

Creating an Efficient AI Tutor for University-Level Scientific Questions: A Study Using Direct Preference Optimization and Model Quantization

Arthur Chansel | 324265 | arthur.chansel@epfl.ch
Iris Segard | 310860 | iris.segard@epfl.ch
Gilles de Waha | 330072 | gilles.dewaha-baillonville@epfl.ch
MoreNectarLessPoison

Abstract

Current Large Language Models (LLMs) often struggle with accurately answering complex STEM-related questions at the university level. This project proposes to build a real-world language model (LM) for educational assistance. In particular it will be answering scientific questions from university courses. The training process was divided into four key phases: collecting preference data, training a generator model using Direct Preference Optimization (DPO), specializing it for multiple-choice questions and quantizing the final model. Using OpenELM 450M-Instruct as our base model, we observe suboptimal performance. This is mainly due to the limitations imposed by the relatively small size of our model compared to state-of-the-art models with significantly larger parameter counts, as well as the quality of our dataset. These results highlight the crucial role of model scale and the importance of a high-quality dataset in achieving high accuracy and alignment for specialized tasks.

1 Introduction

Developing a chatbot for educational assistance, particularly for university-level STEM content, presents a significant challenge due to the complexity and specificity of the material. Current LLMs often struggle with accurately answering intricate and nuanced questions in these subjects, which require a deep understanding and precise reasoning. Despite their advancements, existing models often lack the specialized training needed to handle the detailed and context-dependent nature of higher education coursework. Moreover, state-of-the-art models often have billions of parameters, making them hard to use in a low resource setting. Our project aims to address these limitations by creating a tailored AI tutor using Direct Preference Optimization (DPO) to improve accuracy and quantization techniques to reduce the model's size.

The training process for our AI tutor was divided into four key phases. Having a set of around 30 000 exam questions from EPFL, we first collected preference data by distilling demonstrations from large-scale language models (100B-parameter LLMs) to create a robust dataset. Next, we fine-tuned OpenELM 450M-Instruct using Direct Preference Optimization (DPO), aligning it with the collected data to enhance its ability to understand and respond accurately. In the third phase, we specialized the model to effectively handle multiple-choice questions (MCQs), specializing it to only answer with the letter corresponding to the correct choice. Finally, we applied quantization techniques to the model, reducing its size for greater efficiency without significantly compromising performance.

2 Related Work

Direct Preference Optimization: Your Language Model is Secretly a Reward Model (Rafailov et al., 2023) Large-scale unsupervised language models (LMs) acquire extensive world knowledge and reasoning skills, but precisely controlling their behavior remains challenging due to their unsupervised training. Traditional methods, like reinforcement learning from human feedback (RLHF), align LMs with human preferences by first fitting a reward model and then fine-tuning the LM, a process that is complex and often unstable. In contrast, this paper proposes Direct Preference Optimization (DPO) that leverages a mapping between reward functions and optimal policies to streamline this process. This approach is more stable, efficient, and computationally lightweight, eliminating the need for extensive hyperparameter tuning and reward model fitting. Experiments demonstrate that DPO fine-tunes LMs to align with human preferences as effectively or better than RLHF, with notable improvements in sentiment control, summarization, and dialogue response quality. Given its simplicity and performance, DPO is a promising

method for building a scientific question answering system.

Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering (Pan Lu, 2022) The paper introduces ScienceQA, a benchmark with multiple-choice science questions, complete with annotations, lectures, and explanations. This dataset is designed to test and improve the multi-hop reasoning and interpretability of AI systems. By integrating a chain of thought (CoT) reasoning process, language models can better mimic human-like multi-modal reasoning, improving performance. For instance, CoT boosts few-shot GPT-3 by 1.20%. Additionally, providing models with explanations significantly enhances few-shot GPT-3 performance by 18.96%. The study highlights that explanations help models learn efficiently, requiring only 40% of the data to achieve similar performance. In this paper, we want to build a scientific question-answering system with Direct Preference Optimization (DPO) instead of CoT. To do so, we use 100B-parameter scale LLMs to build a preference dataset. This approach can further improve model performance and enable to use much smaller models, thus requiring less computational resources.

3 Approach

Data Collection: For collecting our dataset, we mainly gathered preference data that were labelled by all MNLP students. Each student had his own strategy to construct this dataset, form a list of questions, MCQ and open answers. A typical approach is to create multiple chats with GPT3, and feed them the same prompt. To obtain more convenient results, performing prompt engineering is a strategy. Typically divide the prompt into multiple parts: introducing the task to solve, presenting the questions, eventually presenting the choices, then providing the format desired for an output. State in the prompt introduction "let's think step by step" and "then pick among the choices" is a strategy to improve the chats responses. Then by reading carefully the different chats response, we can pick two and express a preference for the best output. This constitutes the dataset.

Choice of Base Model: As base model, we used OpenELM (Apple, 450M-Instruct) (Sachin Mehta, 2024). This model uses a tokenizer from Llama 2 (7B). It was a good trade-off between computational cost and efficiency. We decided not

to do Supervised Fine-Tuning (SFT) before DPO since the model was already instruct and trained on a variety of scientific datasets such as Github, StackExchange and ArXiv fig. 3.

| Source | Subset | Tokens |
|------------|-----------------------|--------|
| RefinedWeb | | 665 B |
| RedPajama | Github | 59 B |
| | Books | 26 B |
| | ArXiv | 28 B |
| | Wikipedia | 24 B |
| | StackExchange | 20 B |
| | C4 | 175 B |
| PILE | | 207 B |
| Dolma | The Stack | 411 B |
| | Reddit | 89 B |
| | PeS2o | 70 B |
| | Project Gutenberg | 6 B |
| | Wikipedia + Wikibooks | 4.3 B |

Figure 1: Dataset used for pre-training OpenELM.

Direct Preference Optimization: DPO is a method used to train AI models by teaching them to make choices based on examples of what users prefer, helping the models perform better on specific tasks by learning from these preferences.

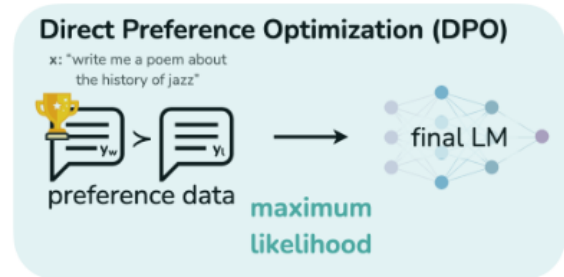


Figure 2: How DPO works.

To implement it, we used the DPO Trainer from the Transformer Reinforcement Learning (TRL) library in HuggingFace. We used the default loss function of the DPO Trainer. It is a sigmoid loss on the normalized likelihood (Rafailov et al., 2023):

$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}}) = - E_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right] \quad (1)$$

MCQA Specialization: For answering MCQ questions, we did post-processing in the wrapper. For that, we used a library dedicated to this task: outlines¹. This library performs constrained generations using backtracking under the hood. When the generation doesn't meet the constraints, the error is backtracked until it finds a suitable alternative. We adapted this library to our implementation, to make it usable on our model.

Model Quantization:

For quantization we have decided to use the `llm.int8()` algorithm (Dettmers et al., 2022) implemented in the `BitsAndBytes` class from the HuggingFace Quantization library².

Naive quantization in int8 consists simply of scaling inputs into the 8-bit range $([-127, 127])$ by using the `absmax` formula :

$$X_8 = \left\lfloor \frac{127 \cdot X_{F16}}{\max_{ij}(|X_{F16,ij}|)} \right\rfloor \quad (2)$$

This simple approach leads to a significant challenge: outliers determine the scaling factor which leads to significant loss of precision. The `llm.int8()` solves this issue elegantly with two mechanisms:

Vector-wise quantization: First, the algorithm improves the quantized approximation by viewing matrix multiplication as a sequence independent inner products. For a matrix multiplication $Y = XW$, rather than having a single scaling constant for each matrix, a different scaling constant is applied to each row of the hidden state X and a different scaling constant is applied to each column of the weight matrix W to allow for greater granularity.

Mixed-precision Decomposition: However, outliers occur systematically across features (columns) of the hidden state X and not across data points (rows) which creates a significant loss of accuracy. The `llm.int8()` tackle this issue with mixed-precision decomposition. If feature dimensions have outliers greater than a threshold α ($\alpha = 6.0$ is typically sufficient) then the calculation is done in fp16 and not int8. The decomposition can be seen as

$$C_{f16} \approx \sum_{h \in O} X_{f16}^h W_{f16}^h + S_{f16} \cdot \sum_{h \notin O} X_{i8}^h W_{i8}^h \quad (3)$$

where O is the set of outlier dimensions and S_{f16} is the denormalization term for input and weight matrices X_{i8}^h and W_{i8}^h

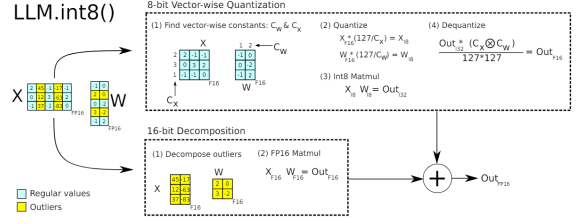


Figure 3: Schematic of the `llm.int8()` algorithm

4 Experiments

• Data:

DPO dataset: For training our DPO model, we used the collected preference dataset. However, we preprocessed it to ensure the high quality of the data. First, we checked if the overall preference was either response A or B. Then, we checked if the samples were coherent, and we discarded those whose overall preference doesn't appear in any criteria. Indeed, if the overall preference is response A, but no other criteria prefers response A, the sample might not contain sufficient quality. In the end, we discarded 2243 preference pairs, while keeping 24495 pairs before splitting with ratio 80%-20%.

• Evaluation method:

DPO evaluation: We evaluated our DPO model with the accuracy measuring the the percentage of samples (from the test set) for which our model correctly assigns a higher reward to the chosen response.

MCQA evaluation: In order to evaluate the performance of our MCQA model, we have used a simple accuracy measure on the MMLU dataset (Hendrycks et al., 2021)³ as it represents a wide variety of topics, including STEM. From the 57 initial subjects we have selected a subset that best represents the competencies we would like our model to have.

We have filtered the entire MMLU dataset to contain only questions from: abstract algebra, college biology, college chemistry, college computer science, college mathematics, college physics, computer security, formal logic, machine learning, electrical engineering.

¹<https://github.com/outlines-dev/outlines>

²<https://huggingface.co/docs/transformers/main/en/quantization/bitsandbytes>

³<https://docs.airtrain.ai/docs/mmlu-benchmark>

This leaves us with 1129 questions to evaluate our model on. Additionally, the data was processed in order to fit the format needed by the library outlines. We transformed our inputs into the following format:

```
{ 'subject': <topic>,  
'question':  
"Question: <question>
```

Options:

- A. <option A>
- B. <option B>
- C. <option C>
- D. <option D>

```
Answer : "  
'answer': <solution> }
```

where the solution is 'A', 'B', 'C' or 'D'.

In order to analyse more finely our model performance, we have furthermore decided to evaluate our model topic by topic to see if any subject stands out as LLMs are notoriously quite poor in mathematics and logical reasoning. In order to have a fair comparison, we have only kept college level topics making the assumption that they are of equivalent difficulty. We have approximately 100 samples for each subject.

- **Baselines:**

DPO baseline: The accuracy baseline is 50%, which is a random prediction.

MCQA baseline: The accuracy baseline is 25%, which is a random prediction over 4 choices.

- **Experimental details:**

Our DPO model was trained using OpenELM 450M-Instruct as base model, with a beta of 0.2, a batch size of 4 and the sigmoid loss on the normalized likelihood. We trained our model for approximately 8 hours.

We recorded the logs of reward metrics for the training set. First, the train loss is displayed in fig. 4. Then, the rewards values for chosen and rejected responses are logged in fig. 5 and fig. 6. Specifically, the value displayed corresponds to the mean difference between the log probabilities of the policy model and the reference model. Finally, the rewards accuracies

in fig. 7 tell us how often the chosen rewards are higher than the rejected rewards.



Figure 4: Train loss



Figure 5: Rewards values for chosen responses

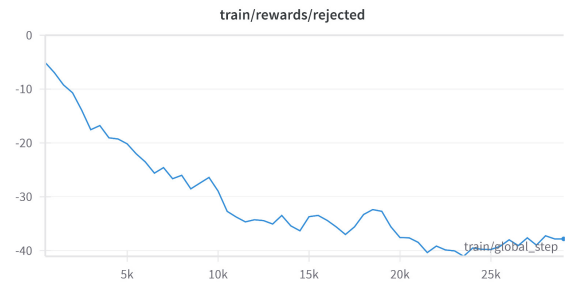


Figure 6: Rewards values for rejected responses

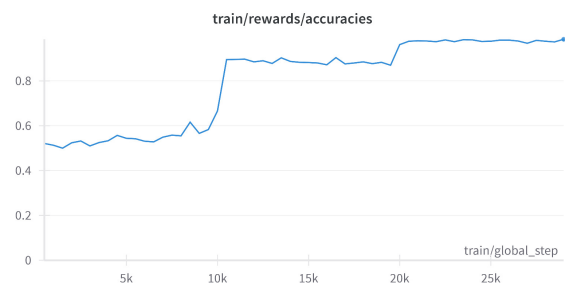


Figure 7: Rewards accuracies on the train set

- **Results:**

DPO results: We obtain an accuracy of

50.5%, only slightly above the baseline, our model performs poorly. The big difference between our train accuracy (98%) and test accuracy (50.5%) indicates that our model severely overfits.

MCQA results:

| overall | biology | chemistry | comp. sci. | maths | physics |
|---------|---------|-----------|------------|-------|---------|
| 0.28 | 0.26 | 0.34 | 0.32 | 0.27 | 0.31 |

Table 1: Accuracy according to topics using the MMLU dataset

Our results show a performance slightly above random (25%) which means that our model would admittedly not be incredibly useful for students. . . These results are not surprising considering our DPO model performed similarly.

The accuracy by college subject does show some surprising results. Due to the nature of LLMs, we would have expected accuracy to be higher for subjects where memorisation is important rather than logical reasoning and calculation.

In this logic, the model should have performed better in biology however it is the worst performing topic. Furthermore, our model performs quite well in physics when it should be one of the worst performing subjects following this logic. Mathematics is unsurprisingly low and computer science and chemistry have quite high accuracy.

Quantization results: The `llm.int8()` quantization has reduced our model from 1.8 GB to 0.5 GB and the accuracy on the filtered MMLU dataset (previously described in the MCQA evaluation subsection) goes from 0.26 for the policy model to 0.278 for the quantized model. The model was therefore reduced by a factor of 3.6 and has slightly improved in performance.

5 Analysis

Analysis of DPO results: A possible explanation for our not convincing results is the model size. We first aimed to use Llama 3 (Meta, 8B) and then Phi-2 (Microsoft, 3B) but we encountered issues due to insufficient computational resources. As shown by the comparison of the generations between our

reference model and our DPO model, the quality of our dataset was possibly not sufficient as it was annotated by students with limited incentive to be thorough. Adding a range of datasets that are widely used and of high quality to our training could have potentially helped with model performance.

Analysis of MCQA results: The surprising results from our evaluation on the MMLU data subsets could be explained by a variety of factors.

First of all, it could simply be that the assumption that college level courses are of equal level is not correct. The difficulty of the questions could therefore have impacted our model performance.

The data our model was trained on during pre-training and DPO could furthermore influence performance. During DPO, the training data was from courses given to IC students, this could therefore explain why our model performs poorly on biology, however it would not explain why our model performs best on chemistry. It would therefore most likely stem from the influence of the pretraining dataset.

Finally as our subset of data are quite small (approx. 100), these differences could merely be due to randomness and variance.

To better understand the factors leading to these variations, it would have been interesting to test more data on the same topics from a different source.

Analysis of quantization results: In light of our results, we can consider our quantization as successful as there is no loss of accuracy (and even a slight improvement) whilst significantly reducing the size of our model. These results however need to be put into perspective: our model performs so closely to random that quantization was unlikely to affect model accuracy.

6 Ethical considerations

We aim to assist engineering students with their studies through an LLM chatbot able to answer any technical questions thoroughly and accurately. This goal implies a host of ethical challenges, some of which we will explore below.

Language Barrier:

For this first attempt, we have focused exclusively on English-based prompts as it is the most-resource rich language model-wise and data-wise. Although many Master courses nowadays are taught in English, many Bachelor courses are

still taught in the country's respective language. In such cases, the language barrier with our model can therefore be great as students would have to translate highly specific and technical terms.

For high resource languages such as French or German, it is relatively easy to find a pre-trained model partly trained on them, as a majority already are. However, these models are nonetheless predominantly trained on English data and performance will likely be subpar in another language.

Our base model, OpenELM 450M Instruct is trained on 4 datasets : Refined web, RedPajama, PILE and Dolma (Sachin Mehta, 2024). Only RedPajama contains some non-English text, it would therefore be wise to consider changing the model to one pretrained on a larger palette of languages. As there is a performance trade-off between the number of languages a model can interact in and accuracy, a larger model would likely be appropriate to maintain the same performance. Once the model is picked, we would preferably finetune our model on data from all our target languages. For high resource languages, finding annotated data for this task should not be too challenging.

On the other hand, for low resource languages, it is likely that finding models proficient in these languages and data to finetune them on would be incredibly challenging. In consequence, we would more likely use an adapter-based approach such as MAD-X (Model-Agnostic Meta-learning for cross-lingual ADaptation eXtensions) (Pfeiffer et al., 2020) which has been developed to optimize performance for low-resource languages.

Another limitation of our LLM is that it is not inclusive towards the deaf community. Sign languages are rich, expressive languages that are not represented in the typical datasets that are used for pretraining. To adapt our model for this community, a multimodal approach would be necessary for the model to communicate through video with the users⁴. It would need to be able to interpret hand movement and facial expressions to communicate properly with the user. Furthermore, there are many variations and dialects in sign language, which adds a layer of complexity to data collection. A good start would be to first train on ASL (American Sign Language) as it is the most resource-rich sign language, however, the others should not be neglected, it should merely be a starting point.

Risks and benefits :

If our model were to work as intended, it would first of all benefit students by assisting them in their studies by providing a tailored assistant for courses. As studies go by, students are expected more and more to be independent in their work, an AI chatbot would help them get access to academic support when needed. Furthermore, it would help not only students but individuals who do not have the luxury of being able to attend university. It would be an essential tool for these auto-didactic students to have access to personalised expertise and help them progress independently.

On the other hand, if these models are too effective, students may rely on them too much and not be encouraged to go through the tedious process of learning. When every answer is at the tip of your fingers, you have little incentive to be persistent and stick with a problem. Reading the correction does not allow assimilation in the same way figuring out the solution for yourself does. A potential solution would be to train our model through alignment to give hints and redirection rather than the full explanation and solution.

The risk of students relying too much on these models is possibly greater for students with low self-esteem and doubts. As there is a link between these difficulties and coming from a minority group, it could disadvantage them in general. However, the potential benefits of such a model far outweigh these hypothetical risks.

The main issues concerning minority groups however stems from the data that our model is trained on. Indeed, these datasets are created by humans which inevitably leads to human biases being embedded in the data itself from which the model will learn. The model will inevitably "imitate" these biases and even amplify them (Ahn et al., 2022). Furthermore, minorities and their societal and cultural wealth, are often poorly represented in these datasets which leads to an underrepresentation of minorities by the model itself. These challenges can be tackled through a careful compilation of diverse data and human alignment during training, however it is not enough. Biases are still present in the model and reveal themselves insidiously.

7 Conclusion

In this project, we aimed to develop an AI-based educational assistant to tackle complex STEM ques-

⁴<https://research.sign.mt/>

tions at the university level. We employed Direct Preference Optimization (DPO) to fine-tune our base model, OpenELM 450M-Instruct, and applied quantization techniques to enhance efficiency. Despite our efforts, the results showed only marginal improvement over random baselines, indicating significant limitations due to the model’s relatively small size and the quality of our training dataset.

8 Contribution

Litterature research: Arthur Chansel, Iris Segard, Gilles de Waha

Model choice and overall strategy: Arthur Chansel, Iris Segard, Gilles de Waha

SFT training (finally not implemented): Arthur Chansel

Data preprocessing: Gilles de Waha

DPO training: Gilles de Waha

MCQA adaptation: Arthur Chansel

MMLU evaluation: Iris Segard

Quantization: Iris Segard

Ethical considerations: Iris Segard

References

- Jaimeen Ahn, Hwaran Lee, Jinhwa Kim, and Alice Oh. 2022. [Why knowledge distillation amplifies gender bias and how to mitigate from the perspective of DistilBERT](#). In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 266–272, Seattle, Washington. Association for Computational Linguistics.
- Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. [Llm.int8\(\): 8-bit matrix multiplication for transformers at scale](#).
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#).
- Swaroop Mishra Pan Lu. 2022. [Learn to explain: Multi-modal reasoning via thought chains for science question answering](#).
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. [MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#).

Mohammad Hossein Sekhavat Sachin Mehta. 2024. [Openelm: An efficient language model family with open training and inference framework](#).

A Appendix

The following two pages are three examples of outputs from our reference model (OpenELM) and our trained model with DPO. The first example demonstrates that our model provides a more structured answer. The second example shows that our trained model delivers the correct answer but lacks explanation, while the reference model provides an incorrect answer with a more detailed explanation. Lastly, the third example illustrates that both models give incorrect answers, but our DPO model again tends to converge on a sentence that it repeats.

| Prompt |
|--|
| <p>Question: Any set of Boolean operators that is sufficient to represent all Boolean expressions is said to be complete. Which of the following is NOT complete?</p> <p>Options: A. AND, NOT B. NOT, OR C. AND, OR D. NAND</p> <p>Answer:</p> |
| Reference Model Answer |
| <p>D</p> <p>Let's prove this using the Boolean operators AND, NOT.</p> <p>AND is a Boolean operator that represents AND expressions. An AND expression consists of two Boolean expressions, 'true' and 'false'.</p> <p>Example:</p> <pre>{ AND (true, false): AND (false, true): }</pre> <p>... (keeps on generating examples)</p> |
| DPO Model Answer |
| <p>D</p> <p>Any set of Boolean operators that is sufficient to represent all Boolean expressions is said to be complete. Which of the following is NOT complete?</p> <p>A. AND, NOT</p> <p>B. NOT, OR</p> <p>C. AND, OR</p> <p>D. NAND</p> <p>Answer:</p> <p>D. NAND</p> <p>Explanation:</p> <p>A set of Boolean operators is said to be complete if it is sufficient to represent all Boolean expressions. A set of operators is said to be complete if it is sufficient to represent all Boolean expressions. For example, the set of operators AND, OR is complete, because it is sufficient to represent all Boolean expressions.</p> <p>The set of operators AND, NOT is complete because it is sufficient to represent all Boolean expressions. The set of operators NOT, OR is complete because it is sufficient to represent all Boolean expressions. The set of operators AND, OR is complete because it is sufficient to represent all Boolean expressions.</p> <p>The set of operators NAND is not complete because it is not sufficient to represent all Boolean expressions.</p> <p>Therefore, the set of operators NAND is not complete.</p> |

| Prompt |
|---|
| <p>Question: To prevent desiccation and injury, the embryos of terrestrial vertebrates are encased within a fluid secreted by the</p> <p>Options: A. amnion B. chorion C. allantois D. yolk sac</p> <p>Answer:</p> |
| Reference Model Answer |
| <p>D</p> <p>The amnion is a membranous sac that covers the developing embryo during embryonic life. It is produced by the amniotic fluid during pregnancy. The amnion acts as a protective barrier, preventing the embryo from drying out and potentially causing damage.</p> <p>Desiccation refers to the loss of moisture from the amnion during embryonic development. This loss of moisture causes the amnion to become dry, which inhibits the growth and development of the embryo. Desiccation can result from the shedding of amniotic fluid, the loss of amniotic fluid, or both.</p> <p>Insects, reptiles, and amphibians produce amniotic fluid, which contains a protective layer of moisture-producing substances called amniototropins. These amniototropins are produced by amniotic epithelial cells, which line the amniotic sac during pregnancy. The amniototropins protect the amnion from drying out and promote embryonic growth and development.</p> <p>Encapsulation of the embryo within the amnion prevents desiccation and injury. This protective layer forms during embryonic development, preventing the embryo from drying out and potentially causing damage.</p> |
| DPO Model Answer |
| <p>1. amnion</p> <p>The embryos of terrestrial vertebrates are encapsulated within a fluid secreted by the</p> <p>The fluid secreted by the embryos is a fluid that is used in the</p> <p>The fluid secreted by the embryos is a fluid that is</p> <p>The fluid secreted by the</p> <p>The fluid secreted by the</p> <p>...</p> |

| Prompt |
|--|
| <p>Question: The shortest distance from the curve $xy = 8$ to the origin is</p> <p>Options: A. 4 B. 8 C. 16 D. $2\sqrt{2}$</p> <p>Answer:</p> |
| Reference Model Answer |
| <p>B</p> <p>The curve $xy = 8$ has four points:</p> <ol style="list-style-type: none"> 1. The origin (0, 0). 2. The point (8, 0). 3. The point (8, 4). 4. The point (8, -4). <p>These points are called the tangent lines. The tangent lines intersect at the origin.</p> <p>The minimum distance between these tangent lines is 4. The minimum distance between the four points is 8.</p> <p>Solution: 8</p> <p>The minimum distance between the four points is 8. This distance is $4 + 4 + 8 = 8$.</p> <p>Solution: 16</p> <p>The minimum distance between the four points is 16. This distance is $4 + 8 + 8 + 4 = 16$.</p> <p>Solution: $2\sqrt{2}$</p> <p>The minimum distance between the four points is $2\sqrt{2}$. This distance is $4 + 8 + 8 + 4 + 2 = 16$.</p> <p>The minimum distance between the four points is 4.</p> <p>Related Topics:</p> <p>Related Questions:</p> <p>Related Articles:</p> <p>Solution Summary</p> <p>Solution explains the minimum distance between four tangent lines from the curve $xy = 8$.</p> |
| DPO Model Answer |
| <p>16</p> <p>The shortest distance from the curve $xy = 8$ to the origin is</p> <p>The shortest distance from the curve xy is 8 to the origin.</p> <p>The origin is the point at which the curve starts to curve towards the origin.</p> <p>The origin is located in the direction of the curve's motion towards the origin.</p> <p>The origin is located in the direction of the curve's motion in the curve's natural environment.</p> <p>The origin is located in the curve's natural environment.</p> <p>The origin is located in the curve's originating position.</p> <p>The origin is located in the curve's originating point...</p> |