A. APPENDIX

A.1. Psychometric Setup Description in Prompt

We hypothesized that including details of the psychometric experiment in the prompt might guide the model towards better predictions. As an example, we included the following description from LS-PCQA when training and testing on it –

> In the subjective experiment, the participants sit in a controlled environment. Specifically, the zoom rate is set as 1:1. The presentation device used in subjective experiments is Dell SE2216H with a 21.5-inch monitor with a resolution of 1920×1080 pixels. The sitting posture of the participants is adjusted to ensure that their eyes are at the same height as the center of the screen. The viewing distance is about three times the height of the rendered point cloud (≈ 0.75 meters). The subjective experiment is conducted indoors, under a normal lighting condition.

We included a similar description for other datasets as available. Row ⑥ in Table 5 shows the effect of adding this psychometric context. We see a slight improvement in our evaluation metrics, but the performance is comparable with the task only prompt (row ⑤). We believe this is likely as the LLM is already able to draw this information as relevant world knowledge from the task section of the prompt and does not particularly need further explicit details.

A.2. Effect of Rendering Parameters on Perceptual Quality

We observed that quality assessment for point clouds is highly dependent on the settings used to render the point cloud and how the user was allowed to interact with it. For example, Figure 2 shows the same point cloud rendered with different point sizes and viewing distances, all of which have significantly different quality characteristics. This is a complexity typically not observed in 2D quality datasets. Accordingly, we added rendering parameters in our prompt as described in the corresponding datasets when available or a best effort reproduction when not. Method ⑦ in Table 5 shows the effect of including these parameters. As an example, we added the following description for LS-PCQA –

> The point cloud is rendered with a point size of 2 mm with cameras at 2.5m from the object and perspective projection with square primitives.

The improvement is modest over the base case. We believe this is likely because this information can be inferred from a combination of the image projections and the text description of the task, so specifying it explicitly has relatively little impact.

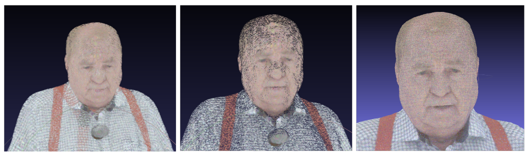

Fig. 2. The same underlying point cloud can have highly different quality characteristics depending on rendering parameters and the radius of interaction, especially in the NR setting. Point cloud taken from LS-PCQA and rendered in MeshLab. Best viewed zoomed in.

A.3. Further Implementation Details

The point cloud projections were rendered with PyTorch3D at a resolution of 512×512. All point cloud samples are $n = 8192$ dimensional with 3 spatial coordinates and 3 RGB color coordinates, which makes $d = 6$. The furthest point sampling was done with the Python package fpsample with the bucket-based FPS algorithm. To sample local patches, we constructed a search tree using the Python package FAISS, sampled a single point randomly and then looked up the closest points near it to construct the final sample. For the two scale patching, uniform downsampling is conducted with Open3D at a factor of 2. The point encoder outputs $m = 513$ point features, each with $c = 384$ dimensions. The point feature projector contains three linear layers with the GeLU activation, which maps point features to tokens with $c' = 5120$ dimensions. Since we added two additional special tokens, the vocabulary size of PIT-QMM is $V = 32003$. The weights of the image encoder and LLM are initialized from Q-Align.

A.4. On Training Efficiency

We report the number of epochs for each model in Table 4 verbatim from the respective technical reports or the code provided. A subtlety in this comparison is that the batch size for all of these models are different, so overall training iterations would vary. However, the batch sizes are within the same range (8-20), so the trends should remain similar even after batch size is normalized. Note that the batch size we used for PIT-QMM is relatively low, so normalizing for a larger batch size as used elsewhere would likely favour our model.