# Text2Head – CLIP-guided Latent Code Optimization

Katharina Schmid
katharina.s.schmid@tum.de

Simon Langrieger
simon.langrieger@tum.de

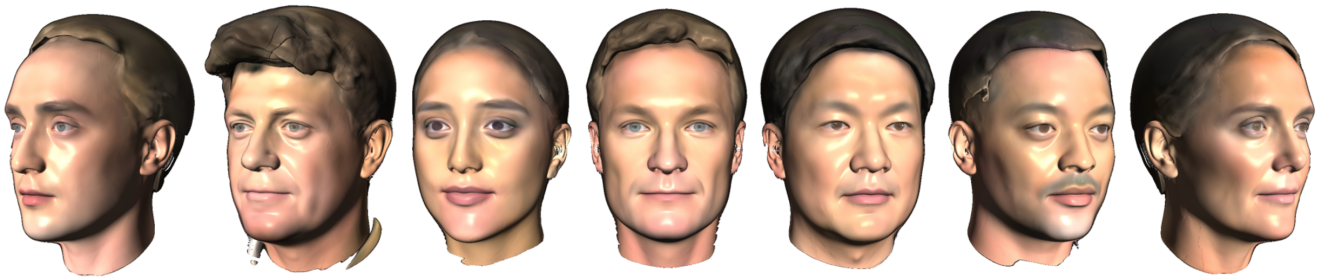


Figure 1. Results for the prompts: "Draco Malfoy", "John F. Kennedy", "Ariana Grande", "Barney Stinson", "Ken Jeong", "Will Smith", "Heidi Klum"

## Abstract

*We propose Text2Head, a novel method for generating neural parametric 3D head models driven by text descriptions. Our approach takes textual prompts describing a person and outputs latent codes for geometry and appearance, which are then used to generate textured 3D head geometries with a pre-trained Monocular Parametric Head Model (MonoNPHM). In contrast to existing approaches, we do not require the prior generation of ground truth pairs of text prompts and latent codes, which can be limited in quality and availability. Instead, our method allows direct optimization of latent codes leveraging a CLIP loss. Our method demonstrates the capability to faithfully generate 3D head models for various applications.*

## 1. Introduction

Easy generation of tailored 3D head models is crucial for various applications such as computer games, movie production, and AR/VR settings. Our objective is to create realistic 3D human head models from textual descriptions. While current methods excel in generating prompt-specific 2D images, 3D object generation remains challenging. We leverage recent advancements in human parametric head models provided by MonoNPHM. MonoNPHM represents a textured 3D head geometry using three latent codes. Contrary to contemporary trends, utilizing this model allows us to formulate the generative task as an optimization problem.

Our major contributions can be summarized as follows:

- Proposing a novel method for faithfully generating 3D human head models from text prompts leveraging CLIP-guided latent code optimization
- Optimizing rendering parameters for maximum CLIP signal strength

## 2. Related Work

**Text to 3D** While significant progress has been made in text-to-image generation [11–13], Text-to-3D generation remains a challenging problem due to the unavailability of paired text and 3D data at large scale. A variety of works alleviates this issue by leveraging CLIP [10], a model that learns a joint embedding space for image and text.

Text2Mesh [9] modifies an input mesh to conform to a target text by predicting color and geometric details. The neural style network is optimized using a CLIP-based semantic loss between the text prompt and rendered images of the 3D mesh.

CLIP-Mesh [6] optimizes the shape and texture of a mesh such that the CLIP score of rendered images and text prompt is maximized.

Dream Fields [5] optimize a Neural Radiance Field per 3D object so that the rendered images score highly with a target caption according to a pre-trained CLIP model.

**3D Human Heads** MonoNPHM [4] make use of a neural parametric representation that disentangles identity, expression and texture in disjoint latent spaces. To this end, three trained models (1) represent the person's head in a neutral expression as a signed distance field (SDF) (2) model facial expressions using a deformation field and (3) texturize the model. This allows the reconstruction of 3D RGB head models.

CLIP-Head [8] leverages two networks mapping the CLIP encodings of an input text prompt to NPHM's [3] disjoint identity and expression codes. To train the mapping networks, they generate ground truth pairs of latent vectors and CLIP embeddings utilizing ControlNet [15] and the dataset proposed in [3]. Additional steps allow the generation of texture details.

# 3. Method

## 3.1. Latent Code Optimization

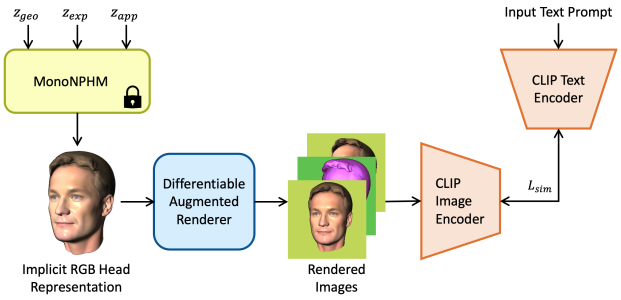



Figure 2. We optimize the latent code $\mathbf{z}$ such that the CLIP similarity between the images rendered from the implicit 3D RGB head representation and an input text prompt is maximized

An illustration of our method is provided in Figure 2. Our work aims to create accurate 3D head models that align closely with a given textual prompt.

We accomplish this by leveraging MonoNPHM [4], employing three latent codes to embed human head features. Specifically, the latent geometry code $\mathbf{z}_{geo}$ stores identity-specific geometric features, the latent expression code $\mathbf{z}_{exp}$ captures the deformations from a canonical pose to a specific facial expression, and the latent appearance code $\mathbf{z}_{app}$ characterizes the texture. Our methodology optimizes the 3D head model defined by the latent representation

$$z = [z_{\text{geo}}, z_{\text{exp}}, z_{\text{app}}]$$

to align with the input text prompt. We initialize $\mathbf{z}$ with zeros and then iteratively render the implicit 3D head representation given by the pre-trained MonoNPHM and update the latent representation $\mathbf{z}$ such that the CLIP similarity between the rendered images and the input text prompt is maximized. Formally, the optimization objective is expressed as:

$$z^* = \underset{z}{\text{argmax}} \sum_{views} \text{CLIP}(I(z, \theta), \text{prompt})$$

where the images $\mathbf{I}$ are rendered based on the rendering parameters $\theta$, the latent representation $\mathbf{z}$ and the pre-trained MonoNPHM.

**Differentiable Augmented Renderer** Given the latent codes and an input point $\mathbf{x}_p$ in pose space, MonoNPHM outputs a signed distance and an RGB value. To render an image from this implicit 3D head representation, we use accelerated sphere tracing [1] and phong shading. The optimal rendering parameters are determined based on 3.2. We render from multiple views, randomly vary the rendering parameters within specified regions and sample the latent representation $\hat{\mathbf{z}}$ around its original value $\mathbf{z}$ following:

$$\hat{z}_i = z_i + \alpha \cdot \mathcal{N}(0, I) \cdot \delta_i \text{ for } i = \{\text{geo}, \text{exp}, \text{app}\}$$

to stabilize the optimization process (details in section 4.2).

## 3.2. Rendering Parameter Optimization

To ensure the effectiveness of our latent code optimization, we employ a strategy to maximize the signal derived from the CLIP model by optimizing our rendering parameters. Specifically, we manually annotated 182 individuals sourced from the NPHM dataset [3]. The categorical annotation encompasses the following attributes: gender, age, ethnicity, hairstyle, beard, facial expression, emotion. For each attribute, we defined a set of prompts $\mathbf{p}$, e.g. "A young person" and "An old person" for the attribute *age*. This gives us correct and incorrect 3D head - prompt pairs. Consequently, we can optimize the rendering parameters $\theta$ by solving:

$$\theta = \underset{\theta}{\text{argmax}} \sum_{p} f\left(\text{Signal Score}_p(\theta)\right)$$

Where $\mathbf{f}$ is a penalty function, which penalizes negative input values. The prompt-specific $\mathbf{SignalScore}_p$ is given by:

$$\begin{aligned}\text{Signal Score}_p(\theta) = \text{CLIP Score}(\text{correct pairs}, \theta).\text{mean} \\ - \text{CLIP Score}(\text{incorrect pairs}, \theta).\text{mean}\end{aligned}$$

# 4. Results

## 4.1. Quantitative & Qualitative Results

To quantitatively evaluate our method, we employ 30 renowned personalities as our reference dataset, given the ease of accessibility to ground truth data. We report three metrics to assess the performance of our method:

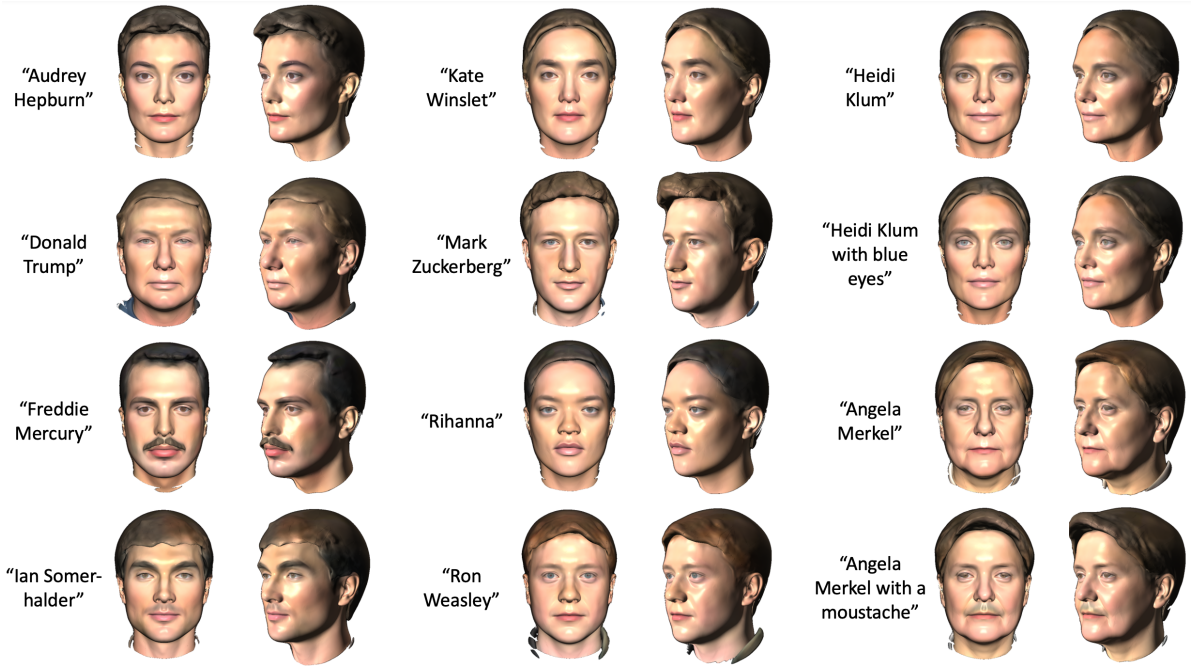



Figure 3. Qualitative results for celebrities (left and middle) and modified celebrities (right)

**CLIP Score** We defined four rendering configurations as depicted in Figure 4. The CLIP similarity [10] between each rendered image and the input prompt is computed, and the mean score is reported.

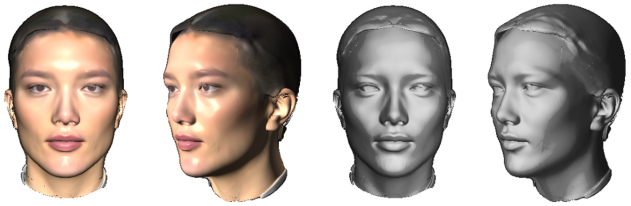


Figure 4. The rendering configurations encompass a frontal view and a 45° side view for both textured and untextured scenarios.

**DINO Score** For each celebrity, we curate a minimum of three images. We compute the cosine similarity between each DINO embedding [2] of the four rendered images and the ground truth images and report the mean value.

**FaceNet Score** Leveraging FaceNet [14], a network trained on the task of measuring face similarity, we report the mean distance between the embedding of a texturized frontal render and the embeddings of ground truth images.

We evaluate our method on 30 renowned personalities. The results are shown in Table 1.

| | CLIP ↑ | DINO ↑ | FaceNet ↓ |
|---|---|---|---|
| lat. mean | 17.67 | 42.21 | 1.40 |
| **ours** | 28.27 | 47.67 | 1.07 |
| gt images | 32.11 | 78.63 | 0.63 |

Table 1. Scores Comparison: We report the score of renders of the latent mean, as upper bound we report the score of ground truth images with the prompt (CLIP) or the ground truth images (DINO, FaceNet).

**Discussion** In the domain of facial analysis, fine-grained geometric disparities hold great significance, as the human eye can easily recognize them. While our investigation demonstrates the utility of CLIP Scores in guiding the generation of 3D heads, we note a limitation in their effectiveness as an evaluation metric. In many cases, we observe higher CLIP Scores for 3D heads that, upon visual inspection, are deemed inferior. Upon scrutiny, we find that neither the DINO nor the FaceNet metric offer standalone reliability. Consequently, we advocate for a cautious approach when interpreting these metrics and suggest evaluating them collectively.

**Qualitative Results** We present the qualitative results of the proposed framework in Figure 1 and Figure 3. We can generate 3D head models of celebrities and modify them as well as create 3D heads from textual descriptions.

### 4.2. Ablations

**Augmentations to stabilize CLIP Signal** As outlined above, high CLIP Scores do not necessarily imply that the 3D head faithfully represents the prompt. As a result, we observed instances where the CLIP Score guides the optimization to an undesired result. Given the strong dependency of CLIP Scores on rendering parameters, we implemented a strategy to mitigate potential adversarial optimization effects. Specifically, we introduced controlled variations within defined regions of the rendering parameters, aiming to reduce the likelihood of unintended optimization paths. Moreover, to enhance the stability of the optimization process, we introduced sampled noise into the latent codes undergoing optimization. Figure 5 demonstrates the effectiveness of this approach.

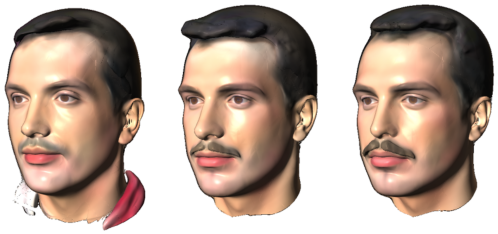


Figure 5. "Freddie Mercury": (left) extrinsic camera parameter variation, (middle) + phong parameter, light and color variation, (right) + latent code variation

**Optimization techniques to combat sparse latent codes** We utilize the AdamW variant [7] for optimization, which outperforms the vanilla implementation significantly. During the CLIP-guided optimization, the latent codes diverge, leading to structural degradation in generated heads.

Standard L2-regularization fails to address this issue, resulting in uneven decreases in code magnitude and a highly sparse latent code post-optimization. This behavior is attributed to the geometry and appearance latent code structure, each comprising 67 sub-codes with widely varying standard deviations. High-variance areas persistently expand or even diverge, while low-variance regions shrink excessively under L2-regularization.

Adam's weight decay mechanism exacerbates this issue by adaptively reducing the learning rate of weight decay in high-variance gradient regions, diminishing the L2 effect. Decoupling weight decay, as offered by AdamW, counteracts this problem. Given the significant variance disparities in the latent code, we also reduce the second momentum parameter of AdamW to mitigate sparseness. While these adjustments and AdamW optimization notably enhance the structure of the latent code, they do not resolve the issue completely (Figure 6).

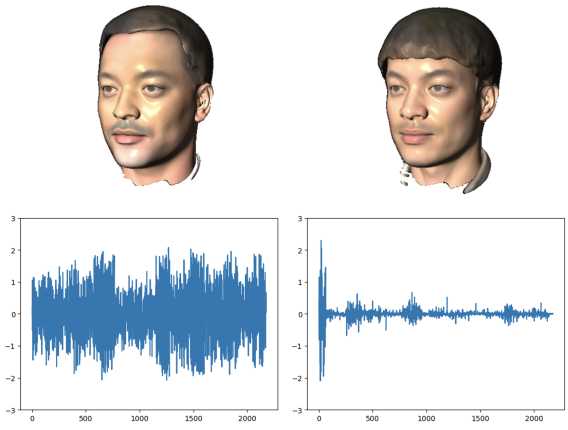



Figure 6. Latent code and textured render of "Will Smith": (left) AdamW and (right) vanilla Adam

**Influence of Texture on the Optimization** We observed that exclusively using textured head model renders during optimization negatively impacted final geometry quality, despite providing CLIP with more information. Though the exact cause remains unclear, two intuitions arise.

Firstly, even humans struggle to discern significant geometry differences in textured renders, suggesting the value of uniformly colored renders, which could benefit CLIP as well.

Secondly, our CLIP analysis demonstrated proficient identity differentiation but struggled with details within one identity. As skin, hair, and eye color define a person substantially, this might overshadow the defining geometric features of a person in the CLIP signal.

However, optimizing the geometry on only untextured head models can also have a detrimental effect on the overall result. Considering these observations, we chose to include both textured and untextured head models in an optimization batch, with a greater proportion of untextured renders. The influence of this decision is illustrated in Figure 7.

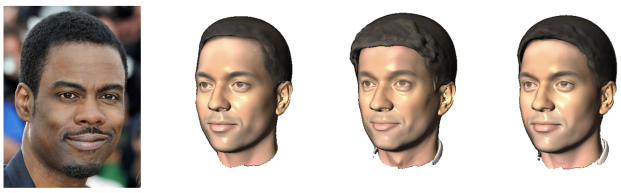
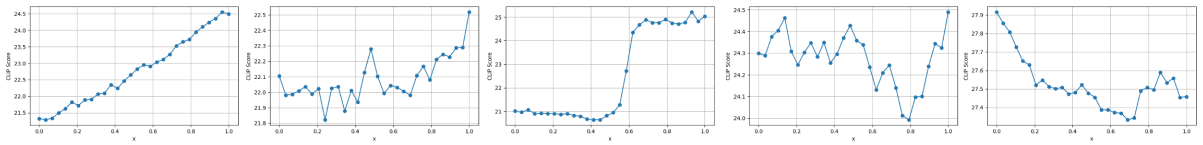


Figure 7. Ablation of texture influence with "Chris Rock" from left to right: ground truth, ours, only texture, no texture

**Non-reliable CLIP expression signal** Although our pipeline theoretically supports the optimization of latent expression codes, practical implementation did not yield the



| | Optimal behavior | Correct signal, very non-smooth | Correct signal, abrupt increase | No signal | Incorrect signal |
|---|---|---|---|---|---|
| Colored | 34.3% | 15.7% | 22.4% | 7.6% | 20.0% |
| Non-colored | 30.0% | 23.3% | 15.2% | 9.0% | 22.4% |

Figure 8. Analysis of 210 expression interpolation curves for textured and untextured renders respectively

desired outcomes in many cases. To explore the CLIP expression signal, we chose seven easily recognizable expressions from the NPHM dataset [3]. For each possible pairwise combination, we analyzed the CLIP Score when interpolating between the two expressions $\mathbf{z}_{exp,1}$, $\mathbf{z}_{exp,2}$ following:

$$\mathbf{z}_{exp} = \mathbf{z}_{exp,1} + x * (\mathbf{z}_{exp,2} - \mathbf{z}_{exp,1})$$

For the prompt, we used the description of the second expression. Latent geometry and appearance codes remained constant during the interpolation. This experiment was conducted across five different individuals in both textured and untextured scenarios. Ideally, the CLIP Score would linearly increase as the interpolation progresses towards the second expression. However, this optimal behavior was only observed in approximately one-third of cases (Figure 8). A second share of results showed a CLIP Score increase but the curves appeared non-smooth or displayed abrupt, rapid increases. These rapid increases were mostly observed for expressions such as "open mouth" or "closed eyes". Most notably, a significant portion of cases showed either no discernible signal or even an incorrect signal between the two expressions.

## 5. Limitations

Our approach encounters limitations stemming from both CLIP [10] and MonoNPHM [4]. Firstly, our reliance on the pre-trained CLIP model makes our method susceptible to inaccuracies. For specific text prompts (e.g., "Taylor Swift") or attributes (e.g., "brown eyes") CLIP provides an incorrect signal, resulting in results that do not match the desired input prompt. Additionally, Section 4.2 shows that optimizing expressions is not possible due to incorrect CLIP signals. Moreover, our method is based on the pre-trained MonoNPHM model, presenting challenges in generating outputs that significantly deviate from MonoNPHM's training data. For instance, our method struggles to optimize for individuals with very dark skin tones and is constrained in offering variations in beard types beyond the training data's scope. Furthermore, achieving an accurate portrayal of hair presents difficulties, particularly in optimizing for long, loose hair. Our method currently lacks the capability to effectively optimize for such hair characteristics.

## 6. Conclusion

In conclusion, we introduced a method for generating 3D head models from text descriptions, utilizing CLIP-guided latent code optimization. Our investigation focused on understanding the peculiarities and limitations of CLIP, CLIP Score optimization, MonoNPHM, and Adam. Notably, we optimized rendering parameters to maximize CLIP signal and augmented optimization variables to mitigate adversarial optimization. Further research is required to address issues related to latent code sparsity. Future work should also explore extending the method to incorporate facial expressions.

## References


[1] Csaba Bálint and Gábor Valasek. Accelerating sphere tracing. 2018. 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 3

[3] Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Learning neural parametric head models, 2023. 2, 5

[4] Simon Giebenhain, Tobias Kirschstein, Markos Georgopoulos, Martin Rünz, Lourdes Agapito, and Matthias Nießner. Mononphm: Dynamic head reconstruction from monocular videos, 2023. 2, 5

[5] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields, 2022. 1

[6] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. CLIP-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers*. ACM, 2022. 1

[7] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 4

[8] Pranav Manu, Astitva Srivastava, and Avinash Sharma. Cliphead: Text-guided generation of textured neural parametric 3d head models, 2023. 2

[9] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes, 2021. 1

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 3, 5

[11] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[13] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1

[14] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. 3

[15] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2