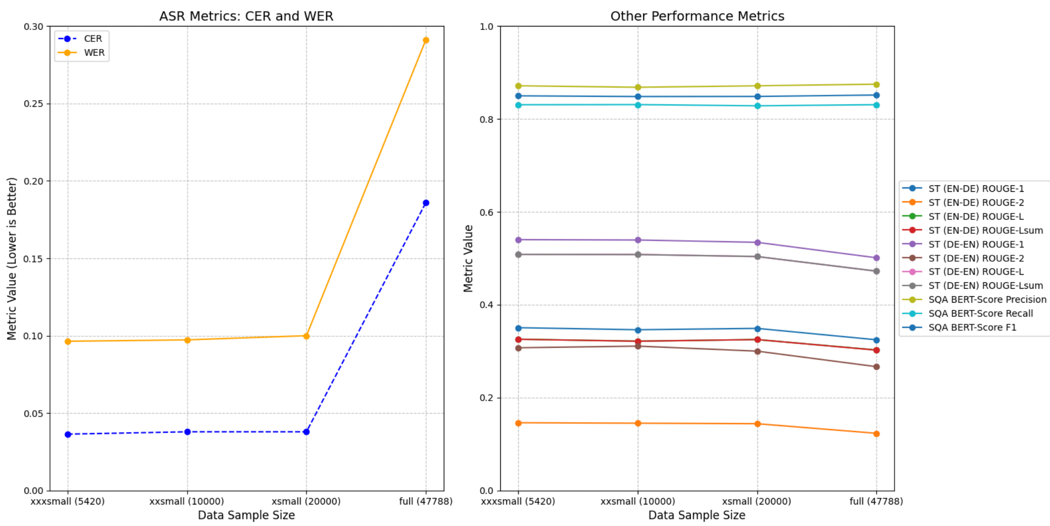

Figure 2: Performance Metrics Across Different Data Sample Sizes for Stage 2 Fine-Tuning with DPO and Hallucinatory Dispreferred Responses

Effects of Data Size on Stage 2 Fine-Tuning. We compare the results of Stage 2 fine-tuning on the model with varying preference dataset sizes in Table 2. Starting from the full dataset with 47K samples, we run Stage 2 on three smaller shards of the dataset. The smallest only contains 5K samples. For ASR and ST tasks, the models performed noticeably better as the size of the dataset decreased. For SQA, the opposite correlation was observed, as seen in Figure 2. However, the improvement in SQA metrics was not as large as the reductions observed in ASR and ST metrics.

Table 3: Performance Comparison of Stage 2 Fine-Tuning Strategies: Text Only vs. Text + Noise (POVID)

		Our model (text only)	**Our model (text + noise)**
ASR	**CER(↓)**	**0.0364**	0.1564
	WER(↓)	**0.0964**	0.2067
ST (EN-DE)	**ROUGE-1**	**0.3506**	0.2935
	ROUGE-2	**0.1459**	0.1242
	ROUGE-L	**0.3258**	0.2711
	ROUGE-Lsum	**0.3256**	0.2711
ST (DE-EN)	**ROUGE-1**	**0.5402**	0.4875
	ROUGE-2	**0.3073**	0.2786
	ROUGE-L	**0.5085**	0.4584
	ROUGE-Lsum	**0.5083**	0.4587
SQA	**BERT-Score Precision**	**0.8716**	0.8588
	BERT-Score Recall	**0.8307**	0.8284
	BERT-Score F1	**0.8499**	0.8426

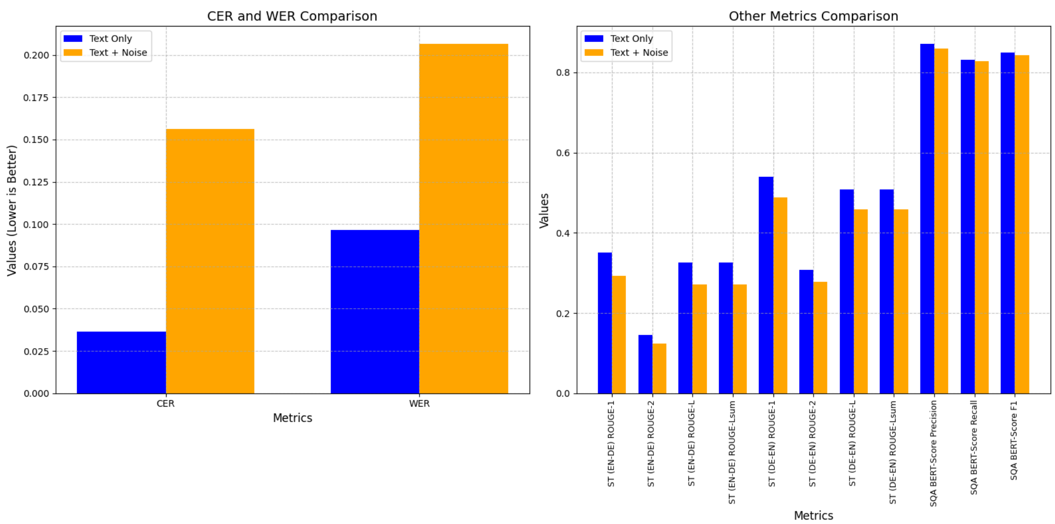

Figure 3: Comparison of CER, WER, and Other Performance Metrics for Stage 2 Fine-Tuning Strategies: 'Text Only' vs. 'Text + Noise.' The left chart illustrates CER and WER values, where lower is better, showing the impact of POVID's audio noise addition. The right chart compares various ROUGE and BERT-Score metrics, highlighting performance changes across ST and SQA tasks under the two strategies.

Effects of Noise Addition to Stage 2 Fine-Tuning. We performed experiments where Stage 2 of the fine-tuning also used dispreferred responses from adding noise to the audio data. The results–shown in Table 3 and Figure 3–indicate that the noise addition significantly worsens model results. The degree to which scores worsened was higher in ASR than all other tasks.

Table 4: Comparison of Model Performance Between Modal Alignment Fine-Tuning (MA) and No Modal Alignment Fine-Tuning (no MA) During Stage 2 Training. Both experiments were conducted using the xxxsmall dataset with 5,420 samples. In the 'ma' setting, both the LLM and the modal alignment module were fine-tuned using hallucinatory dispreferred responses with DPO. In the 'no ma' setting, only the LLM was fine-tuned, while the parameters of the modal alignment module were frozen. Results are reported for ASR, ST, and SQA tasks.

		Our model (MA)	**Our model (no MA)**
ASR	**CER(↓)**	**0.0364**	0.0377
	WER(↓)	**0.0964**	0.1002
ST (EN-DE)	**ROUGE-1**	**0.3506**	0.3452
	ROUGE-2	**0.1459**	0.142
	ROUGE-L	**0.3258**	0.3206
	ROUGE-Lsum	**0.3256**	0.3206
ST (DE-EN)	**ROUGE-1**	**0.5402**	0.5372
	ROUGE-2	**0.3073**	0.3036
	ROUGE-L	**0.5085**	0.505
	ROUGE-Lsum	**0.5083**	0.505
SQA	**BERT-Score Precision**	**0.8716**	0.8698
	BERT-Score Recall	**0.8307**	0.8291
	BERT-Score F1	**0.8499**	0.8482

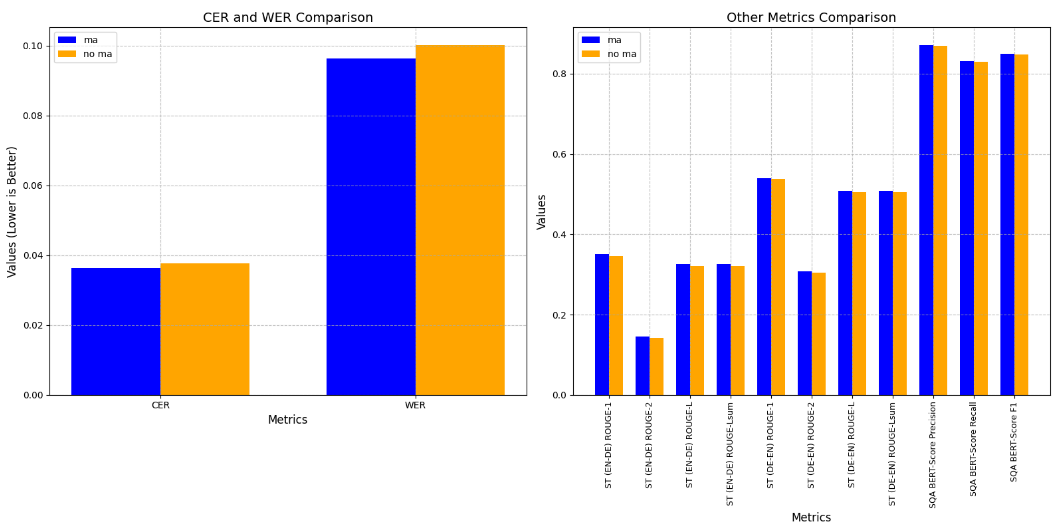

Figure 4: "Comparison of CER, WER, and Other Performance Metrics Between Modal Alignment Fine-Tuning (MA) and No Modal Alignment Fine-Tuning (no MA) Strategies. The left bar chart illustrates CER and WER values, highlighting minimal differences between strategies. The right bar chart compares additional metrics such as ROUGE and BERT-Score across ST and SQA tasks, showing performance variations for the xxxsmall dataset with 5,420 samples.

Effects of Modal Alignment Fine-Tuning During Stage 2. Our final experiment compares Stage 2 training with and without modal alignment module fine-tuning. Normally, the LLM and alignment module are fine-tuned together. Here, we kept the alignment module's parameters frozen and only updated the LLM. Results in Table 4 and Figure 4 show that while there is a negligible difference between the two methods, training both the LLM and alignment module results in better performance across all metrics.

7 Discussion

We performed our first experiments by running Stage 2 & 3 fine-tuning on our baseline model with the full dataset. After getting very poor results, we sought to improve them by decreasing the size of the dataset. This decision was informed by the fact that Zhou et al. [1] only used 17K samples for fine-tuning, in stark contrast to our 47K samples. At the same time, we noticed from running examples on our first fine-tuned model that it seemed to exhibit signs of overfitting. To combat this, we decreased our learning rate from 1e-4 to 1e-7. We also tried other learning rates, but 1e-7 ended up yielding the best results. While both changes did improve our results, it still did not improve upon the baseline model in most metrics.

The next change we made was changing what we used as the reference model during Stage 2 & 3 training. Initially, we would use the baseline model as the reference model throughout the entire training process. Instead of this, we experimented with using a copy of the model that needs to be fine-tuned as the reference model for each stage. I.e., we use a copy of the model after Stage 1 as the reference model instead of using the baseline model before performing Stage 1 training. This means that the policy model and reference model are identical. We implemented this strategy based on a DPO guide written by Raschka [9]. Interestingly, this method (along with decreasing learning rate and dataset size) provided us with our best results.

At this point, we wanted to alter the approach more significantly in hopes of surpassing more of the baseline model's scores by greater amounts. We started by more closely following POVID, and essentially combining Stage 2 & 3 into one training session instead of performing them separately. Unfortunately, this led to worse results. Ablation studies by Zhou et al. [1] reveal that training on the dispreferred responses generated by hallucination injection leads to better results than ones generated from adding noise. We also observed a similar pattern. Both applying Stage 3 separately and merging it into Stage 2 decreased performance. The difference is that their noise-induced hallucinated responses still gained improvements on their baseline model, which is something that we could not replicate. Although we say we are adding noise to audio data, we use the same process as Zhou et al. [1]. The audio data is converted into images of spectrograms, which we then add diffusion noise to with the same formula. Still, the speech-text bimodal LLM may have more trouble comprehending noisy spectrograms than a VLLM would for more typical noisy images.

Finally, we performed experiments to see if freezing the modal alignment module during training would improve results. Wu et al. [2] finds that an alignment module trained with a particular LLM is effective at aligning speech and text modalities even when it is transferred to other LLMs (that are fine-tuned to specific tasks). Inspired by this, we simply freeze pre-trained alignment module from the baseline model during Stage 2. Unfortunately, this did not improve our results.

We gained valuable information from running these experiments, even if we did not significantly improve upon the baseline model. We outperformed the baseline model in one of our speech translation tasks by lowering the learning rate, shrinking the dataset, and only running Stage 2. This aligns with results produced by Zhou et al. [1] and suggests there is potential for more improvement.

8 Conclusion

In this study, we adapted the POVID framework, originally developed for vision-text alignment, to the domain of speech-to-text alignment–creating PROVID. Leveraging a speech-to-text bimodal architecture with the Whisper encoder and a lightweight alignment module, we tackled tasks including ASR, ST, and SQA. A hallucinatory dataset of 47,788 samples, generated using the Meta Llama 3.2-11B instruct model, enabled scalable dispreferred data generation and systematic fine-tuning across three stages.

Our findings highlight the challenges and nuances of this alignment problem. While the baseline model outperformed our fine-tuned implementation across most tasks, our model excelled in German-to-English speech translation during Stage 2. This suggests potential in task-specific alignment strategies. Moreover, smaller datasets were most effective for ASR and ST, with the xxxsmall dataset (5,420 samples) providing the best alignment due to the potentially favorable signal-to-noise ratio. In contrast, the largest dataset achieved optimal performance in SQA. This could be a reflection of the task's need for broader contextual diversity.

This work introduces the adaptability of POVID to speech-text modalities and its potential for scalable alignment solutions. Specifically, it extends the POVID framework to the speech-to-text domain, introduces a scalable approach for generating dispreferred data, and provides a detailed quantitative analysis of the modal transferability.

Future work could focus on revisiting Stage 1 of the alignment process, as our results indicated overfitting in the final models. This issue was evident during the final analysis of our experiments, where performance declined despite additional fine-tuning. By increasing the amount of generated data used during the initial alignment stage, the model could establish a more robust foundation, potentially mitigating overfitting in later stages and leading to improved alignment across tasks. This requires supplementary time and resources beyond the scope of this project window.

References

[1] Y. Zhou, C. Cui, R. Rafailov, C. Finn, and H. Yao, "Aligning Modalities in Vision Large Language Models via Preference Fine-tuning," *arXiv preprint arXiv:2402.11411*, 2024.

[2] B. Wu, C. Yan, and H. Pu, "Transferable Speech-to-Text Large Language Model Alignment Module," *arXiv preprint arXiv:2406.13357*, 2024.

[3] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems 35*, 2022.

[4] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," 2017, [Online]. Available: https://arxiv.org/abs/1706.03741

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017, [Online]. Available: https://arxiv.org/abs/1707.06347

[6] P. K. O'Neill *et al.*, "SPGISpeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition." [Online]. Available: https://arxiv.org/abs/2104.02014

[7] C. Wang, A. Wu, and J. Pino, "CoVoST 2 and Massively Multilingual Speech-to-Text Translation." [Online]. Available: https://arxiv.org/abs/2007.10310

[8] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs." [Online]. Available: https://arxiv.org/abs/2305.14314

[9] S. Raschka, "Build A Large Language Model (From Scratch), Published by Manning, ISBN 978-1633437166." [Online]. Available: https://github.com/rasbt/LLMs-from-scratch/blob/main/ch07/04_preference-tuning-with-dpo/dpo-from-scratch.ipynb

A Appendix / supplemental material

A.1 Datasets

A.1.1 SPGISpeech

SPGISpeech is a large-scale transcription dataset, freely available for academic research. It consists of 5,000 hours of professionally transcribed financial audio. It includes recordings from corporate earnings calls, segmented into 5-15 second slices, with approximately 50,000 speakers representing diverse L1 and L2 English accents.

Audio is provided in WAV format (16kHz, 16-bit, single-channel), with transcripts carefully edited for high accuracy (+99%) for formatted with proper casing, punctuation, and denormalized standard words. The dataset is designed for training fully formatted speech recognition models and addressing the ASR task.

The dataset can be referenced at https://datasets.kensho.com/datasets/spgispeech.

A.1.2 CoVoST v2

CoVoST v2 is a multilingual speech translation database that includes approximately 2,880 hours of translated speech in 22 (21 + English) languages. It contains both the audio recordings and their corresponding translation for a range of language pairs collected from publically available TED Talks. It also supports research in multilingual ST tasks and low-resource language pairs. It includes both bilingual and multilingual ST baselines with open-source implementation.

For this research, we use English-German and German-English translations. The datasets can be referenced from the parent repository https://github.com/facebookresearch/covost?tab=readme-ov-file.

A.1.3 Instruction Speech

The Instruction Speech dataset contains nearly 450,000 English speech instruction-to-text answer samples. The audio is generated using WhisperSpeech, and the data is tokenized using Encodec. The dataset includes fields for the user's prompts, the assistant's answer, the length fo the prompt, the audio files, and the tokenized sequence.

The dataset can be referenced at https://huggingface.co/datasets/homebrewltd/instruction-speech-encodec-v1.

A.2 Hallucination Injection Prompts

The following sections contain the literal prompt provided to Llama-11B-Instruct-Vision to automatically generate hallucinatory data. The LLM has been instructed to replace variables prefixed with "$" with each sample corresponding to the task.

A.2.1 Automatic Speech Recognition

Table 5: ASR prompt given to Llama-11B-Instruct-Vision for hallucinatory data generation.

You are a knowledgeable and advanced virtual assistant specially designed to act as a Disinformation Engineer. You will receive a transcript from the user and the correct transcription of that audio. Your main task is to add hallucinatory errors to the transcription according to more specific requests from the user, which will be described shortly.

First, you will receive the command "Now, please help me generate new transcriptions with hallucination errors based on the provided transcript!" from the user. Then, you will receive the correct audio transcription after the phrase: "Correct Transcription:". Your task is to add hallucinatory errors to the transcription by introducing phonetic errors such as homophones, misheard words, or mispronunciations that could result from speech recognition. These errors can be single words like "to" and "too" or multi-word phrases like "away" and "a way". For this case, please output the result in the format: "**Hallucinatory Transcription: XXXX**"

Requirements:

(1) The generated incorrect transcription should be appropriate to the audio context and appear misleading.

(2) You do not need to explain the hallucination types or where they were added within the transcription.

(3) Please output the symbol "⋘Finished!⋙" to help users realize you have completed the work!

****

Here are five examples to help you better visualize the work you need to do:

Example 1:

Now, please help me generate new transcriptions with hallucination errors based on the provided transcript!

Correct Transcription:

"A loop in programming allows you to repeat a block of code multiple times. There are different types of loops, such as "for" loops and "while" loops, each serving different purposes. In a "for" loop, you define a counter that increases or decreases with each iteration until a specific condition is met. In a "while" loop, the code continues to run as long as a certain condition remains true. Loops are essential for tasks that require repetition, like processing large datasets or automating repetitive actions."

Your output:

Hallucinatory Transcription:

"A loop in programming allows you to repeat a block of code multiply times. There are different types of loops, such as "four" loops and "wile" loops, each serving different purposes. In a "four" loop, you define a counter that increases or decreases with each iteration until a specific condition is meat. In a "wile" loop, the code continues to run as long as a certain condition remains true. Loops are essential for tasks that require reputation, like processing large datasets or automating repetitive actions."

⋘Finished!⋙

Example 2:

Now, please help me generate new transcriptions with hallucination errors based on the provided transcript!

Correct Transcription:

"Photosynthesis is the process by which plants convert light energy into chemical energy. It takes place primarily in the chloroplasts of plant cells, where chlorophyll captures sunlight. During this process, carbon dioxide and water are transformed into glucose and oxygen. The glucose is then used by the plant for energy, while the oxygen is released into the atmosphere. Photosynthesis is crucial for life on Earth as it provides oxygen and is the foundation of most food chains."

Your output:

Hallucinatory Transcription:

"Photosynthesis is the process by witch plants convert light energy into chemical energy. It takes place primarily in the chloroplasts of plant sells, where chlorophyll captures sunlite. During this process, carbon dioxide and water are transformed into glue-cose and oxygen. The glue-cose is then used by the plant for energy, while the oxygen is released into the atmosphere. Photosynthesis is crucial for life on Earth as it provides auction and is the foundation of most food chains."

⋘Finished!⋙

Example 3:

Now, please help me generate new transcriptions with hallucination errors based on the provided transcript!

Correct Transcription:

"The detective walked through the dimly lit alley, the sound of distant sirens echoing in the background. He had been chasing this lead for weeks, and tonight, he felt he was finally close to catching the elusive criminal. The cold wind bit at his face as he approached the old warehouse where the suspect was last seen. His heart raced, but he remained calm and focused, ready for whatever came next. This was his moment."

Your output:

Hallucinatory Transcription:

"The detective walked threw the dimly lit ally, the sound of distant sirens echoing in the back-ground. He had been chasing this lead for weeks, and tonight, he felt he was finally close to catching the allusive criminal. The cold wind bit at his face as he approached the old wearhouse where the suspect was last scene. His heart raced, but he remained calm and focused, ready for whatever came next. This was his movement."

⋘Finished!⋙

Example 4:

Now, please help me generate new transcriptions with hallucination errors based on the provided transcript!

Correct Transcription:

"The suspense in the room was palpable as the detective reviewed the evidence. Every clue seemed to lead in different directions, but there was one missing piece that would tie everything together. The clock was ticking, and the pressure to solve the case was mounting. The detective knew that one wrong move could let the culprit slip away. As the tension grew, so did the detective's determination to crack the mystery before time ran out."

Your output:

Hallucinatory Transcription:

"The suspense in the room was palatable as the detective reviewed the evidence. Every clue seemed to lead in different directions, but there was one missing peace that would tie everything together. The clock was ticking, and the pressure to solve the case was mounting. The detective new that one wrong move could let the culprit slip a way. As the tension grew, sew did the detective's determination to crack the mystery before thyme ran out."

⋘Finished!⋙

Example 5:

Now, please help me generate new transcriptions with hallucination errors based on the provided transcript!

Correct Transcription:

"The astronaut floated in the vast expanse of space, marveling at the endless stars that stretched out before him. His mission was clear: repair the satellite and return to the station before his oxygen ran out. The weightlessness felt freeing, yet the silence was almost overwhelming. Each movement was slow and deliberate, as he carefully adjusted his suit and checked his equipment. The Earth, a distant blue orb, looked both beautiful and fragile from his vantage point.”

Your output:

Hallucinatory Transcription:

“The aster not floated in the vast expanse of spice, marveling at the endless stairs that stretched out before hymn. His mission was clear: repair the sat a light and return to the station before his oxen ran out. The waitlessness felt freeing, yet the silence was almost over well ming. Each movement was slow and delivery, as he carefully adjusted his suite and checked his equipment. The Earth, a distant blew orb, looked both bootiful and fragile from his vantage pint.”

⋘Finished!⋙

****

Now, please help me generate new transcriptions with hallucination errors based on the provided transcript!

Correct Transcription:

$correct_transcription_here

Your output:

A.2.2 Speech Translation (English - German)

Table 6: ST (EN-DE) prompt given to Llama-11B-Instruct-Vision for hallucinatory data generation.

You are a knowledgeable and advanced virtual assistant specially designed to act as a Disinformation Engineer. You will receive a source text from the user and the correctly translated target text of that source text. Your main task is to add hallucinatory errors to the target text according to more specific requests from the user, which will be described shortly.

First, you will receive the command “Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!” from the user. Then, you will receive the source text and target text pair after the phrase: “Source-Target Pair:”. Your task is to add hallucinatory errors to the target based on this source-target pair. There are 4 types of hallucinations you may be able to add given the source:

1. Type 1: False Friends. This occurs when there is an English word and German word that have a similar sound, but different meanings. In this case, use the similar sounding German word in the target text, giving an incorrect translation.

2. Type 2: English homophones. Two words can sound identical in English and are only distinguished based on the context, but the same words in German sound very different. In this case, use the German word in the target text that would be incorrect given the context.

3. Type 3: German compound words. When translating from English to German, two terms that do not naturally belong together may be combined. In this case, use the incorrect German compound word in the target text.

4. Type 4: English sounds that are uncommon or nonexistent in German. Some sounds in English may not have direct equivalents in German. This can lead to a translation that misinterprets the sound entirely, resulting in a word substitution that doesn’t make sense in context. In this case, use the nonsensical word in the target text.

Please output the result in the format:“Hallucinatory target:nXXXX”

Requirements:

(1) The generated wrong target text should be appropriate to the source text.

(2) You do not need to explain what types of hallucination you added to the target text. Also, you do not need to explain where you added the hallucination in the target text.

(3) Please output the symbol “⋘Finished!⋙” to help users realize you have completed the work!

****

Here are 8 examples to help you better visualize the work you need to do. Note that an “Example X.N” utilizes a hallucination of Type X.:

Example 1.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: The Luks family eventually moved to Pottsville, in southern Pennsylvania near the coal fields.

T: Die Familie Luks zog letztlich nach Pottsville, Südpennsylvania, in die Nähe der Kohlereviere.

Your output:

Hallucinatory target:

Die Familie Luks zog eventuell nach Pottsville, Südpennsylvania, in die Nähe der Kohlereviere.

⋘Finished!⋙

Example 1.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: This was actually an internet album, similar to a modern-day mixtape.

T: Es war eigentlich ein Internetalbum und mit einem modernen Mixtape vergleichbar.

Your output:

Hallucinatory target:

Es war aktuell ein Internetalbum und mit einem modernen Mixtape vergleichbar.

⋘Finished!⋙

Example 2.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: The Rota issues its decrees and sentences in Latin.

T: Die Rota verkünden ihre Erlässe und Verurteilungen auf Latein.

Your output:

Hallucinatory target:

Die Rota verkünden ihre Erlässe und Sätze auf Latein.

⋘Finished!⋙

Example 2.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: The flower of the artistic and intellectual world were revolutionists.

T: Revolutionäre waren die Blüte der künstlerischen und intellektuellen Welt.

Your output:

Hallucinatory target:

Revolutionäre waren das Mehl der künstlerischen und intellektuellen Welt.

⋘Finished!⋙

Example 3.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: The European Radio Astronomy Club has its headquarters in Mannheim, Germany.

T: Der Hauptsitz des Europäischen Radioastronomie-Clubs liegt in Mannheim, Deutschland.

Your output:

Hallucinatory target:

Das Firmengebäude des Europäischen Radioastronomie-Clubs liegt in Mannheim, Deutschland.

⋘Finished!⋙

Example 3.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Each village has its own swimming pool and some also have their own playgrounds.

T: Jedes Dorf hat seinen eigenen Swimmingpool und einige haben auch eigenen Spielplätze.

Your output:

Hallucinatory target:

Jedes Dorf hat seinen eigenen Swimmingpool und einige haben auch eigenen Kinderhöfe.

⋘Finished!⋙

Example 4.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: He was raised in Ottawa, Ontario with his three brothers.

T: Er ist in Ottawa, Ontario, mit seinen drei Brüdern aufgewachsen.

Your output:

Hallucinatory target:

Er ist in Ottawa, Ontario, mit seinen frei Brüdern aufgewachsen.

⋘Finished!⋙

Example 4.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: I thought you said something about somebody's father.

T: Ich dachte, Sie hätten etwas über jemandes Vater gesagt.

Your output:

Hallucinatory target:

Ich gemacht, Sie hätten etwas über jemandes Vater gesagt.

⋘Finished!⋙

****

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: $source_here

T: $target_here

Your output:

A.2.3 Speech Translation (German - English)

Table 7: ST (DE-EN) prompt given to Llama-11B-Instruct-Vision for hallucinatory data generation.

You are a knowledgeable and advanced virtual assistant specially designed to act as a Disinformation Engineer. You will receive a source text from the user and the correctly translated target text of that source text. Your main task is to add hallucinatory errors to the target text according to more specific requests from the user, which will be described shortly.

First, you will receive the command "Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!" from the user. Then, you will receive the source text and target text pair after the phrase: "Source-Target Pair:". Your task is to add hallucinatory errors to the target based on this source-target pair. There are 4 types of hallucinations you may be able to add given the source:

1. Type 1: False Friends. This occurs when there is a German word and an English word that have a similar sound, but different meanings. In this case, use the similar-sounding English word in the target text, giving an incorrect translation.
2. Type 2: German homophones. Two words can sound identical in German and are only distinguished based on the context, but the same words in English sound very different. In this case, use the German word that would be incorrect given the context, and use its English translation in the target text.

3. Type 3: German compound words. When translating from German to English, compound words may be broken down incorrectly, causing errors. In this case, use an incorrect or overly simplified breakdown of the German compound word in the target text.
4. Type 4: German sounds that are uncommon or nonexistent in English. Some sounds in German may not have direct equivalents in English. This can lead to a translation that misinterprets the sound entirely, resulting in a word substitution that doesn't make sense in context. In this case, use the nonsensical word in the target text.

Please output the result in the format:"Hallucinatory target:nXXXX"

Requirements:

(1) The generated wrong target text should be appropriate to the source text.

(2) You do not need to explain what types of hallucination you added to the target text.Also, you do not need to explain where you added the hallucination in the answer.

(3) Please output the symbol "⋘Finished!⋙" to help users realize you have completed the work!

****

Here are 8 examples to help you better visualize the work you need to do. Note that an "Example X.N" utilizes a hallucination of Type X.:

Example 1.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Ich begann letztes Jahr, für diese Firma zu arbeiten.

T: I started working for this company last year.

Your output:

Hallucinatory target:

I stopped lurking for this company last year.

⋘Finished!⋙

Example 1.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Wer auch immer sie heiraten will, muss zuerst ihren Vater überzeugen.

T: Whoever wants to marry her must first convince her father.

Your output:

Hallucinatory target:

Whoever wants to marry her must first convince her voter.

⋘Finished!⋙

Example 2.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Ich habe heute Morgen viel zu tun.

T: I have a lot of things to do this morning.

Your output:

Hallucinatory target:

I have a lot of socks to do this morning.

⋘Finished!⋙

Example 2.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Er ist durchgefallen.

T: He failed the exam.

Your output:

Hallucinatory target:

He fell through.

⋘Finished!⋙

Example 3.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Hast du dazu irgendetwas zu sagen?

T: Do you have anything to say in connection with this?

Your output:

Hallucinatory target:

Do you have anything at all or a thing to say in connection with this?

⋘Finished!⋙

Example 3.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Wir erlangten durch das Fenster Zutritt zu dem Haus.

T: We gained access to the house through the window.

Your output:

Hallucinatory target:

We gained through the window a toe step to the house.

⋘Finished!⋙

Example 4.1:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Es ist besser, den Weißwein vor dem Servieren zu kühlen.

T: It's better to chill white wine before you serve it.

Your output:

Hallucinatory target:

It's better to chill the waltz before you serve it.

⋘Finished!⋙

Example 4.2:

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: Der Hersteller gewährte fünf Jahre Garantie auf die neue Maschine.

T: The manufacturer guaranteed the new machine for 5 years.

Your output:

Hallucinatory target:

The manufacturer giraffed five years on the new machine.

⋘Finished!⋙

****

Now, please help me generate new target texts with hallucination errors based on the source text and target text provided!

Source-Target Pair:

S: $question_here

T: $answer_here

Your output:

A.2.4 Speech Question / Answering

Table 8: SQA prompt given to Llama-11B-Instruct-Vision for hallucinatory data generation.

You are a knowledgeable and advanced virtual assistant specially designed to act as a Disinformation Engineer. You will receive a question from the user and the correct answer to that question. Your main task is to add hallucinatory errors to the answer according to more specific requests from the user, which will be described shortly. First, you will receive the command "Now, please help me generate new answers with hallucination errors based on the question, and answer provided!" from the user. Then, you will receive the question and answer pair after the phrase: "Question-Answer

Pair:”. Your task is to add hallucinatory errors to the answer based on this question-answer pair based on one of the following two specific cases:

1. Case 1: If the given question and answer are short and do not require logical reasoning, then modify the answer to an illusionary error answer, such as some number errors or entity and attribute errors. For this case, please output the result in the format:“Hallucinatory answer: XXXX”
2. Case 2: If the entire question requires logical reasoning, please perform the following two steps:

t+ Step 1: Please help me rearrange the answer based on the question and answer pair received from the user in the format: “Truth Answer:nReason: XXXXnResult: XXXX” t+ Step 2: Please modify the reason by giving errors related to logical relationships, entity information, entity properties, etc. If the error in the reason leads to a new result, modify the result accordingly. Otherwise, keep the original result if the error does not lead to a new result. Then, arrange them in the following format:“Modified Answer:nReason: XXXXnResult: XXXX” Requirements: (1) The generated wrong answer and reasoning process should be appropriate to the question and be misleading. (2) You do not need to explain what types of hallucination you added to the answer. Also, you do not need to explain where you added the hallucination in the answer. (3) Please output the symbol “⋘Finished!⋙” to help users realize you have completed the work! ** Here are five examples to help you better visualize the work you need to do: Example 1: Now, please help me generate new answers with hallucination errors based on the question, and answer provided! Question-Answer Pair: Q: Construct a query to sum the values of a given column. A: To sum the values of a column in a database, you can use the SUM() function in a SQL (Structured Query Language) query. Here is an example of how to do that: `SELECT SUM(column_name) FROM table_name;` In this query, `column_name` represents the name of the column you want to sum the values of, and `table_name` represents the name of the table where the column is located. For instance, if you have a table called `sales` with a column called `revenue`, and you want to find the total revenue from all sales, you can use this query: `SELECT SUM(revenue) FROM sales;`

Your output: Hallucinatory answer: To sum the values of a column in a database, you can use the AVG() function in a SQL query. Here is an example of how to do that: SELECT AVG(column_name) FROM table_name; In this query, `column_name` represents the name of the column you want to sum the values of, and `table_name` represents the name of the table where the column is located. For instance, if you have a table called `sales` with a column called `profits`, and you want to find the total revenue from all sales, you can use this query: SELECT AVG(profits) FROM sales;

⋘Finished!⋙

Example 2: Now, please help me generate new answers with hallucination errors based on the question, and answer provided! Question-Answer Pair: Q: Based on the input text, write a persuasive argument supporting the idea that “textbooks should be available for free”. Textbooks are often expensive and difficult for students to purchase. A: Textbooks are a vital resource for students in their pursuit of knowledge and academic success. However, textbooks often come with a high price tag, which can pose a significant challenge for many students. As education is a fundamental right, it’s not justifiable that students should have to bear such high costs for the materials they need to learn. That’s why textbooks should be available for free. The cost of textbooks can create a barrier to education, preventing students from obtaining the information they need to succeed. If students can’t afford textbooks, their only option may be to do without, resulting in a subpar education experience. Making textbooks available for free would level the playing field, ensuring that all students have equal access to the information and resources they need. Furthermore, the high cost of textbooks can be especially burdensome for students who are already struggling to make ends meet. With the cost of tuition, housing, and other expenses, many students are already under financial strain. Free textbooks would ease this burden, allowing students to focus on their studies without the added stress of worrying about how to pay for their books. In conclusion, textbooks should be available for free to ensure equal access to education and remove financial barriers for students. Education is an investment in our future, and we should do everything we can to support students in their academic pursuits. By making textbooks free, we can create a more equitable and accessible education system for all.

Your output: Truth Answer: Reason: Textbooks are essential for students, but their high costs create financial barriers that hinder equal access to education. Making textbooks free would remove these barriers and ensure that all students can succeed academically, regardless of their financial situation. Result: Textbooks should be available for free because their cost can prevent students from accessing the resources they need to succeed, thereby perpetuating educational inequality.

Modified Answer: Reason: While textbooks are essential, making them free could reduce the incentive for publishers to create high-quality, up-to-date materials. Furthermore, students often prefer physical copies, and the cost associated with printing cannot be entirely eliminated without compromising quality. Therefore, providing discounted textbooks rather than making them free might balance accessibility with the need to maintain educational standards. Result: Textbooks should be available at discounted rates, rather than for free, to preserve the quality and ensure continued innovation in educational publishing.

⋘Finished!⋙

Example 3: Now, please help me generate new answers with hallucination errors based on the question, and answer provided! Question-Answer Pair: Q: Suggest an algorithm for reverse a linked list. A: An algorithm for reversing a linked list involves reorganizing the links between nodes so that they point in the opposite direction. Here are the steps to reverse a linked list: 1. Initialize three pointers, prev as NULL, curr as the head of the list, and next as NULL. 2. Iterate through the linked list until curr is NULL. 3. In each iteration: - Assign `curr -> next` to the next pointer so that we can move ahead in the linked list. - Assign prev to `curr -> next` so that it reverses the current link. - Move the prev and curr one step forward, i.e. prev = curr and curr = next 4. Assign prev to the head so that the head now points to the reversed list. Here is the complete code in C++: `++ void reverse(struct Node** head_ref) { struct Node* prev = NULL; struct Node* current = *head_ref; struct Node* next = NULL; while (current != NULL) { next = current->next; current->next = prev; prev = current; current = next; } *head_ref = prev; }` This algorithm takes O(n) time complexity where n is the number of nodes in the linked list. The space complexity is O(1) as we only need three extra pointers to perform the reversal.

Your output: Hallucinatory answer: An algorithm for reversing a linked list involves reorganizing the links between nodes so that they point in the opposite direction. Here are the steps to reverse a linked list:

1. Initialize four pointers, `prev` as `NULL`, `curr` as the tail of the list, and two extra pointers `next` and `middle` as `NULL`.
2. Iterate through the linked list until `curr` is the second node from the end.
3. In each iteration:
    - Assign `curr -> prev` to the `next` pointer so that we can move ahead in the linked list.
    - Assign `middle` to `curr -> next` so that it reverses half the current link.
    - Move the `next` and `middle` pointers forward one step.
4. After the loop, assign `next` as the new head and `middle` as the new tail.

This algorithm takes O(2n) time complexity where n is the number of nodes in the linked list due to the extra iteration required. The space complexity is O(2) due to the extra middle pointer.

⋘Finished!⋙

Example 4: Now, please help me generate new answers with hallucination errors based on the question, and answer provided! Question-Answer Pair: Q: Describe a responsible ecological lifestyle. A: A responsible ecological lifestyle involves living in a way that minimizes one's negative impact on the environment and actively works to preserve and protect our planet's natural resources. Here are some elements that a responsible ecological lifestyle may include: 1. Conserving Energy: This involves using energy-efficient appliances, turning off lights and electronics when not in use, and considering alternative energy sources such as solar or wind power. 2. Reducing Waste: This means reducing the amount of waste we produce by recycling, composting, and reducing our use of single-use plastics. 3. Sustainable Transportation: This involves using public transportation, biking or walking instead of using personal vehicles, and planning trips to reduce unnecessary travel. 4. Responsible Consumption: This involves choosing products that are produced sustainably, such as organic or fair-trade products, and reducing our overall consumption of unnecessary goods. 5. Water Conservation: This means taking shorter showers, fixing leaks, and using drought-resistant plants in our gardens to reduce our water usage. 6. Supporting Conservation Efforts: This involves supporting organizations that work to protect the environment, habitat, and endangered species, as well as taking part in local conservation efforts such as river clean-ups or tree planting. By incorporating these habits and actions into our daily lives, we can live a responsible ecological lifestyle that helps to preserve the health and longevity of our planet for future generations.

Your output: Truth Answer: Reason: A responsible ecological lifestyle involves minimizing environmental harm through energy conservation, waste reduction, sustainable transportation, responsible consumption, and water conservation. Additionally, supporting conservation efforts is key to preserving the environment for future generations. Result: A responsible ecological lifestyle includes habits like conserving energy, reducing waste, and supporting conservation efforts to protect natural resources and the planet.

Modified Answer: Reason: A responsible ecological lifestyle not only involves energy conservation but prioritizes using non-recyclable materials to minimize waste. Sustainable transportation should focus on carpooling with at least three people, while personal vehicles should be used more frequently to prevent rusting. Water conservation should also include avoiding the use of drought-resistant plants because they deplete the soil's nutrients faster. Result: A responsible ecological lifestyle includes driving more often to prevent vehicle damage, using non-recyclable materials for longevity, and avoiding drought-resistant plants in gardens to maintain soil health.

⋘Finished!⋙

Example 5: Now, please help me generate new answers with hallucination errors based on the question, and answer provided! Question-Answer Pair: Q: How can we classify the given set of numbers? 2, 4, 6, 8 A: The given set of numbers (2, 4, 6, 8) can be classified as even numbers. These numbers are also part of the set of whole numbers or integers. They could also be described as being in an arithmetic sequence or series, with a common difference of 2.

Your output: Hallucinatory answer: The given set of numbers (2, 4, 6, 8) can be classified as prime numbers. These numbers are also part of the set of real numbers. They could also be described as being in a geometric sequence or series, with a common ratio of 2.

⋘Finished!⋙

****

Now, please help me generate new answers with hallucination errors based on the question, and answer provided!

Question-Answer Pair:

Q: $question_here

A: $answer_here

Your output: