Potential of 2D Priors for Improving Robustness of Ill-Posed 3D Reconstruction

Elia Fantini (336006), Chengkun Li (340485), Ziwei Liu (336553)
Final Report

***Abstract*—This research aims to enhance the robustness of 3D human full-body mesh reconstruction from single images by leveraging pre-trained 2D priors. The ill-posed nature of the problem, coupled with real-life image imperfections, makes this challenging in real-life scenarios. To the best of our knowledge, no work has previously analyzed or improved the robustness of these methods tp real-life corruptions. We propose several techniques to integrate 2D priors into 3D generative models: self-supervised refinement with CLIP Loss, and multimodal learning. Our research places a significant emphasis on robustness evaluation. While our results did not show significant improvements in performance for most corruption cases, they did provide valuable insights for further future exploration.**

I. Introduction

Our research focuses on the problem of comprehensive human full-body mesh reconstruction from a single image, which is assumed to be taken in daily settings. The inherent challenge in addressing this problem stem from its ill-posed nature, as the image doesn't cover all body parts results in a lack of information. Also, real-life images are often subject to imperfections such as defocusing, inadequate lighting, and motion-induced blurring. These corruptions contribute to a pronounced domain shift, further complicating the task of accurate full-body reconstruction.

The efficacy of large pre-trained models has demonstrated the capability to extract semantic [1] and depth information [2] while exhibiting robustness to corruptions. Therefore we propose to integrate different 2D priors into the workflow to enhance the robustness of 3D generative models. As such, our research project aims to address the question: "*Can we enhance 3D generative performance by leveraging pretrained 2D priors, and if so, how to achieve the best results?*"

II. Related Work

In recent years, CLIP [1] has emerged as a powerful representation learning method for tasks that involve both textual and visual data, and has been applied to various 3D-aware synthesis tasks, such as image generation from 3D models. These efforts have led to promising results [3], [4], [5], [6], [7]. One notable example is ELICIT [8], which extends CLIP-driven NeRF for human-specific rendering, with a complex pipeline that involves several advanced techniques, where semantic consistency is just one of the many forms of supervision. In contrast, our method proposes an integration of the CLIP model-based prior into the simpler PIFu pipeline to enhance its impact on the robustness of corrupted data, with different ways of introducing such prior as well as additional elements.

Some recent works explored the capabilities of diffusion models on the task of 3D reconstruction [9], [10], [11] but none of them is specific to human meshes.

Our proposed approach is influenced by DietNeRF [12], which utilizes CLIP visual transformer for encoding rendered images from different viewpoints to improve reconstruction quality. However, our method differs in that we solely rely on semantic supervision during test time, unlike existing methods. Furthermore, while DietNeRF focuses on limited input images and doesn't specifically address single-image reconstruction, our approach explores the impact of 2D priors on robustness, which has received less attention.

Liu et al. introduced Prismer [13], a vision-language model that excels in data efficiency. Prismer incorporates diverse pre-trained experts and predicted multi-modal signals, achieving enhanced performance across domains. It addresses the challenge of varying input length from different experts, handling heterogeneous data inputs.

Ranftl et al.'s "Vision Transformer for Dense Prediction" [14] explores the application of vision transformers for dense prediction tasks. Their approach leverages large pretrained 2D priors, leading to improved accuracy and efficiency compared to previous methods. This work has implications for computer vision tasks such as 3D reconstruction, scene understanding, and augmented reality, emphasizing the benefits of pretrained models and multi-modal signals in improving performance with limited labeled data.

III. Preliminary Preparation

A. Dataset Generation

PIFu [15] uses the private RenderPeople dataset, hence to overcome its proprietary nature, we switched to the THuman2.0 dataset, consisting of 526 high-quality human scans captured with a DSLR camera rig. Each scan includes a 3D model and texture map.

We adapted the THuman2.0 data for the original PIFu implementation, resulting in a more desirable reconstruction baseline.

To match original PIFu's data format, we rescaled and translated the meshes to be similar to RenderPeople. For training, we created a dataset of image-groundtruth 3D mesh pairs by rendering meshes into 360-degree images using

precomputed radiance transfer. Each mesh generated 360 images, and we stored images, camera parameters, and texture information for training, including UV mappings. Rendering code was borrowed from JIFF [16], a methodology based on THuman2.0 that creates a training dataset similarly to PIFu.

B. Masks for Original Dataset

In our pipeline, we incorporate image segmentation to isolate the human body and remove the background. To accomplish this, we utilize the Rembg library, which offers several pre-trained models. The goal of employing image segmentation is to mitigate the impact of low-quality masks on PIFu's reconstruction process. By isolating the human body accurately, we can focus primarily on analyzing the robustness of PIFu's neural networks.

To evaluate the effectiveness of different segmentation models, we conducted a comprehensive comparison of reconstruction metrics. In addition, we subjected the original masks from the non corrupted dataset to the same corruption process and utilized the resulting masks on the corrupted images. We included this method in the comparison. In our analysis, we found no significant difference among the different methods. As a result, we selected the U^2-Net_p model, a lightweight version of the U^2-Net, to expedite the masking process without compromising the overall quality.

For further details and a comprehensive comparison, please refer to the Appendix A.

C. Robustness Benchmark

Our data corruption approach follows the methodology presented in the referenced paper [17]. An example of all corruptions applied are shown in Appendix A. These corruptions were chosen based on their likelihood of occurring in real-world captures. Out-of-focus, blurry, rotated, dark, and foggy images can commonly occur in outdoor smartphone captures. Digital corruptions, such as compression artifacts, can arise during online image transmission for reconstruction on server.

IV. Method and Deliveries

A. Baseline Method

We take PIFu [18] as our baseline method. We then feed corrupted images into PIFu at the testing time to observe the degradation in reconstruction performance since PIFu is not trained on corrupted data, where the robustness issue originates.

As an additional baseline method we test how a common method to improve robustness work in our scenario, performing test-time adaption techniques to adapt to the domain shift [19].

B. Proposed Method

We propose several ways to incorporate 2D priors into the baseline 3D generative pipeline, and they can be abstracted into *self-supervised finetuning*, and *multimodal learning*.

1) Self-supervised refinement: We propose to impose 2D priors in the form of enforcing semantic consistency between renderings of the estimated reconstruction from unseen views ($\hat{I}_{train}$) and the input image ($\hat{I}_{ref}$).

To do so, we plan to introduce pre-trained visual encoders like CLIP [1] to the original pipeline through a CLIP ViT-based cosine distance (as a loss) defined as follows:

$$\mathcal{L}_{CLIP} = \Phi(I_{ref})^T \Phi(\hat{I}_{train}), \tag{1}$$

where Φ is the normalized embedding function of the CLIP ViT.

The CLIP loss leverages the intuition that the CLIP model has been trained on a large and diverse dataset, enabling it to capture both semantic and geometric information. This makes it a valuable source of supervision signal as CLIP's embeddings are expected to exhibit robustness to various corruptions.

To evaluate this hypothesis, we conducted experiments to measure cosine similarities between embeddings of images of the same person from different viewpoints, as well as between the same image with and without corruption. The results, summarized in Table I, show that CLIP indeed demonstrates robustness to such corruptions. Even the lowest similarity score observed in the experiments remains significantly close to the similarity between images of the same person but from different viewpoints, such as front and back.

Table I
AVERAGE CLIP SEMANTIC SIMILARITY FOR EACH CORRUPTION TYPE AND BETWEEN IMAGES FROM DIFFERENT POINTS OF VIEW (POV), DARKER MEANS LOWER SIMILARITY.

Corruption Type	Average Similarity
XY motion blur	0.8801
Color quantization	0.8733
Camera roll	0.8751
Fog 3D	0.9440
Z motion blur	0.8756
H265_crf	0.8793
Low light	0.8441
Near focus	0.8223
Bit noise	0.8943
Iso noise	0.8580
Zoom blur	0.8643
H265_abr	0.7952
Different POV: Front and Left	0.9214
Different POV: Front and Right	0.9097
Different POV: Front and Back	0.8047

During the training phase, our pipeline operates equivalently to the baseline approach. However, during the testing phase, when the input image is corrupted, the PiFU network may generate 3D mesh reconstructions of lower quality.

To facilitate fine-tuning, we employ a differentiable rendering algorithm to generate four images of the estimated human model from four distinct viewpoints. By minimizing the CLIP loss and backpropagating the loss signal through the rendering process to the texture network, we modify the

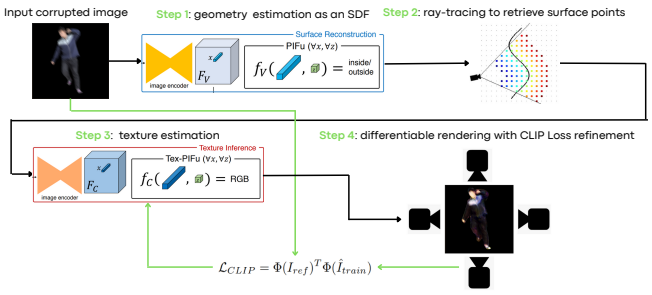

Figure 1. Pipeline for the method of Self-Supervised refinement

generated colors to ensure semantic consistency, as depicted in Figure 1. 1.

Our proposed method implicitly necessitates understanding the optimal strategy for sampling cameras around the estimated mesh and rendering images to compute the CLIP loss. Regarding the camera sampling, we conducted experiments with varying numbers of cameras positioned around the object. However, our results demonstrated that augmenting the number of cameras did not significantly impact the final outcome. Consequently, we opted to generate a minimal set of four cameras, evenly distributed around the object (with an offset of 90 degrees). This approach allows us to supervise each part of the model effectively while minimizing the computational cost associated with the rendering process.

Concerning the rendering algorithm, calculating the gradient through the entire ray tracing process incurs substantial computational time and memory requirements. As such, it becomes impractical without advanced optimizations, which are beyond the scope of this project. Therefore, we employ a two-step approach: first, we obtain the surface points through ray tracing without calculating gradients, and then we enable gradient calculation and compute the colors using the texture network on the retrieved surface points. This design choice limits the refinement process to the texture network exclusively.

2) Multimodal learning: We incorporated pretrained 2D priors into the PIFu pipeline using a multimodal learning framework. The pretrained priors were obtained from various domains, such as CLIP for visual language, DPT for normal estimation, and pure frontal images with patch embeddings.

In our problem setting, the key concept is feature fusion, which involves merging feature vectors from CLIP with feature maps from the PIFu encoder and DPT prediction map. This fusion process enables effective integration of diverse information sources.

In our initial trial, we focused on incorporating CLIP as a semantic encoder to assess its feasibility. We selected CLIP `ViT-L/14` in our experiment setup and established two naive approaches for incorporating CLIP: `tf_concat` and `add`.

In the `tf_concat` approach, we transformed and upsampled the CLIP feature to match the dimensions of the original PIFu encoder feature map. We then concatenated it with the PIFu feature map.

In the `add` approach, we performed a shape transformation and calculated the mean of CLIP feature in the pixel features dimension. And subsequently added this transformed feature map element-wise to the original PIFu encoder feature map.

Taking inspiration from Prismer [13], we developed an experts fusion module utilizing Prismer's Expert Resampler. This module is capable of accommodating multimodal features/signals of varying lengths and quantities, represented as feature maps and feature vectors (as depicted in Fig. 2). To establish consistency in the comparison, we set the hyperparameters of the experts fusion module as follows: $L = 2$ layers, 8 heads, and a latent dimension of 128 (matching the width/height of PIFu's encoder feature map).

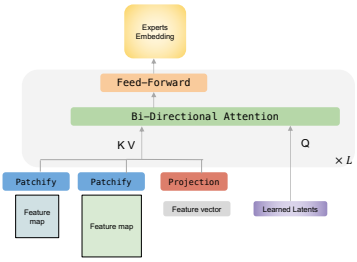

Figure 2. Experts Fusion Module, inspired by the Expert Resampler module in Prismer [13].

For Experts Fusion module, we present further deatils on its number of trainable parameters as well as its total number of parameters in Appendix A.

V. Results

A. Evaluation Metrics

The metrics we use in this experiment as well as the following ones are *Average Point-to-Surface Euclidean distance (P2S)* and *Chamfer Distance* between the reconstructed and the ground truth surfaces.

B. Experiments on baseline

On Milestone 1 we conducted an initial evaluation of the robustness of our baseline pipeline PIFu [15] with the weights provided by the author, trained on RenderPeople dataset, using our benchmark. For each type of corruption, we randomly selected five subjects—individuals from the Thuman2.0 dataset in order to assess the performance variations. We then repeated the same experiment on our PiFU baseline model trained from scratch on Thuman2.0 dataset, with similar results.

The experiments on pretrained PIFu are summarized in . Except for Fog 3D and H265 crf, all corruptions resulted in degradation. Negative values are probably due to slightly better segmentation. Many corruptions led to poor reconstructions due to bad segmentation, with most parts of the body masked out. Corruptions that didn't affect segmentation showed artifacts or missing details, with camera roll showing the worst degradation with perfect segmentation. Examples are shown in figure 3.

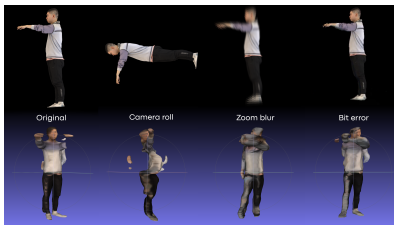

Figure 3. Inference examples: in the higher row there are the input images, in the bottom row the reconstructed meshes (seen from the front).

C. Few-Sample Fine-Tuning: Experiment Setup

This section aims to establish a baseline for enhancing model robustness. Given the constraints of time and resources, our focus will be on test-time adaptation, particularly few-sample fine-tuning.

We conducted experiments with few-sample fine-tuning, a variant of transfer learning, to enhance our baseline model's performance on a corrupted benchmark dataset. The model was fine-tuned with 2 and 3 batches (batch size 16) of corruption data before testing its performance on the robustness benchmark. In the experiment, we use Adam optimizer with a learning rate set at 7e-4 with hyperparameter search based on validation loss on corrupted benchmark (a separate validation set).

The results of these two experiments are presented in Fig. 6 while additional data is reported in Appendix A.

D. Experiments on Self-supervised Refinement

In our experimental evaluation, we conducted tests on five distinct human meshes that were not included in the training dataset. For each input image, we applied all the aforementioned corruptions. Initially, we obtained an initial estimation of the model and proceeded to refine it through 100 iterations. At each iteration, we calculated the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) losses.

Our experiment involved calculating the embedding from a single view, chosen sequentially from the available options. We then compared this embedding to the embedding of the input image using a loss function. Alternatively, we also explored calculating the average of the embeddings from all different views and directly applying the loss on the averaged embedding. The results are illustrated in Figure 4.

The results highlight an important observation: although the supervision signal effectively guides the network to reduce the CLIP loss during iterations, it does not lead to improved texture reconstruction. In fact, quality metrics such as PSNR and SSIM show a decrease. Furthermore, the final plot reveals that the embedding to which the refined model converges differs from the embeddings obtained from the four ground truth images, as the distance between them increases.

For further details and a comprehensive comparison, please refer to the Appendix A.

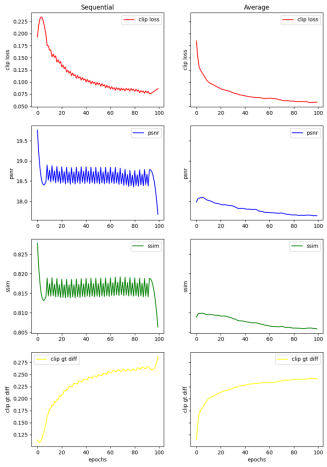

Figure 4. Plots of Clip Loss, PSNR, SSIM, and difference from Clip groundtruth metrics through iterations.

Upon analyzing the rendered images, we observed that the supervision signal does not entirely compromise the initial reconstruction. It successfully addresses certain inconsistencies, such as removing stains of different colors on the clothing, as depicted in Figure 5. However, it also introduces additional artifacts, such as a shifted skin color on the face. Overall, the introduction of these artifacts diminishes the overall quality. Interestingly, the method without calculating the averaged embedding seemed to produce slightly fewer artifacts.

These findings underscore the limitations of relying solely on the provided supervision signal for guiding the reconstruction process. While it addresses some inconsistencies, it also introduces new artifacts that adversely affect the overall quality.

E. Experiments on Multimodal Learning

In this section, we performed experiment with multimodal learning with 2D Priors. To begin with, we experiment with our naive approaches (mentioned in Section IV-B2); and to save computational resource, we first did a comparison on both naive approaches, and found limited difference in terms of training error decay and validation error, therefore we pick

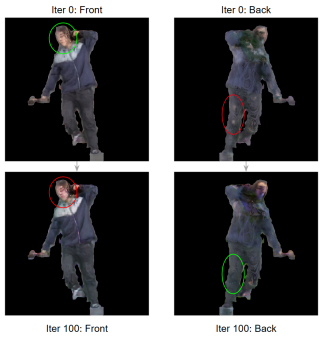

Figure 5. Examples of rendered images before and after refinement (back and front view). Circles show improvements and artifacts.

`add` as our naive approach and report the results based on this method of feature fusion.

We pick the same hyperparameters for this model built with feature fusion module during training. And we report the results of this model trained for 5 epochs and test on corrupted/uncorrupted data in Fig. 6.

We present the results of the model incorporating the Experts Fusion module[1] in Fig. 6 in Fig. 6. The hyperparameters for this model were kept the same as the vanilla baseline and the naive CLIP feature fusion.

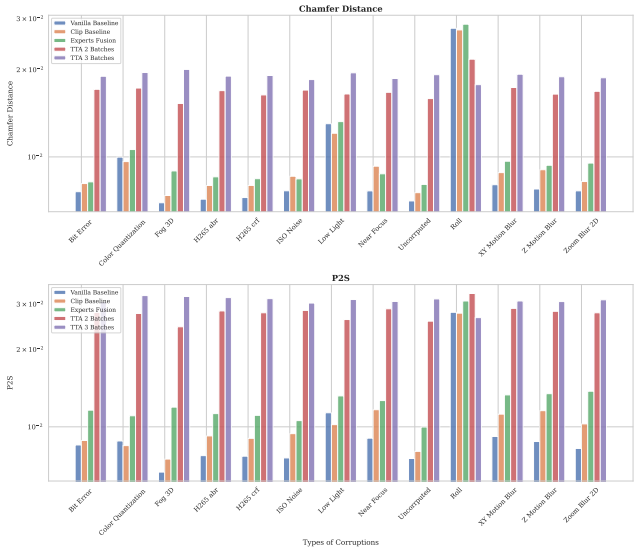

Figure 6. Chamfer Distance and P2S metrics of all experiments conducted in the multimodal learning and vanilla baseline experiments.

In Fig. 6, we observed that all the methods proposed in the multimodal learning section and the test time adaptation segment performed worse than the vanilla baseline method on uncorrupted data. However, we did notice that compared to the vanilla baseline, the naive approach of incorporating CLIP features performed better when tested on *Color Quantization*, *Low Light*, and *Roll* corruptions. Additionally, we observed that the Experts Fusion model, with 1/10 of the trainable parameters and a shorter training time, exhibited similar performance to the naive CLIP feature fusion model. Furthermore, the Experts Fusion model outperformed the naive CLIP feature fusion model slightly in terms of Chamfer Distance on the ISO corruption.

Regarding test time adaptation, we found that training on adaptation data resulted in a degradation of performance on clean data. However, we observed that the degree of degradation decreased with an increase in adaptation batches, as illustrated in Fig. 7.

VI. Conclusion and Limitations

Reflecting on our comprehensive exploration of leveraging 2D priors to enhance the 3D generative ability, we acknowledge that it remains a challenging and worthwhile problem. This is primarily due to the risk of the model learning irrelevant features and the inherent gap between 2D and 3D information. Further research and exploration are necessary to bridge this gap and improve the overall performance of the 3D generative models.

A. Limitations

- Due to time and resource constraint (we didn't forsee the limitation of 12 hour usage of cluster per training) and it took 4-5 hours to train the baseline per epoch, so all our reported results are only before epoch 5, which generates considerable limitation in the comparison.
- More ablation studies could be done.

B. Conclusion and Future Work

Our experiments show a potential in the incorporation of 2D priors which is worth further exploration. Self supervised finetuning sligthly decreases quality introducing artifacts, but it also improves little details, so exploring a better incorporation of such prior to guide refinement is needed. Although we incorporated an experts fusion module into our multimodal learning framework, we were unable to perform Test Time Augmentation (TTA) using this framework due to time/resource limitation. This missed opportunity is regrettable considering that our framework has a reduced number of trainable parameters (see Appendix A), implying potential ease of training and adaptability to corrupted data. Furthermore, considering the argument put forth in the paper [20], expecting the model to generalize well with stronger 2D priors may not always be practical, as there is a lack of training data (i.e., corrupted images) to guide the model in eliminating non-robust features. Consequently, we suggest exploring if the results could be improved by conducting test time adaptation with a fully trained expert fusion module.

[1]Regrettably, the training of this model could not be completed within the project's deadline. Therefore, we present the results of this model at epoch 4, in comparison to the results of other models at epoch 5.

REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[2] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[3] C. Wang, M. Chai, M. He, D. Chen, and J. Liao, "Clip-nerf: Text-and-image driven manipulation of neural radiance fields," 2022.

[4] A. Mirzaei, Y. Kant, J. Kelly, and I. Gilitschenski, "Laterf: Label and text driven object radiance fields," 2022.

[5] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu, "Avatarclip: Zero-shot text-driven generation and animation of 3d avatars," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–19, 2022.

[6] D. Xu, Y. Jiang, P. Wang, Z. Fan, Y. Wang, and Z. Wang, "Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360° views," 2022.

[7] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," 2022.

[8] Y. Huang, H. Yi, W. Liu, H. Wang, B. Wu, W. Wang, B. Lin, D. Zhang, and D. Cai, "One-shot implicit animatable avatars with model-based priors," *arXiv*, 2022.

[9] J. Tang, T. Wang, B. Zhang, T. Zhang, R. Yi, L. Ma, and D. Chen, "Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior," 2023.

[10] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv*, 2022.

[11] E. R. Chan, K. Nagano, M. A. Chan, A. W. Bergman, J. J. Park, A. Levy, M. Aittala, S. D. Mello, T. Karras, and G. Wetzstein, "GeNVS: Generative novel view synthesis with 3D-aware diffusion models," in *arXiv*, 2023.

[12] A. Jain, M. Tancik, and P. Abbeel, "Putting nerf on a diet: Semantically consistent few-shot view synthesis," 2021.

[13] S. Liu, L. Fan, E. Johns, Z. Yu, C. Xiao, and A. Anandkumar, "Prismer: A vision-language model with an ensemble of experts," *arXiv preprint arXiv:2303.02506*, 2023.

[14] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 179–12 188.

[15] S. Saito, , Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," *arXiv preprint arXiv:1905.05172*, 2019.

[16] Y. Cao, G. Chen, K. Han, W. Yang, and K.-Y. K. Wong, "Jiff: Jointly-aligned implicit face function for high quality single view clothed human reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 2729–2739.

[17] O. F. Kar, T. Yeo, A. Atanov, and A. Zamir, "3d common corruptions and data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 963–18 974.

[18] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[19] F. Fleuret *et al.*, "Test time adaptation through perturbation robustness," in *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[20] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.

APPENDIX

A. Author contribution statements

This is an autor contribution statements for Elia Fantini (E.F.), Chengkun Li (C.L.), and Ziwei Liu (Z.L).

- E.F, C.L, and Z.L. equally contributed in conceiving the idea for the first proposal and wrote the proposal in equal way.
- E.F, C.L equally worked on changing the original proposal idea to fit to the request presented in the revision by TA team.
- In the revised proposal, Z.L wrote Introduction and section II.A (baseline method). E.F wrote Abstract, and III. C.L wrote IV. E.F and C.L equally contributed to II.B (proposed methods).
- C.L. debugged and solved problems with installation due to cluster dependencies incompatibilities. He adapted code from ICON GitHub repository to generate training dataset and test inference with pretrained baseline.
- E.F searched for an alternative solution to generate training data and found JIFF GitHub repository.
- Z.L installed JIFF environment and run code to generate training dataset.
- E.F. installed PiFU GitHub repository environment and wrote guidelines to successfully install it on cluster, solving bugs and incompatibilities.
- E.F installed the environment for 3D Common Corruptions and Data Augmentation GitHub Repository and adapted code to suit our needs. Experimented the possibility to apply corruptions using Blender but got stuck in the adaptation of code to run headless on the cluster.
- E.F wrote evaluation code to test pre-trained baseline on corrupted data. Integrated Chamfer, P2S metrics calculation in the evaluation pipeline. Tried to adapt Normal Reprojection Error code to render images headless on cluster but failed. E.F experimented masking creation solutions on corrupted images with segmentation model and integrated it into the evaluation process. C.L. integrated masking in training and dataset creation code.
- E.F performed experiments on performance degradation of baseline on corrupted data and wrote script to generate qualitative and quantitative tables/images.
- C.L. wrote code to adapt dataset loading class to correctly load corrupted images and relative altered camera parameters.
- C.L wrote code to perform Test Time Adaptation (TTA) with Few Sample Finetuning and adapted original training code.
- C.L. performed different experiments with different hyperparameters on TTA and wrote code to generate plots for the report.
- For Milestone 1 Report, Z.L. wrote I, II.A, and III. E.F wrote Abstract, II.B, IV, Appendix. C.L. wrote V, VI, Appendix.
- L.Z. adapted E.F.'s code to generate corrupted data to apply corruptions on masks and integrated it to generate augmented training dataset.
- E.F. wrote code to run a comparison of different masking creation techniques to evaluate the impact on final reconstruction performance.
- C.L. adapted training code to train on THuman2.0 and trained the vanilla baseline model, and E.F. evaluated the model comparing it to the pretrained one.
- C.L. wrote code to evaluate CLIP robustness on corruptions.
- E.F. implemented differentiable ray tracing rendering code and integrated it into the PiFU pipeline.
- E.F. wrote code to introduce CLIP Loss, perform finetuning of the texture network at test time, compute metrics, and save them with rendering results. Experimented with different rendering hyperparameters, different sampling camera strategies, and different ways of incorporating CLIP embeddings in refinement. Run experiments on all corruption input images and wrote code to produce quantitative/qualitative plots/images.
- C.L. implemented the code for the multimodal learning framework, which included the tf_concat, add, and experts fusion modules. C.L. conducted hyperparameter search and trained models for both add-based feature fusion and experts fusion-based feature fusion.
- E.F. evaluated the results on both uncorrupted data and the corrupted data benchmark. C.L. and E.F. made equal contributions to the analysis of the multimodal learning results.
- C.L. and E.F. created presentation slides and recorded explanatory video.
- C.L. and E.F. equally contributed to writing of the Final Report.
- Z.L. cleaned code from commented lines, wrote a readme file and instructions on installing the environment and running corruption data. C.L. and E.F. merged branches and solved conflicts, integrated readme with instructions to run experiments.
- C.L. trained different models and TTA finetuned models, and E.F. run evaluation on such models. C.L wrote code to generate a visualization of the comparison of different methods.

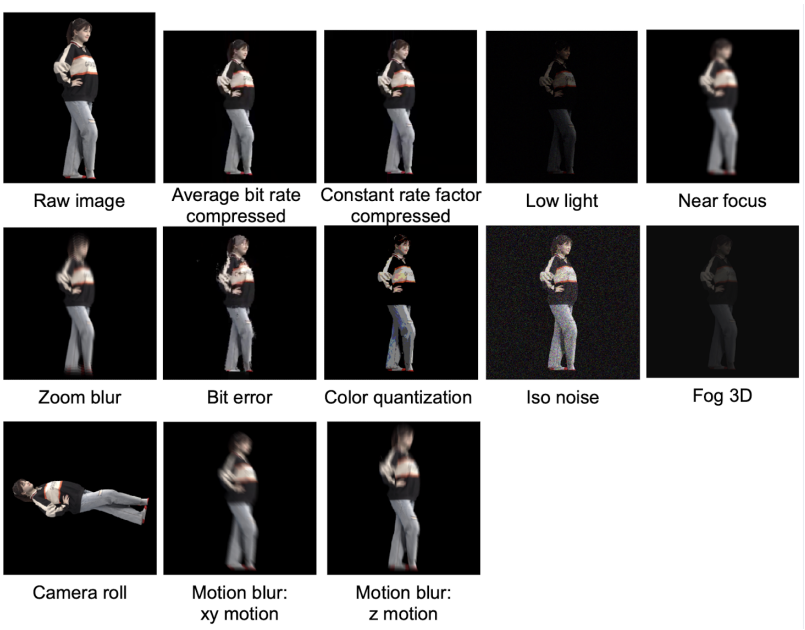

Figure 7. The original and corrupted pictures for a fixed object and yaw angle. The corruptions are designed to resemble real life scenarios.

Table II
COMPREHENSIVE COMPARISON OF CHAMFER AND P2S DISTANCES USING DIFFERENT MASKING GENERATION TECHNIQUES ON THE MOST CRITICAL CORRUPTIONS ("MASK" IS CORRUPTING THE ORIGINAL MASK, OTHERS ARE SEGMENTATION MODELS) .

	mask		isnet		silueta		u2net		u2nethuman		u2netp	
Corruption type	Chamfer	P2S	Chamfer	P2S	Chamfer	P2S	Chamfer	P2S	Chamfer	P2S	Chamfer	P2S
no corruption	0.0405	0.0387	0.0409	0.0385	0.0405	0.0387	0.0409	0.0379	0.0402	0.0376	0.0404	0.0377
low light	0.2484	0.1703	0.069	0.0689	0.2552	0.1868	0.1855	0.1585	0.2434	0.1786	0.2369	0.1963
iso noise	0.1575	0.1375	0.0646	0.0602	0.1463	0.1268	0.0996	0.0738	0.1449	0.1309	0.1131	0.1019
zoom blur	0.0695	0.0753	0.0721	0.0789	0.0697	0.0754	0.0705	0.0761	0.0701	0.0761	0.07	0.0756
camera roll	0.2417	0.2441	0.2352	0.2289	0.2407	0.2436	0.2425	0.2399	0.2417	0.2428	0.2436	0.2426

Table III
COMPARISON OF TRAINABLE PARAMETERS AND TOTAL PARAMETERS

Model	**Trainable Parameters**	**Total Parameters**
Baseline	15,604,738	15,604,738
Experts fusion	**1,185,795**	**567,078,959**
`add`-based fusion	15,604,738	443,221,251
`tf_concat`-based fusion w/ frozen PIFu encoder	**1,185,795**	15,604,738

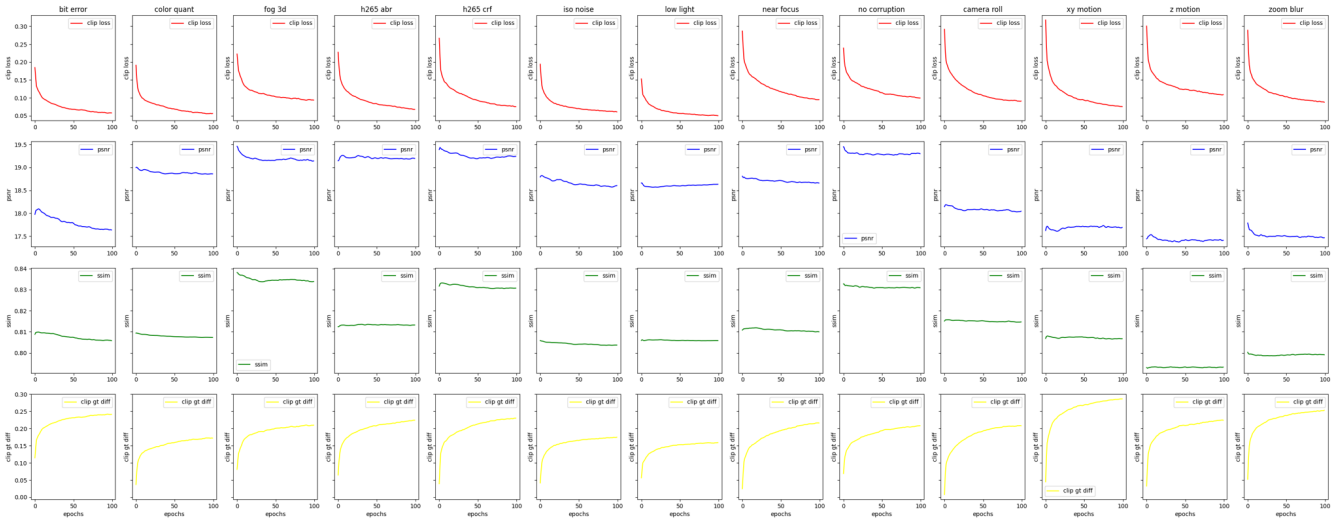

Figure 8. Comprehensive comparison of metrics on all corruptions during Clip Self-Supervised Refinement through iterations: Averaged embedding calculation

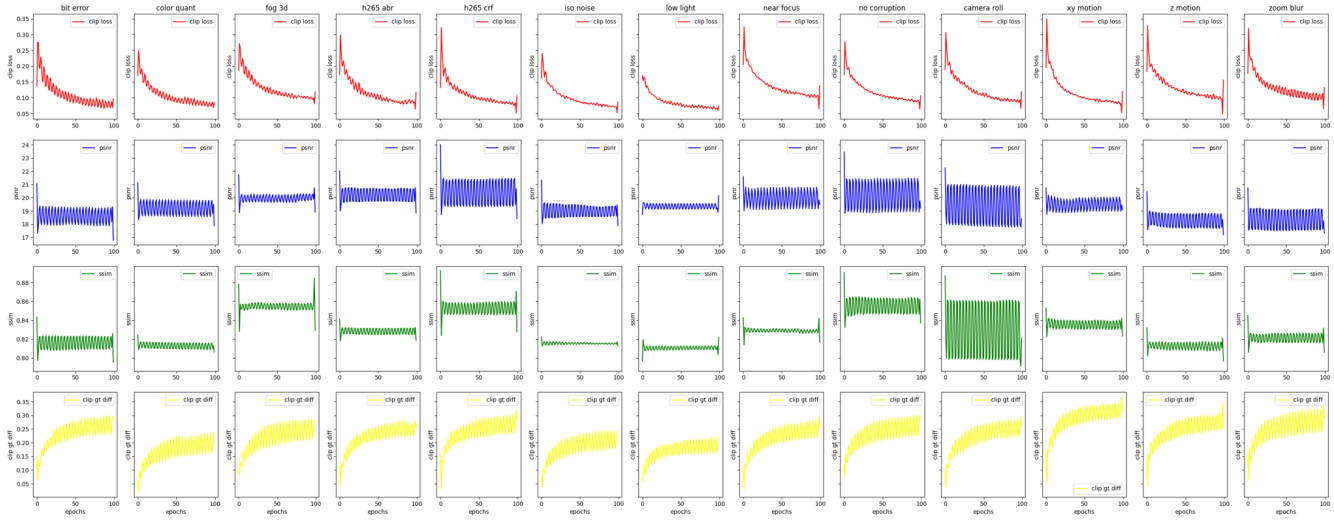

Figure 9. Comprehensive comparison of metrics on all corruptions during Clip Self-Supervised Refinement through iterations: Sequential embedding calculation

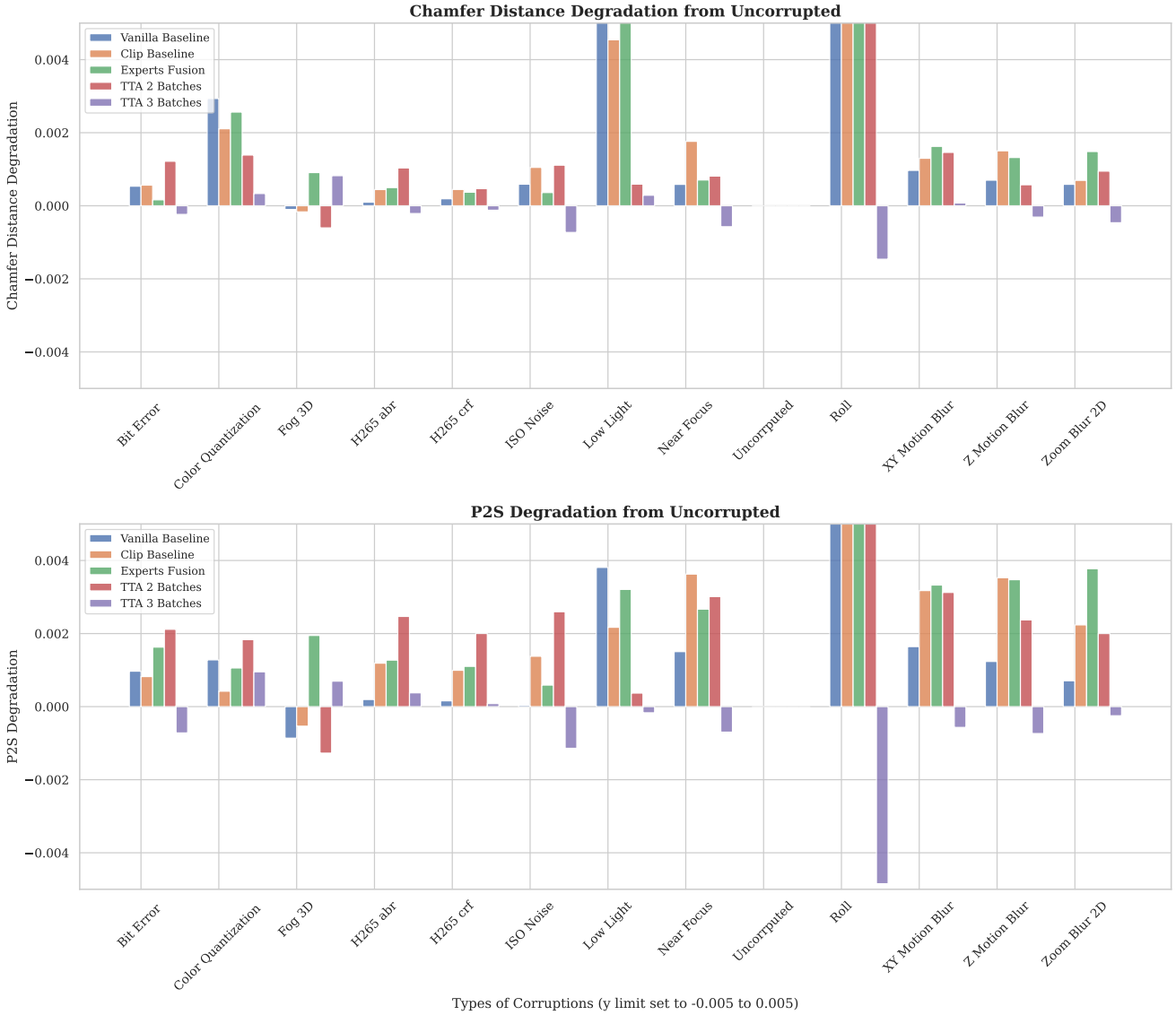

Figure 10. Chamfer Distance and P2S metrics of all experiments conducted in the multimodal learning and vanilla baseline experiments.