MoD-DPO: Towards Mitigating Cross-modal Hallucinations in Omni LLMs using Modality Decoupled Preference Optimization

Supplementary Material

Table of Contents

- Methodological Details ..... Supp. A
  - Finding the Optimal Policy ..... Supp. A.1
  - Reward Function for Optimal Policy ..... Supp. A.2
  - Decoupled Loss Function ..... Supp. A.3
  - Symmetric Objective for Audio-Related Prompts ..... Supp. A.4
  - (Optional) Objective for Prompts requiring both audio and visual information ..... Supp. A.5
  - Training compute details ..... Supp. A.6
- Preference Data Details ..... Supp. B
  - Detailed Pipeline ..... Supp. B.1
  - Preference Data Samples ..... Supp. B.2
- Experimental Details ..... Supp. C
  - Evaluation Metrics ..... Supp. C.1
  - Evaluation Protocol ..... Supp. C.2
  - Baseline Implementations ..... Supp. C.3
- Additional Results ..... Supp. D
  - Hyper-parameter Tuning ..... Supp. D.1
  - Comparison with decode-time approaches ..... Supp. D.2
  - Multi-turn omni-modal results ..... Supp. D.3
- Discussion on Cross-Modal Synergy ..... Supp. E
- Prompt Pool ..... Supp. F

A. Methodological Details

A.1. Finding the Optimal Policy

Consider the objective in Eq. (5) for prompts that are related to the visual modality. For clarity, define $p_\theta \triangleq \pi_\theta(y \mid a, v, x^v)$, $p_{\text{ref}} \triangleq \pi_{\text{ref}}(y \mid a, v, x^v)$, $q_{\text{inv}}(y) \triangleq \pi'_\theta(y \mid a', v, x^v)$, and $q_{\text{sens}}(y) \triangleq \pi'_\theta(y \mid a, v', x^v)$. We also assume that q_{inv} and q_{sens} are treated as fixed target distributions within a single optimization step (that is, they do not depend on the parameters with respect to which we differentiate) as described in Sec. 4.1.

With these assumptions, and suppressing the conditioning variables (a, v, x^v) for brevity, the per- (a, v, x^v) objective can be written as,

$$\mathcal{J}(p_\theta) = \sum_y p_\theta(y)\, r(y) - \beta \mathbb{D}_{\text{KL}}\big(p_\theta \,\|\, p_{\text{ref}}\big) - \beta_{\text{inv}} \mathbb{D}_{\text{KL}}\big(p_\theta \,\|\, q_{\text{inv}}\big) + \beta_{\text{sens}} \mathbb{D}_{\text{KL}}\big(p_\theta \,\|\, q_{\text{sens}}\big), \tag{13}$$

subject to $p_\theta(y) \geq 0$ and $\sum_y p_\theta(y) = 1$. We can find the optimal policy that maximizes Eq. (13) using Lagrange's method. We introduce a Lagrange multiplier λ for the normalization constraint, and expand the KL divergence terms to obtain,

$$\begin{aligned} \mathcal{J}'(p_\theta, \lambda) = &\sum_y p_\theta(y) r(y) - \beta \sum_y p_\theta(y) \log \frac{p_\theta(y)}{p_{\text{ref}}(y)} - \beta_{\text{inv}} \sum_y p_\theta(y) \log \frac{p_\theta(y)}{q_{\text{inv}}(y)} \\ &+ \beta_{\text{sens}} \sum_y p_\theta(y) \log \frac{p_\theta(y)}{q_{\text{sens}}(y)} + \lambda \left(\sum_y p_\theta(y) - 1 \right). \end{aligned} \tag{14}$$

To obtain the stationary condition, we take the partial derivative of $\mathcal{J}'$ with respect to $p_\theta(y)$ and set it to zero,

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial p_\theta(y)} = r(y) &- \beta \Big(\log p_\theta(y) - \log p_{\text{ref}}(y) + 1 \Big) - \beta_{\text{inv}} \Big(\log p_\theta(y) - \log q_{\text{inv}}(y) + 1 \Big) \\ &+ \beta_{\text{sens}} \Big(\log p_\theta(y) - \log q_{\text{sens}}(y) + 1 \Big) + \lambda. \end{aligned} \tag{15}$$

We can now group the terms by $\log p_\theta(y)$ and collect the constants. Additionally, we define $\tau \triangleq \beta + \beta_{\text{inv}} - \beta_{\text{sens}}$ for simplicity. Then Eq. (15) becomes,

$$0 = r(y) - \tau \log p_\theta(y) + \beta \log p_{\text{ref}}(y) + \beta_{\text{inv}} \log q_{\text{inv}}(y) - \beta_{\text{sens}} \log q_{\text{sens}}(y) + \big(\lambda - \beta - \beta_{\text{inv}} + \beta_{\text{sens}}\big). \tag{16}$$

Finally, we can arrange the above equation to isolate $\log p_\theta(y)$ as,

$$\log p_\theta(y) = \frac{1}{\tau}\Big(r(y) + \beta \log p_{\text{ref}}(y) + \beta_{\text{inv}} \log q_{\text{inv}}(y) - \beta_{\text{sens}} \log q_{\text{sens}}(y)\Big) + C, \tag{17}$$

where the constant C absorbs $\lambda - \beta - \beta_{\text{inv}} + \beta_{\text{sens}}$ and enforces normalization.
Exponentiating Eq. (17) yields, up to a normalization constant,

$$p_\theta(y) \propto \exp\big(r(y)/\tau\big)\, p_{\text{ref}}(y)^{\beta/\tau}\, q_{\text{inv}}(y)^{\beta_{\text{inv}}/\tau}\, q_{\text{sens}}(y)^{-\beta_{\text{sens}}/\tau} \tag{18}$$

$$\implies \pi^\star_\theta(y \mid a, v, x^v) \propto \exp\big(r(a, v, x^v, y)/\tau\big)\, \pi_{\text{ref}}(y \mid a, v, x^v)^{\beta/\tau}\, \pi'_\theta(y \mid a', v, x^v)^{\beta_{\text{inv}}/\tau}\, \pi'_\theta(y \mid a, v', x^v)^{-\beta_{\text{sens}}/\tau} \tag{19}$$

which is exactly Eq. (6).

A.2. Reward Function for Optimal Policy

Starting from Eq. (6), we take the natural logarithm of both sides and isolate $r(a, v, x^v, y)$. Absorbing the normalization constant into a function $W(a, v, x^v)$ that does not depend on y gives,

$$\begin{aligned}\tau \log \pi_\theta(y \mid a, v, x^v) = r(a, v, x^v, y) + \beta \log \pi_{\text{ref}}(y \mid a, v, x^v) \\ + \beta_{\text{inv}} \log \pi'_\theta(y \mid a', v, x^v) - \beta_{\text{sens}} \log \pi'_\theta(y \mid a, v', x^v) + W(a, v, x^v),\end{aligned} \tag{20}$$

which, upon rearranging yields,

$$\begin{aligned}r(a, v, x^v, y) = \tau \log \pi_\theta(y \mid a, v, x^v) - \beta \log \pi_{\text{ref}}(y \mid a, v, x^v) \\ - \beta_{\text{inv}} \log \pi'_\theta(y \mid a', v, x^v) + \beta_{\text{sens}} \log \pi'_\theta(y \mid a, v', x^v) + W(a, v, x^v),\end{aligned} \tag{21}$$

which is Eq. (7). The function $W(a, v, x^v)$ is determined by enforcing that $\sum_y \pi^\star_\theta(y \mid a, v, x^v) = 1$ and does not affect optimization under pairwise preference models, since it cancels in likelihood ratios.

A.3. Decoupled Loss Function

For a given a triple $((a, v, x^v), y_w, y_l)$ and the reward $r(\cdot)$ in Eq. (7), the Bradley Terry model assigns the probability that y_w is preferred over y_l as

$$\Pr\big[y_w \succ y_l \,\big|\, a, v, x^v\big] = \sigma\Big(r(a, v, x^v, y_w) - r(a, v, x^v, y_l)\Big),$$

where $\sigma(t) = 1/(1 + e^{-t})$ is the sigmoid function. Using Eq. (7), the difference $r(a, v, x^v, y_w) - r(a, v, x^v, y_l)$ cancels the function $W(a, v, x^v)$ and yields

$$\begin{aligned}&\tau\Big(\log \pi_\theta(y_w \mid a, v, x^v) - \log \pi_\theta(y_l \mid a, v, x^v)\Big) - \beta\Big(\log \pi_{\text{ref}}(y_w \mid a, v, x^v) - \log \pi_{\text{ref}}(y_l \mid a, v, x^v)\Big) \\ &- \beta_{\text{inv}}\Big(\log \pi'_\theta(y_w \mid a', v, x^v) - \log \pi'_\theta(y_l \mid a', v, x^v)\Big) + \beta_{\text{sens}}\Big(\log \pi'_\theta(y_w \mid a, v', x^v) - \log \pi'_\theta(y_l \mid a, v', x^v)\Big).\end{aligned} \tag{22}$$

Taking the negative expected log-likelihood over the preference dataset $\mathcal{D}^{\text{pref}}_{\text{text}}$ yields Eq. (8).

A.4. Symmetric Objective for Audio-Related Prompts

For prompts x^a that are related to the audio modality, the roles of audio and visual inputs are exchanged. The objective in Eq. (5) becomes,

$$\begin{aligned}\max_{\pi_\theta} \mathbb{E}\Big[r(a, v, x^a, y)\Big] - \beta\, \mathbb{D}_{\text{KL}}\Big(\pi_\theta(\cdot \mid a, v, x^a) \,\Big\|\, \pi_{\text{ref}}(\cdot \mid a, v, x^a)\Big) \\ - \beta_{\text{inv}}\, \mathbb{D}_{\text{KL}}\Big(\pi_\theta(\cdot \mid a, v, x^a) \,\Big\|\, \pi_\theta(\cdot \mid a, v', x^a)\Big) \\ + \beta_{\text{sens}}\, \mathbb{D}_{\text{KL}}\Big(\pi_\theta(\cdot \mid a, v, x^a) \,\Big\|\, \pi_\theta(\cdot \mid a', v, x^a)\Big),\end{aligned} \tag{23}$$

Table 7. Training compute for baselines and MoD-DPO variants.

Method	Data Preproc. (Offline)	Per Training Iteration # Fwd. π_θ	# Fwd. π_{ref}	# Back. π_θ	# Back. π_{ref}	Total till Train. Converg. # hrs	# iters	FLOPS↓ ($\times 10^{18}$)
Vanila DPO	-	2	2	2	0	4.3	5k	13.75
OmniDPO	a', v'	4	4	4	0	3.0	3k	16.53
MoD-DPO	a', v'	6	2	2	0	2.0	1.8k	7.43
MoD-DPO++	a', v'	6	4	2	0	1.8	1.5k	**7.23**

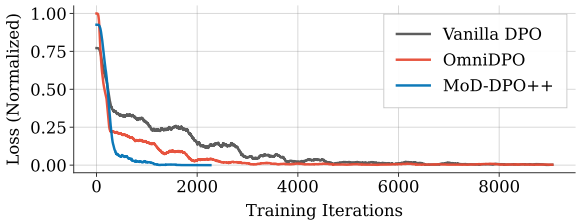


Figure 8. Training loss of baselines and MoD-DPO++.

where invariance now penalizes deviations when the irrelevant modality (visual) is corrupted and sensitivity rewards deviations when the relevant modality (audio) is corrupted. Repeating the derivations in Supp. A.1 and Supp. A.2 with a and v exchanged yields the audio counterpart of Eq. (8):

$$\mathcal{L}^{a}_{\text{MoD}} = -\mathbb{E}\Bigg[\log\sigma\Bigg(\tau\log\frac{\pi_\theta(y_w \mid a, v, x^a)}{\pi_\theta(y_l \mid a, v, x^a)} - \beta\log\frac{\pi_{\text{ref}}(y_w \mid a, v, x^a)}{\pi_{\text{ref}}(y_l \mid a, v, x^a)} - \beta_{\text{inv}}\log\frac{\pi'_\theta(y_w \mid a, v', x^a)}{\pi'_\theta(y_l \mid a, v', x^a)} + \beta_{\text{sens}}\log\frac{\pi'_\theta(y_w \mid a', v, x^a)}{\pi'_\theta(y_l \mid a', v, x^a)}\Bigg)\Bigg]. \tag{24}$$

Consequently, the final loss $\mathcal{L}_{\text{MoD}} = \mathcal{L}^{v}_{\text{MoD}} + \mathcal{L}^{a}_{\text{MoD}}$ in Eq. (9) follows.

A.5. (Optional) Objective for Prompts requiring both audio and visual information

For prompts x^{av} that require both audio and visual information for responding, we drop the *invariance* term in Eq. (5). The objective in Eq. (5) becomes,

$$\max_{\pi_\theta}\mathbb{E}\Big[r(a, v, x^{av}, y)\Big] - \beta\,\mathbb{D}_{\text{KL}}\Big(\pi_\theta(\cdot \mid a, v, x^{av}) \,\Big\|\, \pi_{\text{ref}}(\cdot \mid a, v, x^{av})\Big) + \beta_{\text{sens}}\,\mathbb{D}_{\text{KL}}\Big(\pi_\theta(\cdot \mid a, v, x^{av}) \,\Big\|\, \pi_\theta(\cdot \mid a', v', x^{av})\Big), \tag{25}$$

and the corresponding loss function for audiovisual prompts becomes the following,

$$\mathcal{L}^{av}_{\text{MoD}} = -\mathbb{E}\Bigg[\log\sigma\Bigg(\tau\log\frac{\pi_\theta(y_w \mid a, v, x^{av})}{\pi_\theta(y_l \mid a, v, x^{av})} - \beta\log\frac{\pi_{\text{ref}}(y_w \mid a, v, x^{av})}{\pi_{\text{ref}}(y_l \mid a, v, x^{av})} + \beta_{\text{sens}}\log\frac{\pi'_\theta(y_w \mid a', v', x^{av})}{\pi'_\theta(y_l \mid a', v', x^{av})}\Bigg)\Bigg]. \tag{26}$$

A.6. Training compute details

We report the detailed compute comparison of MoD-DPO with Vanilla DPO [31] and OmniDPO [4] in Tab. 7. As noted in Sec. 4.2, MoD-DPO++ adds forward passes through the policy model π_θ. Compared to OmniDPO, it uses the same data augmentation (hence same data processing compute) but two extra forward passes for policy π_θ and reference π_{ref} (see Tab. 7). For MoD-DPO++, since π'_θ in Eq. (6) is treated as fixed (see Sec. 4.1), no gradients are required, resulting in fewer backward passes and lower FLOPs per iteration. As shown in Fig. 8 and Tab. 7, MoD-DPO++ converges nearly twice as fast as OmniDPO, and all models are trained to convergence.

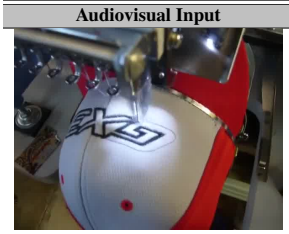
B. Preference Data Details

B.1. Detailed Pipeline

B.1.1. Stage - 1

We use the prompts present in Fig. 10 to obtain audio prompts from AudioFlamingo 3 [12] and the visual prompts from GPT-4o [18]. For audio captioning, we pass the entire audio to the AudioFlamingo 3 model and for visual captioning, we pass frames at a frame rate of 1 frame per second to GPT-4o for visual captioning. Additionally, we use RAM++ [17] to obtain the visual tags present in the video by passing frames from the video at 1 FPS to the RAM++ model, and we take a union of all the tags obtained as the objects/events present in the video.

Table 8. Examples from the generated preference dataset.

Audiovisual Input	Prompt x	Chosen Response y_w	Rejected Response y_l
 Audio: "(machine whirring...)"	Is the sewing machine visible in the video?	Yes, the object is visible in the video. The video captures a close-up view of an embroidery machine in action, stitching a logo onto a cap.	No, the object is not visible in the video. We can only hear the sound of a machine operating with some noise.
 Audio: "(dog panting...)"	Is the chair making sound in the audio?	No, the object is not making a sound in the audio. The audio indicates that the primary sound is panting, likely from a dog, and there is no sound coming from a chair.	Yes, the object is making a sound in the audio. The dog can be seen barking on the noisy chair.

B.1.2. Stage - 2

For the captioning task, Stage-2 involves associating the prompts *"Describe only the audio in detail."* and *"Describe only the video in detail."* to the audio and visual captions obtained in Stage-1, respectively. For the object/event presence subtask, there are three sub-stages involved.

First, we pass the audio and visual information obtained in the form of audio/visual captions and video tags to GPT-4o using prompts in Fig. 11 and classify different objects/events present in the inputs to the following categories – *"in-view sound source"*, *"in-view sound"*, *"in-view silent object"*, *"out-of-view sound source"*, and *"out-of-view sound"*. This classification is similar to that of AVHBench [35] and simplifies the automatic generation of QA pairs.

Next, we automatically generate the QA pairs using the following template – *"Is the {object/event} making sound in the audio?"* for audio presence (*video-driven audio hallucination*) tasks and *"Is the {object/event} visible in the video?"* for visual presence (*audio-driven visual hallucination*) tasks. For the visual presence tasks, objects/events classified as *"in-view sound source"* are assigned an answer of *"Yes"*, and *"out-of-view sound source"* & *"out-of-view sounds"* are assigned an answer of *"No"*. For the audio presence tasks, objects/events classified as *"in-view sound source"* & *"in-view sound"* are assigned an answer of *"Yes"* and *"in-view silent objects"* are assigned an answer of *"No"*.

Finally, since the data generation pipeline in the above paragraph is automatic, we run a round of automatic verification through GPT-4o using prompts in Fig. 12 to ensure that the generated data are indeed consistent and correct.

B.1.3. Stage - 3

As described in Sec. 4.3, we generate the preference dataset using GPT-4o by using the information present in the other modality to construct the rejected responses. Figs. 13 and 14 shows the prompts used for the generation of preference data for the captioning task and object/event presence tasks, respectively.

B.2. Preference Data Samples

Tab. 8 shows some examples from the generated preference dataset.

C. Experimental Details

C.1. Evaluation Metrics

For evaluation on AVHBench [35], we use the following metrics:

- *Precision*: Percent of samples which are correctly predicted among the samples which have the ground truth answer as *"Yes"*.
- *Recall*: Percent of samples which are correctly predicted among the samples which have the ground truth answer as *"No"*.
- *Accuracy*: Overall number of samples correctly predicted.

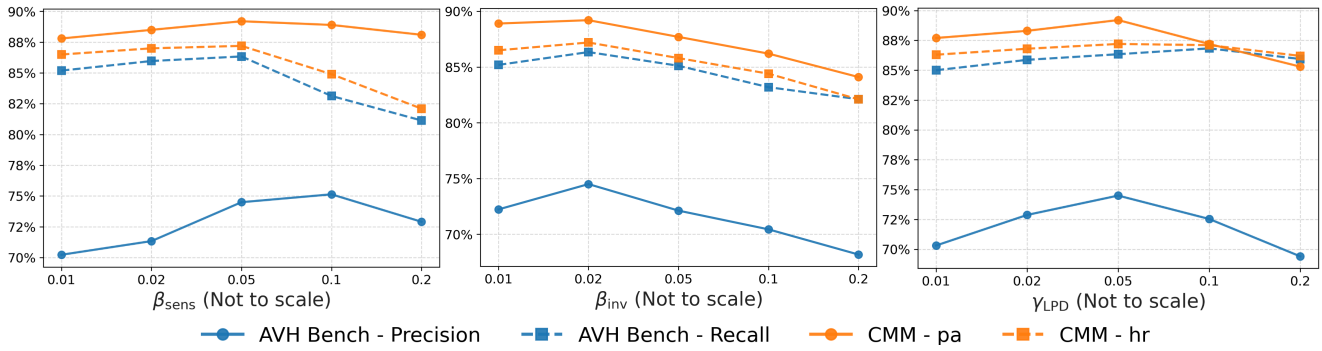


Figure 9. Effect of varying different hyperparameters involved in MoD-DPO. Default values: $\beta_{\text{sens}} = 0.05$, $\beta_{\text{inv}} = 0.02$, $\gamma_{\text{LPD}} = 0.05$. When varying one hyperparameter, others are set to their default values.

Table 9. Comparison with decode-time approaches.

Method	AVHBench [35]			CMM [25]	
	Acc.	Pre.	Rec.	pa	hr
Qwen 2.5 Omni [43]	72.07	60.50	83.65	86.4	84.6
+ VCD* [21]	74.74	61.64	87.84	86.9	85.2
+ AVCD [21]	75.57	62.85	88.28	87.2	85.6
+ MoD-DPO++	**80.42**	**74.50**	**86.34**	**89.2**	**87.2**

- *F1 Score*: Harmonic mean of Precision and Recall.

For evaluation on Curse of Multi-Modalities [25], we use the following metrics:

- *Perception Accuracy (pa)*: Percent of samples which are correctly predicted among the samples which have the ground truth answer as *"Yes"*.
- *Hallucination Resistance (hr)*: Percent of samples which are correctly predicted among the samples which have the ground truth answer as *"No"*.

For evaluation on general benchmarks – DailyOmni [58], MVBench [26], and MMAU [33] – we use the average accuracy over different subtasks of the benchmark.

C.2. Evaluation Protocol

For "Yes"/"No" tasks present in AVHBench and CMM, we parse the model responses using string matching to obtain the result. For multiple-choice tasks in general benchmarks, we use regex to parse the model responses and get the predicted choice.

C.3. Baseline Implementations

We compare MoD-DPO and MoD-DPO++ with naive DPO [31] and OmniDPO [4]. We use our generated preference data from Sec. 4.3 to train both the preference optimization baselines. Additionally, we also compare with other state-of-the-art omni LLMs, including VideoLLaMA 2 [8] (7B params), VITA-1.5 [10] (7B params), Qwen 3 Omni [44] (35B params), and OmniVinci [47] (9B params). For all the omni LLM baselines, we use their official codebase and use default parameters for inference.

D. Additional Results

D.1. Hyper-parameter Tuning

As described in Sec. 6.2, Fig. 9 shows how different hyperparameter values affect the performance of the proposed approach on different benchmarks. Please refer to Sec. 6.2 (**Strength of hyperparameters**) for a detailed explanation.

D.2. Comparison with decode-time approaches

Tab. 9 shows comparison with decode-time approaches AVCD [21] and VCD* (default settings used from [21]). These decode-time approaches result in moderate gains over the reference model, whereas MoD-DPO++ gains are superior.

D.3. Multi-turn omni-modal results

To show the performance improvement on multi-turn dialog scenarios, we perform evaluation on the audiovisual task subset of OmniDialog benchmark [32] and report results in Tab. 10. We can observe that both MoD-DPO and MoD-DPO++ result in a performance improvement over the baseline, however the improvement in the case of MoD-DPO++ is more pronounced. Moreover, MoD-DPO++ results in a superior performance compared to OmniDPO [4].

Table 10. OmniDialog Results

Method	OmniDialog
Qwen 2.5 Omni	83.91
+ OmniDPO	84.18
+ MoD-DPO	83.96
+ MoD-DPO++	**85.86**

E. Discussion on Cross-Modal Synergy

The MoD-DPO objective defined in Sec. 4.1 mitigates spurious inter-modality correlations by introducing explicit regularization terms that enforce modality invariance and sensitivity. However, many multimodal tasks benefit from constructive synergy between audio and visual modalities for richer audiovisual understanding and reasoning. Extending the MoD-DPO framework with mechanisms to selectively promote beneficial cross-modal interactions, while still suppressing spurious ones, remains an important direction for future work.

F. Prompt Pool

Describe the given audio in 2-3 sentences.

(a) Audio captioning prompt

Describe the given video in detail in a paragraph.

(b) Visual captioning prompt

Figure 10. **Audio and visual captioning prompts** used in Stage-1 of Preference Data Generation Sec. 4.3. Audio captions are obtained from AudioFlamingo 3 [12] and visual captions are obtained from GPT-4o [18].

```
"""The “caption” provides the explanation about audio event. The “visual tagging” identifies the observable visual objects or actions.
If similar objects from the “caption” are found in the “visual tagging”, it is assumed to exist in-view. If not, it is assumed to exist out-of-view.
The “information” aims to summarize the inferable audio and visual information:
- In-view sound source is the object that makes sound and is visible in the scene,
- In-view sound is the type of sound that the object in the scene makes,
- In-view silent object is the object that does not make sound but is visible in the scene,
- Out-of-view sound source is the expected object that makes sound and is not visible in the scene, and
- Out-of-view sound is the type of sound that the object out of the scene makes.

You are required to classify and return the objects into the above categories for the given caption and visual tagging.

Following are a few examples:
======== Example 1 ==========
Caption: In the room, a woman held a can of stuff in her hand and showed it back and forth. The sound of xylophone and children's music came from the background music.
Visual Tagging: beverage | green | squeeze | food | orange juice | market | shopping cart | catch | red | fill | shopping basket | play | cart | bin | basket | can | woman | hand | push |
produce | container | bottle | juice
--------
In View Sound Source: []
In View Sound: []
In View Silent Object: [beverage, food, orange juice, shopping cart, shopping basket, cart, bin, basket, hand, container, bottle, juice]
Out of View Sound Source: ["xylophone", "children"]
Out of View Sound: ["xylophone music", "children's music"]
==============================
======== Example 2 ==========
Caption: With the sound of chewing, a red bike was parked outside, and a man in a dark top stood looking at the bike.
Visual Tagging: motorbike | red | motorcycle | lean | hose | man | pavement | brown | moped | floor | park | lock | attach | walk | night | person | tire | curb | stand | pole | bicycle
--------
In View Sound Source: []
In View Sound: []
In View Silent Object: [motorbike, motorcycle, pavement, moped, floor, park, lock, tire, curb, pole, bicycle]
Out of View Sound Source: ["man"]
Out of View Sound: ["chewing"]
==============================

Now provide the classification for the following:

Caption: """ + REPLACE_AUDIO_CAPTION_STRING + """
Visual Tagging: """ + REPLACE_VIDEO_TAGS_STRING + """

Return the "information" in the following JSON format:
{
  "in_view_sound_source": [list of strings],
  "in_view_sound": [list of strings],
  "in_view_silent_object": [list of strings],
  "out_of_view_sound_source": [list of strings],
  "out_of_view_sound": [list of strings]
}"""
```

Figure 11. **QA Generation Prompt - Stage 2.1.** used for classifying the events/objects in the current audiovisual input into different classes similar to AVHBench [35].

"""You will be given with a audio-related question and an answer for a given audiovisual input. Additionally, you will also have access to the audio caption for the input.
- Your task is to verify whether the answer is correct for the question.
- The question is about whether a specific object/event is audible in the audio or not.

Return your response in the following JSON format:
{
  "verdict": "Yes" or "No" or "Question Invalid",
  "explanation": "brief explanation justifying your verdict based on the audio caption"
}

====================
Audio Caption: """ + **REPLACE_AUDIO_CAPTION_STRING** + """
Question: """ + **REPLACE_QUESTION_STRING** + """
Answer: """ + **REPLACE_ANSWER_STRING**

(a) Video-driven Audio Hallucination

"""You will be given with a vision-related question and an answer for a given audiovisual input. Additionally, you will also have access to the visual caption for the input.
Your task is to generate two sets of detailed answers for the given question.
- First, should start with the given answer (yes/no) and then justify it based on the provided visual caption.
- Second, should be opposite to the given answer (no/yes) and then provide a random reason to justify it.

Return your response in the following JSON format:
{
  "chosen": "detailed answer starting with the given answer",
  "rejected": "detailed answer starting with the opposite of the given answer"
}

====================
Visual Caption: """ + **REPLACE_VIDEO_CAPTION_STRING** + """
Question: """ + **REPLACE_QUESTION_STRING** + """
Answer: """ + **REPLACE_ANSWER_STRING**

(a) Audio-driven Video Hallucination

Figure 12. **QA Generation Verification Prompt - Stage 2.3.** used to verify the automatic QAs generated for video-driven audio hallucination and audio-driven video hallucination.

"""You will be provided with a visual and audio information for a given video. Visual information includes a video caption and a set of tags associated with the video, while audio information includes an audio caption.

Based on this information, you need to generate two sets of captions ONLY ABOUT THE AUDITORY INFORMATION as per the following requirements:
1. Correct caption: Generate an accurate caption that describes the content of the audio based on the provided AUDIO caption.
2. Inconsistent caption (video-relevant): This should follow the same tone and style as the correct caption but should include WRONG information about AUDITORY entities in the audio, based on the VIDEO CAPTION and TAGS.

Frame your responses in a way that they answer to the following question: "Describe only what you hear?". Do not say anything about the visual or video content in the generated captions. Only describe the audio content and respond based on the above two criterias.

Return your response in the following JSON format: {"chosen": "correct caption here", "rele_reject_1": "inconsistent caption (video-relevant)"}
=======
Video Caption: """ + **REPLACE_VIDEO_CAPTION_STRING** + """\n
Video Tags: """ + **REPLACE_VIDEO_TAGS_STRING** + """\n
Audio Caption: """ + **REPLACE_AUDIO_CAPTION_STRING** + """\n
"""

(a) Audio Captioning

You will be provided with a visual and audio information for a given video. Visual information includes a video caption and a set of tags associated with the video, while audio information includes an audio caption.

Based on this information, you need to generate two sets of captions ONLY ABOUT THE VISUAL INFORMATION as per the following requirements:
1. Correct caption: Generate an accurate caption that describes the content of the video based on the provided video caption and tags.
2. Inconsistent caption (audio-relevant): This should follow the same tone and style as the correct caption but should include WRONG information about VISUAL entities in the video, based on the AUDIO CAPTION.

Frame your responses in a way that they answer to the following question: "Describe only what you see?". Do not say anything about the audio content in the generated captions. Only describe the visual content and respond based on the above three criterias.

Return your response in the following JSON format: {"chosen": "correct caption here", "rele_reject_1": "inconsistent caption (audio-relevant)"}
=======
Video Caption: """ + **REPLACE_VIDEO_CAPTION_STRING** + """\n
Video Tags: """ + **REPLACE_VIDEO_TAGS_STRING** + """\n
Audio Caption: """ + **REPLACE_AUDIO_CAPTION_STRING** + """\n
"""

(b) Visual Captioning

Figure 13. **Preference Data Generation Prompt - Stage 3 (Captioning)** used to generate preference data pairs for the captioning task for training MoD-DPO.

"""You will be given with a audio-related question and an answer for a given audiovisual input. Additionally, you will also have access to the audio caption for the input.
Your task is to generate two sets of detailed answers for the given question.
- First, should start with the given answer (yes/no) and then justify it based on the provided audio caption.
- Second, should be opposite to the given answer (no/yes) and then provide a reason related on the given VISUAL CAPTION to justify it.

Return your response in the following JSON format:
{
  "chosen": "detailed answer starting with the given answer",
  "rejected": "detailed answer starting with the opposite of the given answer"
}

====================
Audio Caption: """ + **REPLACE_AUDIO_CAPTION_STRING** + """
Visual Caption: """ + **REPLACE_VIDEO_CAPTION_STRING** + """
Question: """ + **REPLACE_QUESTION_STRING** + """
Answer: """ + **REPLACE_ANSWER_STRING**

(a) Video-driven Audio Hallucination

"""You will be given with a vision-related question and an answer for a given audiovisual input. Additionally, you will also have access to the visual caption and audio caption for the input.
Your task is to generate two sets of detailed answers for the given question.
- First, should start with the given answer (yes/no) and then justify it based on the provided visual caption.
- Second, should be opposite to the given answer (no/yes) and then provide a reason related on the given AUDIO CAPTION to justify it.

Return your response in the following JSON format:
{
  "chosen": "detailed answer starting with the given answer",
  "rejected": "detailed answer starting with the opposite of the given answer"
}

====================
Visual Caption: """ + **REPLACE_VIDEO_CAPTION_STRING** + """
Audio Caption: """ + **REPLACE_AUDIO_CAPTION_STRING** + """
Question: """ + **REPLACE_QUESTION_STRING** + """
Answer: """ + **REPLACE_ANSWER_STRING**

(b) Audio-driven Video Hallucination

Figure 14. **Preference Data Generation Prompt - Stage 3 (Presence)** used to generate preference data pairs for the object/event presence task for training MoD-DPO.